


# RoomKey: Extracting a Volatile Key with Information from the Local WiFi Environment Reconstructable Within a Designated Area

Philipp Jakubeit<sup>1</sup> <sup>a</sup>, Andreas Peter<sup>1,2</sup> <sup>b</sup> and Maarten van Steen<sup>1</sup> <sup>c</sup>

<sup>1</sup>University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands

<sup>2</sup>Carl von Ossietzky Universität Oldenburg, Ammerländer Heerstraße 114-118, 26129 Oldenburg, Germany

**Keywords:** Fuzzy Extraction from WiFi Data, Volatile Key Extraction, Location-Based Authentication Factor.

**Abstract:** We present a WiFi signal-based, volatile key extraction system called RoomKey. We derive a room's key by creating a deterministic key from the ever-changing WiFi environment and investigating the extraction capabilities of a designated area. RoomKey uses wireless beacon frames as a component, which we combine with a strong random key to generate and reconstruct the same volatile key in the room. We provide an exemplary use case using RoomKey as an authentication factor using the location-specific WiFi environment as an authentication claim. We identified and solved two problems in using location as an authentication factor: location being sensitive to privacy and the location of a user constantly changing. We mitigate privacy concerns by recognizing a particular location without the need to localize its precise geographical coordinates. To overcome the problem of location change, we restrict locations to work environments for laptop usage and allow a per-location-predetermined, designated area (e.g., a room). With the concept RoomKey, we demonstrate the potential of including environmental WiFi measurements for volatile key extraction and show the possibility of creating location-aware and privacy-preserving authentication systems for continuous authentication and adaptive security measures.


## 1 INTRODUCTION


Volatile key extraction describes a concept in which a key is not stored but reconstructed 'on the fly' when required. The main benefit of a volatile key is that the key is not stored and, therefore, cannot be extracted from any adversary. The main drawback is that the measurement used to create the key must be available and sufficiently stable for key reconstruction. Examples used for volatile key extraction are biometrics (Jagadeesan et al., 2010) and physically unclonable functions, PUFs (Schrijen and Van Der Leest, 2012). In this work, we use wireless data sent from WiFi access points as a component in building a volatile, location-specific key.


The challenge of volatile key extraction is that a key must be deterministic and identical during the initialization of the system and during the reconstruction. Due to the instability of WiFi environments, this requirement introduces a fundamental problem for the use of WiFi measurements for volatile key extraction. In this work, we show that minor variations

of the device positioning of a couple of decimeters change the WiFi measurements considerably. We aim to overcome this challenge and create a volatile key from WiFi measurements in a designated area (e.g., a room). To do so, we distinguish two phases, the generation phase and the reconstruction phase. To derive a deterministic and exact matching key, biometrics and PUF research use a concept known to the literature as fuzzy extractors (Dodis et al., 2004). We make use of this concept for RoomKey.

Creating an identical key from WiFi measurements during the generation phase and the reconstruction phase is possible (Jakubeit et al., 2022). However, they assume that the sensor (the measuring device) is spatially fixed, which considerably limits their sensor adaptability. In contrast, we want to realize key extraction from a designated area. We achieve this by conducting measurements at various locations within the respective room during the generation phase, and we succeeded in reproducing the same key during the reconstruction phase from any location within the room. Additionally, we consider off-site locations in the proximity of the room from which reconstruction must not be possible. If these measurements from off-site locations are used as negative examples during

<sup>a</sup>  <https://orcid.org/0000-0001-6216-6100>

<sup>b</sup>  <https://orcid.org/0000-0003-2929-5001>

<sup>c</sup>  <https://orcid.org/0000-0002-5113-2746>

the generation phase, we obtain an average success rate for the reconstruction within the room above 91% while reducing reconstruction successes from the off-site locations, which should not be able to reconstruct, to at most half a percent.

One application of RoomKey is to use the key as an authentication factor in a multi-factor authentication (MFA) system. In an MFA system, location recognition has the potential to establish a more reliable way to verify user identities. The whereabouts of a user add an additional validation layer. A user's consistent position across multiple authentication requests at a specific, access-controlled location provides strong evidence for the system that it deals with the valid user. For example, if the system knows a login attempt comes from the user's office, it is more likely that the entity trying to log in is indeed the user working at this office. An added benefit of location as an authentication factor is that it is seamless in itself, as it does not require user interaction.

In this paper, we present RoomKey a system to derive a volatile key in a designated area, i.e. a room. We use fuzzy extractors to transform WiFi data from on-site WiFi measurements into a secure and consistent representation, which we harden by also considering off-site WiFi measurements. Through extensive experimentation and evaluation, we demonstrate the effectiveness and feasibility of our proposed system. Our main contributions are as follows.

- We conceptualize and implement a system to convert WiFi measurements of a designated area to a volatile key such that the exact key can be reconstructed later from within the same area.
- We distinguish and discuss parameters that can be tuned to enable reconstruction capabilities inside the designated area while reducing the reconstruction capabilities for off-site locations.
- We analyze the options that an adversary has to successfully impersonate a user.
- We show that RoomKey can be used to include location as an authentication factor in an MFA system in a way that preserves privacy.

## 2 WIFI ENVIRONMENT

The widely used wireless communication protocol known as WiFi, or the 802.11 WiFi standard (IEEE Standard, 2007), facilitates device connectivity and communication within a local network. It offers wireless connectivity for a variety of applications. The 802.11 WiFi standard uses beacon frames for network discovery, synchronization, and management

within a WiFi network. Access points (APs) periodically transmit beacon frames to broadcast information about the network's capabilities. By detecting and interpreting these frames, nearby devices can identify and join the network.

A beacon frame carries various parameters and information elements that aid in network management. Their accessibility is dependent on the operating system (OS), and for unprivileged users on the Linux OS, it is limited to the Service Set Identifier (SSID), the media-access-control (MAC) address, the capability flags, the mode of the AP, the WiFi-protected-access (WPA) and robust-security-network (RSN) security flags, the frequency, and the maximum bit rate. The complete beacon frame, perceived on Windows or with privileged access on Linux, contains more fields that represent the network name, supported data rates, channel information, security settings, and other network details. A recent work identified 35 stable fields next to SSID and MAC performing on the Windows OS (Ciresica, 2023).

