Journal of
Sensor and Actuator Networks

MDPI

Article

Robust Physical-Layer Key Generation Using UWB in Industrial IoT: A Measurement-Based Analysis

Lorenzo Mario Amorosa [1,2,*], Stefano Caputo [3], Lorenzo Mucchi [3] and Gianni Pasolini [1,2]

[1] Department of Electrical, Electronic and Information Engineering, University of Bologna, 40136 Bologna, Italy; gianni.pasolini@unibo.it
[2] WiLab—National Wireless Communication Laboratory (CNIT), 40133 Bologna, Italy
[3] Department of Information Engineering, University of Florence, 50139 Florence, Italy; stefano.caputo@unifi.it (S.C.); lorenzo.mucchi@unifi.it (L.M.)
* Correspondence: lorenzomario.amorosa@unibo.it

Abstract

This paper addresses the confidentiality of wireless communications in industrial internet-of-things environments by investigating the feasibility of secret key generation for link-layer encryption using ultra wideband (UWB) signals. Taking advantage of the nanosecond-level temporal resolution offered by ultra wideband, we exploit channel reciprocity to extract highly detailed, noise-like channel measurements, in line with the physical-layer security paradigm. Three key generation algorithms, operating in both the time and frequency domains, are evaluated using real-world data collected through a dedicated measurement campaign in an industrial setting. The analysis, conducted under realistic conditions, examines the impact of practical impairments, such as imperfect channel reciprocity and timing misalignments, on the key agreement rate and the length of the generated keys. The results confirm the strong potential of ultra wideband technology to enable robust physical-layer security, offering a viable and efficient solution for securing wireless communications in complex and dynamic industrial internet-of-things environments.

Keywords: physical-layer security; ultra wideband (UWB); industrial internet-of-things (IIoT); secret key generation

1. Introduction

In modern industrial environments, wireless sensors are widely deployed to monitor key physical parameters and evaluate the operational status of production processes. The collected data are typically transmitted to a central controller, enabling real-time supervision of workflows, equipment performance, and safety [1,2]. To keep costs low, such sensors are often designed with limited computational power. Many also operate under strict energy constraints, especially when battery-powered or relying on energy harvesting, which further restricts their ability to handle complex processing tasks [3].

At the same time, ensuring communication confidentiality is critical. In industrial settings, wireless transmissions are exposed to eavesdropping risks, potentially resulting in severe security breaches. Traditional cryptographic solutions, while secure, are often too resource-intensive for these constrained devices. This creates a pressing need for lightweight security mechanisms, tailored to the specific limitations of industrial internet-of-things (IIoT) deployments, that offer strong protection without overburdening the system [4]. For instance, industrial internet-of-things devices often deliberately reduce the key size to 128 bits in schemes, such as AES or Ascon, thereby achieving a favorable

Academic Editor: Antonio Virdis

Received: 17 November 2025
Revised: 12 December 2025
Accepted: 18 December 2025
Published: 23 December 2025

security-complexity trade-off under tight resource constraints. To address the limitations of conventional cryptographic algorithms in constrained environments, lightweight cryptography has emerged as a promising alternative. Algorithms such as Ascon-128, recently selected by NIST as the standard for lightweight cryptography, are specifically designed to provide robust security with minimal resource consumption, making them ideal for industrial internet-of-things applications. A key size of 128 bits is generally reported as sufficient to ensure robust protection against brute-force attacks while remaining feasible for implementation on resource-constrained industrial internet-of-things nodes.

More broadly, these solutions build upon the foundational principles of traditional cryptography, which rely on computational security, i.e., the assumption that reversing the encryption process without the secret key is computationally infeasible [5]. These techniques are generally classified into two main categories: symmetric and asymmetric encryption. In symmetric encryption, the same key is used for both encryption and decryption, whereas asymmetric encryption employs a public-private key pair.

Although symmetric encryption is computationally efficient, it suffers from the well-known key distribution problem. Asymmetric encryption addresses this limitation; however, it requires significantly higher computational resources, which makes it less suitable for resource-constrained devices commonly found in the industrial internet-of-things [6]. Moreover, its security hinges on the presumed hardness of mathematical problems—an assumption unproven and potentially threatened by quantum computing [7].

To overcome these limitations, researchers are exploring information-theoretic security, which offers strong, unconditional guarantees without relying on computational assumptions. This approach can remove the need for key exchange and ensures security regardless of an adversary's computational power [8]. The origins of this framework date back to Shannon's seminal work [9], where he introduced perfect secrecy through the one-time pad—a cipher that achieves theoretical security by XOR-ing the message with a random key of equal length. A cipher is perfectly secure if the ciphertext C reveals no information about the plaintext M, formalized as null mutual information $I(M;C) = 0$, assuming the eavesdropper has full access to C. Shannon further showed that perfect secrecy requires the entropy of the secret key K to be at least as large as that of the message M [10]. Wyner later relaxed the assumption of full eavesdropper access by introducing the wiretap channel model, in which the eavesdropper receives only a degraded version of the transmission [11]. He proved that secure communication is possible—even without a shared secret key—if the legitimate receiver's channel is better than the eavesdropper's. The key concept here is the secrecy capacity, the maximum rate at which secure transmission is achievable. Despite its appeal, implementing this advantage-based method in practice is challenging. In particular, it depends on knowledge of the eavesdropper's channel state, which is often unknown or difficult to estimate reliably [12–14].