### 2.1 WiFi Landscape and Challenges

The WiFi signal is perceived as ubiquitous and constantly available due to its everyday accessibility and seamless usage. However, in reality, its availability is not always seamless. This fluctuating nature of the WiFi environment becomes evident in Figure 1, which shows the different availability frequencies of 33 APs observed on the same desk divided into four-by-four grids in a five-minute scan. We observe that constant availability of APs is given for four APs, with the majority of APs not being perceived consistently within and across grid cells. This experiment highlights the heterogeneous coverage of WiFi on a small scale. This poses a challenge to extracting the identical key during the generation and reconstruction phase.

### 2.2 Fuzzy WiFi Measurements for Key Extraction

Separate measurements of physical details, including WiFi signals, are prone to temporal changes and therefore pose an application challenge for key extraction. Specific criteria must be met for a measurement to be considered suitable for key extraction. In the following sections, we outline the prerequisites for a volatile key and explain how WiFi measurements and RoomKey. RoomKey satisfy these requirements.

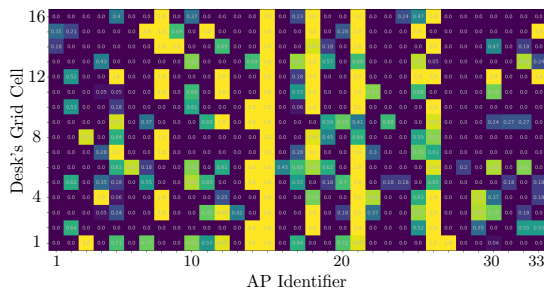


Figure 1: A heat map of a five-minute measurement showing the frequency of AP occurrences for each AP, grid-cell pair. The y-axis describes sixteen 25 by 25 centimeter cells of a physical 1-by-1-meter desk, labeled by numbers. The x-axis encodes the observed APs from AP-1 to AP-33. Cell color indicates the frequency of occurrence of the AP during five-minute measurements.

**Volatile Key Requirements**

- *Distinctiveness*: Measurements should be distinctive enough within the population. This ensures that the extracted keys are sufficiently different and contain sufficient entropy.
- *Reproducibility*: The measurements should be reproducible, which means that repeated measurements under similar conditions should yield consistent results. This reproducibility is essential for regenerating the same key from multiple measurements, allowing for successful key reconstruction.
- *Robustness*: Measurements should be robust to variations, noise, or distortions that can occur during data acquisition. Optimally, the measurements could handle inherent variations in the captured data without significantly compromising the extractability of the key.
- *Sensitivity*: The measurements should be sensitive to distinguish correct and incorrect similar measurements.

**WiFi Measurements Provisions**

- *Distinctiveness*: The distinctiveness of a location in terms of WiFi measurements is determined by the number of APs available at that location. Each location has a unique WiFi environment that makes it suitable or unsuitable for applying RoomKey. Our proposed system will be able to detect its applicability and inform the user (and the system) whether the location's WiFi environment is suitable. If two locations' WiFi measurements have no APs in common, they are distinct. Independent of the measurement, the key will be unique per user due to the usage of a device-specific key component.

- *Reproducibility*: The reproducibility of a measurement is supported by the stability over five minutes, as can be seen from the light cells in Figure 1. As we rely on a non-user-controlled WiFi infrastructure, there might be changes in the AP composition. We use error correction to deal with the fuzziness. Therefore, we can compensate for additions, removals, or changes in APs up to the number of errors that we are able to correct. If the WiFi environment changes more drastically, reproducibility is not guaranteed. However, such drastic changes are not to be expected. In a private environment, we expect that each AP changes less frequently than annually. In work environments, the company will be aware of changes in the WiFi infrastructure that allow for dedicated and announced resets for key derivation. If the system allows for a fall-back mechanism and provides re-enrollment capabilities, such reproducibility issues can be addressed under normal operation.
- *Robustness*: The robustness of a measurement is guaranteed by using error correction. There is a certain robustness against variations, noise, and distortions. However, due to the measurement of signals, attacks such as jamming, flooding, or AP pool poisoning are possible. Their real-world applicability, though, is questionable. Any of these attacks can be detected during the reconstruction phase. A denial-of-service attack is always possible, especially when dealing with wireless signals. However, with a fall-back mechanism in place, none of these attacks could cause problems. What we assume is a clean WiFi environment during the generation phase. This is reasonable as it occurs only at the initialization of the RoomKeysystem and would require an adversary to actively engage at this moment in time at the specific location. However, even when all APs are controlled by an adversary to know the exact WiFi composition, our construction with a device-specific key component makes the room's key still inaccessible to the adversary.
- *Sensitivity*: The measurement's sensitivity is the main focus of this work. We analyze how we can create a key that is extracted from WiFi measurements from the room's on-site location measurements, while close-by, off-site locations' measurements are incapable of reconstructing the key.

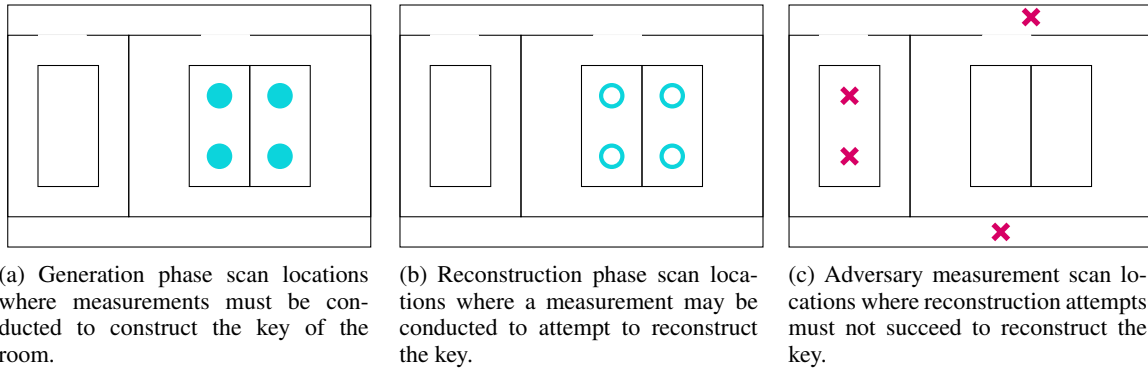


Figure 2: Schematic depiction of an office with four workstations, an adjacent office, and the immediate surroundings. In Figure 2a we depict the four on-site locations where we conduct scans during the generation phase to generate the room’s key. In Figure 2b we represent the same four on-site locations that can now be individually chosen to attempt a reconstruction. In Figure 2c we depict four off-site locations in the immediate surroundings, which should not be able to produce a measurement that allows for successful reconstruction.