An alternative approach to secure communication leverages a shared source of randomness between legitimate users, only partially known to an eavesdropper. This allows two users to independently generate a common secret key K for secure encryption over a public channel [15]. Maurer [16] showed that secret key agreement is possible even when the channel's secrecy capacity is zero, provided both parties access correlated random variables. His method involves an interactive public exchange—fully observable by the eavesdropper—that enables the legitimate users to establish a shared key. This key can then be used in one-time pad encryption or as input to standard symmetric-key algorithms.

Building on the idea introduced in [17], the wireless propagation environment can be leveraged as a source of common randomness. Due to channel reciprocity, both ends of a communication link observe similar channel conditions, enabling them to independently generate matching secret keys. In contrast, an eavesdropper, being spatially separated,

is unlikely to experience the same channel and therefore cannot easily reconstruct the key [18,19].

A variety of techniques have been explored to generate secret keys from the observation of physical-layer channel characteristics. Among them, the most widely used is received signal strength, due to its accessibility on commercial wireless devices [20]. Beyond received signal strength, alternative methods exploit various forms of channel diversity, such as:

- Amplitude and phase information in narrowband channels [21];
- Frequency diversity in wideband systems like orthogonal frequency-division multiplexing [22] and ultra wideband (UWB) [23–25];
- Spatial diversity offered by multiple-input multiple-output setups [18].

This approach to secret key generation is particularly well suited to energy- and compute-constrained IIoT devices, thanks to its lightweight processing requirements. However, implementing physical-layer key generation requires addressing practical challenges that can impair channel reciprocity. These challenges arise from hardware asymmetries between the transmitter and receiver (e.g., mismatched amplifiers, filters, or antennas) as well as synchronization imperfections. Such impairments reduce the correlation between channel observations at the two ends of the link, thereby compromising the effectiveness and reliability of the key generation process.

Typically, research on physical-layer security overlooks these practical aspects, which are nevertheless critical for real-world deployments. In what follows, we address these issues and provide an experimental validation of the physical-layer approach to secret-key generation in an industrially relevant scenario.

1.1. Related Work

Physical-layer key generation has been explored across a range of wireless technologies as a means to extract secret keys from the inherent randomness of the wireless channel. In [26], the authors propose a low-complexity security framework for IoT authentication and secure communication. Their approach leverages radio frequency fingerprints and wireless channel randomness to generate cryptographic keys at the physical layer. While comprehensive in integrating multiple security components, the framework relies on technologies like Wi-Fi, LoRa, and ZigBee and assumes channel reciprocity.

Similarly, ref. [27] proposes a key generation scheme specifically tailored for Bluetooth low energy, leveraging the constant tone extension field within Bluetooth low energy protocols to extract physical-layer information. While innovative, such approaches rely on narrowband signals and lack the spatial resolution necessary to fully exploit rich multipath environments, a beneficial factor for improving the robustness of key generation against attempts by an eavesdropper to derive the same key.

Other works instead adopt UWB, which provides gigahertz-scale bandwidth and nanosecond-scale pulses. These features enable substantially higher entropy per channel probe, precise multipath separation, and centimeter-level ranging. As shown in [23], these characteristics make UWB particularly well-suited for robust and secure key generation, especially in industrial environments, where clutter and non-line-of-sight conditions severely impact narrowband performance. The work in [28] takes advantage of ultra wideband multipath characteristics to minimize latency and hardware demands in physical-layer key generation. The authors demonstrate promising results with a 0.5% key disagreement rate at 10 dB SNR. However, the study is conducted in idealized indoor office and underground parking environments with predominantly line-of-sight conditions and assumes perfect channel reciprocity. Similarly, the work in [29] proposes a practical ultra wideband-based key sharing method using commercial off-the-shelf hardware. By aligning and denois-

ing channel impulse response magnitudes in the frequency-domain, the authors produce unique channel samples suitable for key extraction. However, their evaluation is limited to controlled conditions using sinusoidal probing assuming reciprocal channel conditions.

Authors in [30] explore key generation for Industry 4.0 applications, aiming to replace traditional public-key cryptography to reduce computational overhead. While it provides valuable real-world insights, the work does not consider the impact of synchronization mismatches, which can significantly affect key agreement. In [31], the authors propose a secret key generation technique for IoT communication using UWB and a discrete Fourier transform-based pre-processing stage to extract randomness from the channel impulse response. While relevant to our domain, this work is tested in controlled, non-industrial indoor scenarios with limited non-line-of-sight conditions.