### 3 SYSTEM DESCRIPTION

The goal of RoomKey is to generate and reconstruct a volatile key of a room based on the measured WiFi environment within that room. We define a scan  $\mathbf{S}$  as a set of measurements within an interval from *start* to *end* in a fixed location  $L$ , assuming a hypothetical set  $\mathbf{C}$  of all correct measurements, by:

$$\mathbf{S} = \{m \in \mathbf{C} | l(m) = L, t(m) \in [start, end]\}$$

Each *measurement*  $m$  of the WiFi environment with a WiFi sensor  $s(m)$  specifies an AP  $ap(m)$  measured at a certain time  $t(m)$  at a specific location  $l(m)$ . Using this definition of a scan, we distinguish a generation from a reconstruction phase, and on-site from off-site measurements. The sensor and location are represented as strings and the time as a UNIX timestamp, the AP is represented by the Shake Hash (Bertoni et al., 2011) of its beacon frame features.

**The Generation Phase.** In this phase (Figure 2a) the user conducts first-time measurements to create the key of a room. Therefore, the user conducts measurements at various locations within the room and combines them to extract the room’s key using RoomKey. As depicted in Figure 3, during the generation phase  $r$  different on-site locations are used to derive the room’s key and the room’s helper data. This helper data is stored by the user and can be considered to be information-theoretically secure (Dodis et al., 2004) with respect to revealing the original key.

**The Reconstruction Phase.** In this phase (Figure 2b), a user conducts measurements to reconstruct the key of a room. The user conducts measurements at

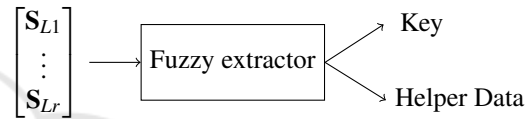


Figure 3: Schematic depiction of the generation phase. A set of scans is used to extract a room’s key and helper data.

any spot within the room and derives the room’s key using RoomKey. As depicted in Figure 4 the reconstruction phase of RoomKey consumes measurements from one location within the room and the helper data to produce the room’s key.

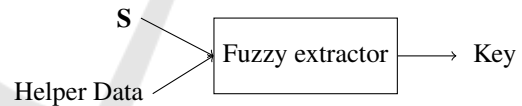


Figure 4: Schematic depiction of the reconstruction phase. A scan and the room’s helper data are used to extract the room’s key.

**On-Site and Off-Site.** Scans from two different locations  $l(m) \neq l(m')$  will vary in their AP composition depending on their spatial proximity. Close-by locations will have some overlap, whereas locations not in the WiFi range of one another are distinct. A *room* is a physically separated area (e.g. apartments, offices, classrooms). In Figure 2 we give an example of an office with four measurement locations on-site. We include off-site scans (depicted in Figure 2c) later on in the process to fine-tune a location’s room key (see Section 5).

### 3.1 Fuzzy Extractors

Fuzzy extractors as depicted in Figure 3 and Figure 4 are cryptographic constructions designed to extract stable and secure cryptographic keys from noisy or error-prone sources of biometric or environmental data (e.g., (Jagadeesan et al., 2010)). The primary goal of fuzzy extractors is to account for the inherent variability and inconsistency in measurements to ensure reliable and robust key generation. Fuzzy extractors employ error-correcting codes and secure-sketch methods to compensate for variations in the input data up to a certain threshold. By this, a stable internal baseline measurement can be generated and reconstructed from sufficiently similar fuzzy measurements. This internal baseline measurement is subsequently used as input to a strong extractor. We use universal hash function families to build such a strong extractor, which combines the internal baseline with strong randomness, such that the bits of the resulting key are uniformly distributed.

#### 3.1.1 Secure Sketch Instantiation: PinSketch

The PinSketch protocol (Dodis et al., 2004) is an instance of a secure sketch, a cryptographic method designed to handle noisy input data and produce consistent output from it. PinSketch works on a vector representation  $w$  of a set  $\mathbf{W}$  as input and reconstructs the same set vector  $w$  from a noisy input  $w'$  if  $w\Delta w' < t$ . In other words, PinSketch reconstructs the original input if the symmetric difference between the two input sets is smaller than the error-correcting capabilities  $t$ . To do so, PinSketch produces a secure sketch  $SS(w)$ , which is optimal in size and leakage: for being able to correct  $t$  elements of an  $n$ -bit vector, it has a size and leakage of  $t \times \log(n+1)$  bits. PinSketch further stands out by its optimized nature and its capability to handle varying set sizes of unordered elements. PinSketch is based on binary Bose–Chaudhuri–Hocquenghem (BCH) codes, represented as a  $[n, k, \delta]$  code. Here,  $n$  is the total length of the BCH code,  $k$  is the number of bits in each code word, and  $\delta$  is the minimum distance of the code, such that  $\delta \geq 2t + 1$ . PinSketch uses only the support vector  $v$  (non-zero elements) of the vector  $w$ , the input set's vector representation. Therefore, the syndrome vector  $syn(v)$  can be computed in polynomial time with the parameters  $\delta$ , the logarithm of  $n$ , and the size of the support vector  $v$ . During the recovery phase, if a vector  $w'$  of an input set has a weight (i.e., the number of non-zero elements) that does not exceed  $t$ , thus half of  $\delta - 1$ , computing the support vector  $v$  from its syndrome vector  $syn(v)$  also takes a polynomial time in the parameters  $\delta$  and the logarithm of  $n$ .

#### 3.1.2 Strong Extractor Instantiation: Universal Hash Function Family

Universal hash functions are a fundamental cryptographic construction used for various applications, including data integrity verification, message authentication, and strong extractors. We focus on the approach of Carter and Wegman (Carter and Wegman, 1977) who designed a family of hash functions that possess strong randomization properties, making it computationally infeasible for an adversary to predict the hash value for a given input. This family of hash functions is known as a universal hash function family. In this construction, a random key  $(a, b)$  is selected from a predefined key space. This key is used to initialize the universal hash function family. The hash function operates by mapping the input data to a hash value. The key acts as a source of randomness, ensuring that different inputs produce distinct hash values. The formula used by Carter and Wegman to define their universal hash function family is:

$$h_{a,b}(x) = ((ax + b) \bmod p) \bmod m$$

In this formula,  $h_{a,b}(x)$  represents the hash value of input  $x$  using key  $(a, b)$ ,  $p$  is a prime number that defines the range of the hash function's output, and  $m$  is the desired hash size. By varying the values of  $a$  and  $b$  within the defined constraints, different hash functions can be generated from the same universal hash function family, providing the necessary randomization properties for secure and efficient data processing. In our scenario, this translates to having two users at the same location producing different location keys with the same measurement  $x$  as their keys  $(a, b)$  and  $(a', b')$  differ. We fix only the two moduli  $m = 2^{128}$  (respectively  $m = 2^{256}$ ) as we aim for keys of this bit size and  $p = 2^z - 1$  as the prime modulus for tuples of  $(z, l) \in (129, 25), (257, 93)$  depending on the key size, such that  $p > m$ .

#### 3.1.3 Combining Pinsketch and the Universal Hash function Family

We conduct measurements at  $r$  scan locations within one room and create a shared set of observed APs as our key set  $\mathbb{S} = \{\mathbf{S}_{L1}, \dots, \mathbf{S}_{Lr}\}$ . Using  $n$ , the number of APs,  $H_\infty$ , the minimum entropy assumed per AP, the number of bits per code word  $k$ , and the desired key size  $s$ , we determine  $t$ , the number of APs we are capable of correcting, per location:

$$t = \lfloor \frac{(n \times H_\infty) - s}{k} \rfloor \quad (1)$$

We then use PinSketch to derive a secure sketch from the key set  $SS(\mathbb{S})$ . In the following, we hash each element of the key set  $\mathbb{S}$  using Shake (Bertoni et al.,

2011) and use it as input for the universal hash function (Carter and Wegman, 1977), which we initialize with the key  $(a, b)$  per location and user. Note that the sketch  $SS(\mathbb{S})$  and the key  $(a, b)$  form the device-specific key component, the helper data, and must be stored securely on the user’s device.

### 3.2 Parameters and System Tuning

In this section, we describe the parameters of the RoomKey system that can be adapted to tune the performance of RoomKey.

**Entropy.** We focus on two types of entropy. The maximum entropy  $H_0$  and the minimum entropy  $H_\infty$  of APs. We assume a set  $\mathbf{APs}$  of all APs. With  $H_0$  being the Hartley function (Hartley, 1928) defined as:

$$H_0(\mathbf{APs}) = \log_2 |\mathbf{APs}|$$

Under the assumption of APs being limited by the unique MAC addresses, we observe that there is at most  $H_0(\mathbf{APs}) = 34$ -bit of entropy in each MAC address. The specification for MAC addresses lists 48-bit (IEEE Standard, 2007). However, due to the construction of a MAC address, the first half represents the manufacturer, online available lists of manufacturer encodings (e.g., (Allan, 2023)) show variation of about only 10 bits. Therefore, we assume the 24-bit of the second half of the MAC address plus the 10-bit of the manufacturer. This maximum entropy provides an upper bound for the unique information in an AP.

The minimum amount of entropy in an AP, the min-entropy  $H_\infty$ , is crucial for the key extraction. In the literature, two min-entropies for a WiFi AP are provided. The authors of (Jakubeit et al., 2022) estimate  $H_\infty = 9$ -bit of entropy when having only access to a restricted feature list. The authors of (Ciresica, 2023) estimate at least  $H_\infty = 17$ -bit of entropy using all stable beacon frame features. To determine the min-entropy, we assume a discrete random variable  $X$  with possible outcomes in the set of APs  $\mathbf{S} = \{x_1, \dots, x_n\}$  with each element having the probability  $p_i = Pr(X = x_i)$  for  $i \in \{1, \dots, n\}$ , the min-entropy is defined as:

$$H_\infty(\mathbf{S}) = -\log_2 \max_i p_i$$

Where  $H_\infty(\mathbf{S})$  describes the AP knowledge an entity has based on a measurement. Under normal operations, the entity is the user. However, the adversary’s knowledge determines the entropy available in an AP. We discuss this in Section 4.

**Location-Specific Parameters.** We focus on four parameters regarding the APs observed. First, the duration of a scan  $\mathbf{S}$ . It is the time we consider to construct a key during the generation phase and is defined as  $duration = end - start$  for the  $start$  and  $end$  values from a specific scan  $\mathbf{S}$ . Second, the number of unique APs observed  $n = |\mathbf{A}|$ . This number  $n$  is location-dependent, as the WiFi environment varies per location. The authors of (Jakubeit et al., 2022) report between one and one hundred observed APs. Third, the number of APs we are capable of correcting,  $t$ . For  $t$  we have two ways of determining it. Either, we choose the maximum possible as described in Equation 1 or we choose  $t$  by training it based on the off-site scans observed as described in Equation 2. To do so, we split the off-site measurements into a training and a test set and use the off-site training set  $\mathbf{O}$  to determine  $t$  to be one less than the lowest symmetrical difference of the off-site training set  $\mathbf{O}$  and each observation from the generation phase:

$$t = \min\left(\left\{|\mathbf{O} \Delta \mathbf{S}| \mid \mathbf{S} \in \mathbb{S}\right\}\right) - 1. \quad (2)$$

**Sensor Details.** We focus on two sensor details, which OS is running, and hence how much information we have available from the beacon frames and the antenna type we use. The OS determines the access to beacon frames. The difference in min-entropy from (Jakubeit et al., 2022) and (Ciresica, 2023) is rooted in the OS and the user’s access rights. Regarding the choices of antenna, Table 1 shows a real-world comparison of the three different antenna types and their implications for location measurements.

We are considering three types of antennas: PCB antennas, laptop antennas, external omnidirectional antennas. Our observation is that the type of sensor used determines how the WiFi environment is perceived. This can be seen in Table 1. We note that the use of a laptop or an external antenna increases the number of perceived access points (APs), and that an external antenna provides more consistent measurements in terms of the number of APs. However, when considering the rate of occurrence, we observe that the three antennas do not vary significantly. This suggests that the choice of antenna does not solve the issue of WiFi signal inconsistencies shown in Figure 1. Based on the observation that the laptop and the external antenna provide more APs for representing a location, we have selected an external antenna for our experiments.

Table 1: Comparison of distributions of the number of measured APs and their rate of occurrence from example measurements conducted at the same measurement site between three different antennas.

Antenna type	Number of APs			
	min	max	mean	$\sigma$
PCB	25.00	33.00	29.32	1.74
Laptop	1.00	70.00	48.74	21.96
Omnidirectional	18.00	71.00	52.83	6.88

## 4 SECURITY ANALYSIS

In this work, we focus on *semi-public spaces*: locations where an on-site and an off-site exist with anyone being able to access the close by off-site. The locations we choose to measure are university offices. Even a stranger could access the perimeter of the office, while we consider adversaries not to be capable of entering offices unnoticed.

We assume the user to be cooperative and honest. Both assumptions are reasonable as they are in the user's best interest. RoomKey enables a user to derive a volatile key from measuring the environment to be used as additional secret information (e.g., in an authentication context). Therefore, the user storing the key or the AP set would render a volatile key obsolete and threaten the user's authentication claim. Nonetheless, we expect some users to do this analog to token forwarding in classical two-factor authentication systems. However, due to the use of dedicated software forwarding requires reengineering the inner workings of the scanning software, or emulating the WiFi signals in the environment.

Recall that the device-specific key component, the helper data, contains the secure sketch  $SS(w)$  and the key  $(a, b)$ . From the secure sketch, we can derive  $k$ , the number of bits per AP representation, and  $t$  the number of APs we are capable of correcting. In the following, we distinguish whether the adversary possesses this helper data or not.

**Without the Helper Data.** An adversary not knowing the helper data is required to brute force the key  $(a, b)$ . Assuming  $a$  and  $b$  to be sufficiently strong, 129-bit (257-bit respectively) in our exemplary use case, it is infeasible for an adversary to guess them correctly.

**With the Helper Data.** An adversary knowing the helper data has knowledge of the coefficients. Therefore, the success of an attack depends on the amount of knowledge the adversary has on the user's WiFi composition. We assume the adversarial WiFi composition to be the set  $\mathbf{A}$ , the set of APs the adversary

assumes to be in  $\mathbf{S}$ . We split this adversarial set in  $\mathbf{A} = \mathbf{A}^+ \cup \mathbf{A}^-$ , with  $\mathbf{A}^+$  containing only APs that are correct,  $\mathbf{A}^+ \subseteq \mathbf{S}$  and  $\mathbf{A}^-$  containing only APs that are incorrect,  $\mathbf{A}^- \cap \mathbf{S} = \emptyset$ . We distinguish four levels of adversary knowledge:

1. *No awareness of the WiFi-composition:* The adversary has to guess at least  $n - t$  correct APs from  $2^k$  possible choices. Assuming from our example  $k = 34$  and the lower assumed min-entropy of 9-bit, an adversary is required to guess at least  $n - t = 15$  APs. This number is fixed as we require at least 16 APs to derive a 128-bit key and we assume that we have at least 1 AP more to correct for. Note that this is the minimum of APs required and error correction applied. This implies that the probability of a correct reconstruction gets only worse for the adversary if more APs are considered. It is infeasible for an adversary to guess correctly with a probability of:

$$Pr[\text{Reconstruct} \mid |\mathbf{A} \cap \mathbf{S}| = n - t] = \frac{1}{\binom{2^k}{n-t}}$$

$$\text{e.g., at least } \frac{1}{\binom{2^{34}}{15}} = 2^{-469.74}$$

2. *The adversary knows too few correct APs,  $|\mathbf{A}^+| < n - t$ :* The probability of an adversary completing the set with correct APs not observed opens the whole search space of 1 in  $2^{34}$  per AP. The probability for an adversary to guess correctly is:

$$Pr[\text{Reconstruct} \mid |\mathbf{A}^+| < n - t] = \frac{1}{\binom{2^k}{(n-t)-|\mathbf{A}^+|}}$$

Assuming  $n - t = 15$  APs required, this implies a probability of  $2^{-34}$  for  $(n - t) - |\mathbf{A}^+| = 1$ . From two missing APs onwards, we exceed the NIST recommendations for passwords (NIST, 2021) with 66.99 bits required. The search space is increased per missing AP until only one AP is correct, in which case the adversary would have a probability of  $2^{-439.65}$  to guess correctly.

3. *The adversary knows sufficient correct APs, but the adversarial WiFi-composition set contains too*

many incorrect APs,  $|\mathbf{A}^+| \geq n - t$  and  $|\mathbf{A}^-| > t$ : The probability of an adversary to reconstruct correctly is turned upside down. The adversarial set  $\mathbf{A}$  contains sufficient information to attempt a successful reconstruction, however,  $\mathbf{A}$  contains too many incorrect elements for reconstruction,  $|\mathbf{A}^-| > t$ . This implies that the adversary is required to either choose only the correct APs from  $\mathbf{A}$  or to drop incorrect APs from  $\mathbf{A}$ . The probability for an adversary to choose only from  $\mathbf{A}^+$  with  $choose = \mathbf{A} \cap \mathbf{S} = n - t, |\mathbf{A}^+| < n - t, |\mathbf{A}^-| > t$  is:

$$Pr[\text{Reconstruct by choosing} | choose] = \frac{1}{\binom{|\mathbf{A}|}{n-t}}$$

The probability of an adversary dropping the incorrect APs from  $\mathbf{A}^-$  for  $drop = |\mathbf{A} \cap \mathbf{S}| \geq n - t, |\mathbf{A}^+| < n - t, |\mathbf{A}^-| > t$  is:

$$Pr[\text{Reconstruct by dropping} | drop] = \frac{1}{\binom{|\mathbf{A}|}{|\mathbf{A}^-|-t}}$$

The adversary would try to estimate the lower of these two probabilities. Assuming again the minimum of 15 correct APs being required, and a scenario of  $t = 1$ , the constraints of this case are satisfied as soon as the adversary observes at least 17 APs. In that case, the probability of choosing the correct 15 APs would be  $1/\binom{17}{15}$  thus a chance of 1 in 136. Removing the single incorrect AP would be the better choice in this setting. As  $\binom{17}{1}$  equals a chance of 1 in 17. Generally speaking, dropping incorrect APs is beneficial if fewer than  $n - t$  APs are dropped. If more APs must be dropped, selecting the correct  $n - t$  APs is more beneficial. The two options to decrease the chances of success for the adversary are either to decrease  $t$  or to increase  $n$  or  $|\mathbf{A}|$ .

4. *The adversary knows sufficient correct APs and the adversarial WiFi-composition set contains sufficiently few incorrect APs,  $|\mathbf{A}^+| \geq n - t$  and  $|\mathbf{A}^-| \leq t$ :* The probability of an adversary to reconstruct the correct WiFi-composition is 1. For an adversary holding the helper data, this scenario implies that the system is secured only by the protection mechanisms of the device containing the helper data and the password chosen by the user (assuming an MFA scenario).

The previous analysis shows that we need to keep an adversary either in a state of knowing not enough correct APs or knowing too many incorrect APs. The safer approach is to reduce knowledge of the correct APs. This is difficult to implement in the real

world as most WiFi APs are placed in hallways or hub spaces to guarantee wide coverage. Reducing adversary knowledge of APs can be done in two ways from our perspective. On the one hand, a site could be physically access-controlled, and the APs present in the access-controlled area are placed such that a sufficient number of APs are not measurable from outside the access-controlled area. On the other hand, WiFi 6 introduced fine time measurement (FTM) in the updated 802.11 standard as described by (Henry, 2021), which provides the capability of measuring accurate range. If ranging information is included, such a limit can be used to control the desired perimeter of WiFi signals. However, it is more likely that the adversary observed too many APs when being at a close or even multiple locations (e.g., in several positions around a desired location).

Recall that all probabilities described imply that the adversary got hold of the helper data of the user (compromised the user's device and security) and that the adversary is sufficiently close by, which is achievable only by a very limited set of adversaries: those adversaries capable of attacking WiFi-capable sensor in the surroundings and the device of the user remotely, and adversaries who physically get hold of the helper data and can be sufficiently close by to conduct WiFi measurements in the proximity of the user's measured location.

To prevent an adversary from getting hold of the user's device. We can apply methods known to the literature to safeguard a second factor. Due to the choice of transparency, storing the helper data in a secure element or utilizing the trusted execution environment does not suffice. This also holds for the locking features employed by the device. Therefore, when using a transparent setting having possession of the user's device might be sufficient for an adversary. Therefore, the only sound option is for the user to be educated and to carry out best practices. As soon as the device is stolen or lost, the user is asked to use locking functionality or perform a remote wipe to resolve any chance that the adversary will use the device masking as the user.

To prevent an adversary from conducting a similar WiFi scan, we can utilize Equation 2 from Section 3.2. We can mitigate attacks of a close adversary by deliberately reducing the reconstruction probability for a valid user while reducing the reconstruction probability for an adversary considerably. Assuming a system with limited trials for a user request these probabilities indicate that an adversary would not just be required to acquire the helper data (extracted from the user's device) but also needs to be in the room the user chose for applying RoomKey.



## 5 EXPERIMENTS

We conduct measurements for 120 seconds from different offices on different university campuses.<sup>1</sup> We split the measurement data into a training and test set of 60 seconds each and repeated the experiment a hundred times. We present the average probabilities for a successful reconstruction. Location-wise, we consider different office sizes in terms of workspaces (desks) available. First, we consider an office with one desk only (Office 1). Next, we extend the measurements to more than one position within a room. We continue with two desks (Office 2), three and four desks (Office 3 and Office 4), and an office with ten desks (Office 10). We conducted multiple scans in the offices, one per desk. Further, we conduct scans in the surroundings of the office, either in public spaces or within an adjacent office. We present results for both choices of min-entropy discussed above, 9-bit and 17-bit of min-entropy.

### 5.1 Results

We observe from Table 2 the relevant results for a min-entropy of assuming 9-bit per AP. Firstly, the office with a single desk performs the best, which we expected as the key set  $\mathbb{S} = \{\mathbf{S}\}$  is effectively one stationary scan. Further, we observe that longer measurement times during reconstruction increase the probability of successful reconstruction. This relationship also holds for intermediate measurement times like two or five seconds but we will omit these in this paper to focus on the capabilities of RoomKey. Furthermore, we can see that deriving longer keys reduces the probability of a successful reconstruction and that the overall probability of a successful reconstruction is, as we expected, location-dependent. For an office with only a single desk, we can reconstruct in 100% of the trials when considering enough seconds during reconstruction. For the office with ten desks, we got the worst results of being able to reconstruct the key only in 78% of the trials. Further, we want to highlight the fact that choosing not the maximum value for  $t$  (Equation 1), but instead training for a value (Equation 2), always decreases the probability of successful reconstruction (for locations with multiple positions considered). In general, it is notable that with the low min-entropy of 9-bit, none of the off-site locations would be capable of reconstructing successfully, which makes sense as we require most APs for key derivation and have less information left for error correction.

<sup>1</sup><https://gitlab.com/roomkey1/data>

We observe from Table 3 the relevant results for a min-entropy of assuming 17-bit per AP. We get very good reconstruction capabilities for the smaller offices for the on-site locations but unfortunately also for some off-site locations. We observe that using a trained  $t$ , applying Equation 2, reduces the probability of a successful reconstruction as fewer APs can be corrected for the on-site locations as well as the off-site locations. Again, increasing the measurement time, improves the probability of a successful reconstruction. However, compared to the 9-bit min-entropy case we only require a third of the duration. The key length again reduces the probability of a successful reconstruction, although less compared to the 9-bit min-entropy case. Even with 256-bit keys we get an average probability of a successful reconstruction greater than 91% for the on-site location and smaller 0.1% for off-site location.

The results in Table 3 show that we can fine-tune the performance of the RoomKey system to our needs and the location modalities. In general, the reconstruction with the assumption of a min-entropy of 17-bit performs well for the on-site and unfortunately also for the off-site locations. We can change that by reducing the general performance by lowering the number of APs that can be corrected, and train  $t$  based on the off-site locations. Explicitly tailoring the error correction capabilities to the environment and the off-site locations has the disadvantage of decreasing the reconstruction probabilities from on-site locations as well. We increase these probabilities of a successful reconstruction by measuring for an increased duration. How long we need to scan is determined by the information, the min-entropy, in each AP. While we require ten-second measurements to get decent results for the 9-bit min-entropy case, three seconds are sufficient when assuming 17-bit min-entropy to outperform these results. This is also congruent with our expectation, as we require only two APs of 17-bit min-entropy to correct one AP represented by 34-bit, while we require four APs when assuming 9-bit min-entropy per AP.

### 5.2 Comparison

In our work, we present a system to generate a key within one designated area, a room. We achieve this by utilizing WiFi beacon frames and fuzzy extraction. The underlying concept is presented by (Jakubeit et al., 2023) for static locations. We utilized this concept and showed that WiFi consistency becomes more relevant with multiple scanning locations for key extraction compared to a static location. We extend it by a thorough adversary analysis, result-

Table 2: The results of 9-bit min-entropy, comparing different office locations, derived key strengths, scan time, and applied error correction tolerances.

Location	1s for reconstruction		10s for reconstruction					
	64-bit key		64-bit key		128-bit key		256-bit key	
	Maximum $t$	Trained $t$	Maximum	Trained $t$	Maximum	Trained $t$	Maximum	Trained $t$
Office 1	0.62	0.63	1.0	1.0	1.0	1.0	1.0	1.0
Office 1 Hallway	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Office 1 Adjacent Office	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Office 2	0.0005	0.0005	0.997	0.998	0.996	0.994	0.962	0.95
Office 2 Hallway	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Office 2 Adjacent Office	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Office 3	0.022	0.024	1.0	0.999	1.0	0.999	0.994	0.993
Office 3 Hallway	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Office 3 Adjacent Office	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Office 4	0.17	0.16	0.992	0.98	0.992	0.991	0.98	0.97
Office 4 Hallway	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Office 4 Adjacent Office 1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Office 10	0.0	0.0001	0.78	0.72	0.65	0.64	0.40	0.39
Office 10 Hallway	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Office 10 Adjacent Office	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 3: The results of 17-bit min-entropy, comparing different office locations, derived key strengths, scan time, and applied error correction tolerances.

Location	1s for reconstruction		3s for reconstruction			
	64-bit key		64-bit key		128-bit key	256-bit key
	Maximum $t$	Trained $t$	Maximum $t$	Trained $t$	Trained $t$	Trained $t$
Office 1	0.95	0.94	1.0	1.0	0.999	1.0
Office 1 Hallway	0.0	0.0	0.0	0.0	0.0	0.003
Office 1 Adjacent Office	0.42	0.009	0.41	0.007	0.007	0.0
Office 2	0.79	0.026	0.99	0.78	0.77	0.77
Office 2 Hallway	0.59	0.0	0.59	0.0	0.0	0.0
Office 2 Adjacent Office	0.88	0.0	0.89	0.0	0.0	0.005
Office 3	0.96	0.55	1.0	0.995	0.994	0.996
Office 3 Hallway	0.83	0.005	0.86	0.005	0.003	0.003
Office 3 Adjacent Office	0.35	0.0	0.37	0.0	0.0	0.0
Office 4	0.88	0.35	0.998	0.92	0.92	0.92
Office 4 Hallway	0.0	0.0	0.0	0.0	0.0	0.0
Office 4 Adjacent Office	0.87	0.0	0.86	0.0	0.0	0.0
Office 10	0.24	0.21	0.98	0.925	0.97	0.91
Office 10 Hallway	0.025	0.002	0.02	0.003	0.0008	0.0
Office 10 Adjacent Office	0.0	0.0	0.0	0.0	0.0	0.0

ing in a system tailored to one room. By this, we minimize the likelihood of an adversary successfully reconstructing a user's room key while allowing a more seamless experience for the user.

Other works utilize WiFi for indoor navigation (e.g., (Yang and Shao, 2015)) or for position-based cryptography (e.g., (Chandran et al., 2009)). What these solutions have in common, is their dependence on control of the AP. In contrast, our work is stand-alone and can adapt to changes in the specification (e.g., if new stable entropy features are added the entropy estimations can be updated).

Already existing aspects of updates to the WiFi specification such as channel state information (e.g.,

utilized for indoor positioning (Song et al., 2021)) and round-time trip (RTT) already utilized to measure the distance to nearby RTT-capable APs (Mohsen et al., 2023). However, it remains a task for future research on how to utilize such updates for key derivation.

In this work, we build upon previous research such as (Rayani and Changder, 2023) by emphasizing the distinct and measurable attributes of location. While location is only mentioned in the context of transparent authentication as a component of behavioral biometrics, we propose considering location as an independent factor for transparent authentication, alongside physiological and behavioral biometrics. By analyzing the sequence of locations, behavioral biometric

profiles can be derived. However, often the measuring the context of a user in terms of location suffices. E.g., in cases where the office's room key is used to access the work email, there is no need for a behavioral profile to determine its validity.

## 6 DISCUSSION

We evaluated the functionality and feasibility of RoomKey. In the following, we highlight three distinguishing elements of RoomKey and propose one perspective of fine-tuning.

**Privacy-Preserving Location Claim.** In classical setups in which a location claim is based on the geolocation of a user, this information is considered personally identifiable information (PII) (European Parliament and Council, 2016). In the case of RoomKey we do not care about the localization of a user. Our focus lies on the recognition of a previously observed environment, expressed in measurements of the WiFi beacon frames observed. This composition of detected signals is location-specific. However, we process it in such a way (hardening it with Shake and the universal hash function) that it produces a location-specific key without revealing the location. Due to the use of the random coefficients to initialize the universal hash function, the location information differs per entity. Because neither the system nor an adversary is capable of deriving any PII from either the AP details or a location, RoomKey provides a privacy-preserving location claim in the form of the location measurement and the user-specific key.

**Authentication Towards Transparency.** We see RoomKey as a way to reduce the 'human in the loop' requirement by adding two authentication claims. A location factor and a possession factor. We base these claims on location recognition and the strong random key stored on the user's device. Our proposed scheme allows for an authentication mechanism to balance the need for confirmation. If a check of physical origin and device access is sufficient, an authentication system can make a RoomKey-authentication request at any time. Our findings show that one up to ten seconds suffices to enable the validation of location persistence. This can be used to either replace traditional authentication prompts or to introduce another layer of transparent authentication.

**Including Progression in Authentication Systems.** We discussed single locations and their specific WiFi

composition. We included off-site locations only in our adversary analysis. However, using more than one location could be a valid use case for behavioral profiles. Imagine a scenario where you go to a certain restaurant on Wednesdays after sports, which you do after work. In this routine, you have a valid recognition of your location at the office, at the gym for about one hour, and at the restaurant about twenty minutes later. An additional hour later, you pay the bill without requiring confirmation due to your behavioral profile match. In case of an alternation to this routine, the system could require a "human in the loop" factor.

**Fine Tune.** We can fine-tune the system further by considering the rate of occurrence of an AP during the key generation in  $\mathbb{S}$  or during the key reconstruction in  $\mathbb{S}$ . Considering only APs with a rate of occurrence of more than 10%, we can tune our worst performing location, Office 2, to 93% successful reconstructions. By only considering the most frequently occurring APs each of the examined locations reconstructs in at least 91%. The minor downside is that the reconstruction from an off-site location is increased from at most 0.5% to at most 0.8%.

## 7 CONCLUSION

We explore how to derive a location-specific key for location recognition of an area (e.g., a room) in terms of measured WiFi beacon frame features. One application is to enhance authentication systems with a user's location claim. Traditional authentication methods seldom include location as an authentication claim. The reason for this is presumably rooted in location information being privacy sensitive as well as temporal and spatial heterogeneity of the users' locations. We address both shortcomings of location as an authentication factor by focusing only on recognition instead of localization and fixing the environment of a location to workspaces in which a user is regularly using the device.

Integrating location recognition into MFA systems adds the benefit of context-sensitive authentication. RoomKey assures that the entity that successfully authenticates has access to the helper data and a previously registered location. This extra knowledge reduces the risk of user impersonation or remote attacks.

We emphasize the privacy-preserving nature of RoomKey. Instead of relying on geolocation expressed in coordinates or similar, which can raise privacy concerns, our approach uses fuzzy extractors to extract a secure key from the WiFi signal com-

position. This ensures that users' locations remain protected while still enabling effective authentication based on the unique features of a location.

We focus on a dedicated area to apply RoomKey. We consider on-site and off-site locations to balance the set of measured APs for robust authentication. By selecting the number of APs considered and additionally training on proximity locations that must not be able to reconstruct a key, locations are distinct, while on-site and off-site measurements can be differentiated. We observe an average success rate of 91% successful reconstructions from an on-site location, while we have at most 0.5% successful reconstruction from an off-site location. Our results further show that smaller rooms can deal with and perform better with less information assumed (e.g., a minimum entropy of 9 bits), while larger rooms require more information per AP to provide similar performance (a minimum entropy of 17 bits). However, an increased minimum entropy demands the mitigation techniques provided.

In conclusion, the integration of location recognition into MFA systems through RoomKey represents an improvement in authentication capabilities and the use of WiFi beacon frames to derive a key not in a fixed location but in a designated area. Our approach adds an employable method to use location in a privacy-preserving manner while enhancing the user experience by reducing the number of authentication prompts. We envision RoomKey to strengthen the existing authentication infrastructure and open up new possibilities for seamless authentication.

## REFERENCES

- Allan, A. (2023). Macvendor. Website. <https://gist.github.com/aallan/b4bb86db86079509e6159810ae9bd3e4>.
- Bertoni, G., Daemen, J., Peeters, M., and Van Assche, G. (2011). The keccak sponge function family: Specifications summary. Ref: [http://keccak.noekion.org/specs\\_summary.html](http://keccak.noekion.org/specs_summary.html).
- Carter, J. L. and Wegman, M. N. (1977). Universal classes of hash functions. In *Proceedings of the ninth annual ACM symposium on Theory of computing*, pages 106–112.
- Chandran, N., Goyal, V., Moriarty, R., and Ostrovsky, R. (2009). Position based cryptography. In *CRYPTO*, volume 9, pages 391–407. Springer.
- Ciresica, V. (2023). Authentication method for windows os based on location classification using wifi signals. Master's thesis, University of Twente.
- Dodis, Y., Reyzin, L., and Smith, A. (2004). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International conference on the theory and applications of cryptographic techniques*, pages 523–540. Springer.
- European Parliament and Council (2016). Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). Accessed: 07.2023.
- Hartley, R. V. (1928). Transmission of information 1. *Bell System technical journal*, 7(3):535–563.
- Henry, J. (2021). *Indoor Location: study on the IEEE 802.11 Fine Timing Measurement standard*. PhD thesis, Ecole nationale supérieure Mines-Télécom Atlantique.
- IEEE Standard (2007). Wireless lan medium access control (mac) and physical layer (phy) specifications. <https://www.iith.ac.in/~tbr/teaching/docs/802.11-2007.pdf>.
- Jagadeesan, A., Thillaikarasi, T., and Duraiswamy, K. (2010). Cryptographic key generation from multiple biometric modalities: Fusing minutiae with iris feature. *International Journal of Computer Applications*, 2(6):16–26.
- Jakubeit, P., Peter, A., and van Steen, M. (2022). The measurable environment as nonintrusive authentication factor on the example of wifi beacon frames. In *International Workshop on Emerging Technologies for Authorization and Authentication*, pages 48–69. Springer.
- Jakubeit, P., Peter, A., and van Steen, M. (2023). Lockey: Location-based key extraction from the wifi environment in the user's vicinity. In *Proceedings of the eighteenth international conference on information security practice and experience*.
- Mohsen, M., Rizk, H., Yamaguchi, H., and Youssef, M. (2023). Locfree: Wifi rtt-based device-free indoor localization system.
- NIST (2021). Digital identity guidelines, authentication and lifecycle management. <https://pages.nist.gov/800-63-3/sp800-63b.html>.
- Rayani, P. K. and Changder, S. (2023). Continuous user authentication on smartphone via behavioral biometrics: a survey. In *Multimedia Tools and Applications*, pages 1633–1667. Springer.
- Schrijen, G.-J. and Van Der Leest, V. (2012). Comparative analysis of sram memories used as puf primitives. In *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1319–1324. IEEE.
- Song, Y., Chen, B., Wu, T., Zheng, T., Chen, H., and Wang, J. (2021). Enhancing packet-level wi-fi device authentication protocol leveraging channel state information. *Wireless Communications and Mobile Computing*, 2021:1–12.
- Yang, C. and Shao, H.-R. (2015). Wifi-based indoor positioning. *IEEE Communications Magazine*, 53(3):150–157.