The study in [32] offers an extensive experimental analysis of wireless key generation, investigating the effects of temporal variation, channel reciprocity, and spatial decorrelation. The findings support the role of multipath and mobility in enhancing key randomness, although they are conducted in diverse but non-industrial indoor environments. Finally, ref. [33] investigates key generation using simultaneously-transmitting-and-reflecting reconfigurable intelligent surfaces in a multi-user context. The study formulates a sum-secret-key-rate maximization problem, optimizing phase shifts and amplitude coefficients, and derives closed-form solutions under coupled phase constraints. While conceptually rich, the study relies solely on simulations and does not consider real ultra wideband propagation or address synchronization and non-reciprocity effects that are central to our work.

Beyond the literature explicitly focused on physical-layer key generation, recent studies have investigated the operational behavior and maintenance requirements of industrial manufacturing environments. In particular, the works in [34,35] introduce mission-reliability-oriented models and predictive maintenance strategies for multistate manufacturing systems, emphasizing how equipment degradation, process quality, and task scheduling jointly influence system performance and operational risk. Although these studies do not address wireless security or channel-based key generation, they highlight the complexity and variability of real industrial environments, characteristics that directly affect wireless propagation conditions and motivate the need for lightweight, robust security techniques such as UWB-based physical-layer key extraction.

1.2. Our Contributions and Paper Organization

While previous studies have explored the use of UWB signals for physical-layer secret key generation, they often suffer from key limitations: many rely on simulations or non-industrial measurements, assume ideal channel reciprocity, or neglect critical practical impairments such as synchronization mismatches and the presence of complex environmental conditions (e.g., line-of-sight/non-line-of-sight variability, dense multipath).

In contrast, this paper provides a comprehensive, measurement-driven evaluation of physical-layer key generation based on UWB signals acquired in a real industrial environment. Our contributions can be summarized as follows:

- We conduct a systematic analysis of ultra wideband-based key generation using real-world measurements collected in a challenging industrial setting, where dense multipath, metallic obstacles, and synchronization errors are present.
- We evaluate and compare multiple key generation techniques, including time-domain, envelope-based, and frequency-domain methods, under both idealized and practical conditions.

- We explicitly account for and analyze the impact of imperfect channel reciprocity and synchronization mismatches, quantifying how these impairments affect both key agreement rates and achievable key lengths.
- We identify algorithm configurations and processing strategies that ensure robust performance even in the presence of non-idealities. Without the goal of deriving the maximum performance achievable by physical-layer key generation algorithms, instead, we aim to uncover the inherent challenges that arise when such techniques are applied to real-world ultra wideband signals, and to demonstrate their practical viability when suitable strategies and countermeasures are adopted.
- We analyzed the computational complexity of the best-performing key generation strategy, quantifying both its time complexity and memory requirements.

The remainder of this paper is organized as follows. Section 2 defines the problem and outlines the key steps involved in physical-layer key generation. Section 3 introduces the principle of *Randomness Sharing*, illustrating how reciprocal wireless channels yield correlated observations, which are used to generate raw keys at both ends of the link. Section 4 focuses on *Information Reconciliation*, detailing methods to align raw keys and correct mismatches. Section 5 describes the measurement campaign conducted in a real industrial setting, including the experimental setup and data acquisition process. Section 6 presents and analyzes the results, emphasizing how practical parameters and impairments affect system performance. Finally, conclusions are drawn in Section 7.

2. Problem Statement

Alice and Bob are legitimate users aiming to establish a secure wireless connection while a passive eavesdropper, referred to as Eve, attempts to intercept their communications without actively interfering. Throughout this paper, we assume that Eve remains passive, merely attempting to listen to Alice and Bob's transmissions without being detected. By leveraging the reciprocity of the wireless channel, Alice and Bob can utilize their shared channel as a common source of randomness to independently generate an identical secret key. Since Eve is likely positioned differently from Alice and Bob, she experiences a different channel response, preventing her, in principle, from deriving the same key. This key is then employed to encrypt and decrypt messages exchanged between Alice and Bob over a public channel.

A typical algorithm for key generation includes the following steps [36] (Figure 1):

- *Randomness Sharing* (or Channel Probing): This step involves both Alice and Bob observing a specific channel feature in the time-domain (e.g., impulse response or received signal strength) or in the frequency-domain.
- *Advantage Distillation*: An optional step designed to "distill" observations in which Alice and Bob have an advantage over Eve.
- *Information Reconciliation*: This step is aimed at correcting key mismatches resulting from noise, interference, asymmetric equipment, and other factors. It usually includes a quantization phase of the observed feature. Key agreement is reached through public discussions conducted by Alice and Bob over a channel that is fully accessible to the eavesdropper (the public channel).
- *Privacy Amplification*: This step involves a deterministic and independent processing of the common bit sequences. Hash functions are typically used in this phase to improve key security, as they are designed to produce significantly different outputs for even slightly different inputs. As a result, any minor discrepancies between Eve's key and the legitimate key lead to considerable differences after the hash function processing.

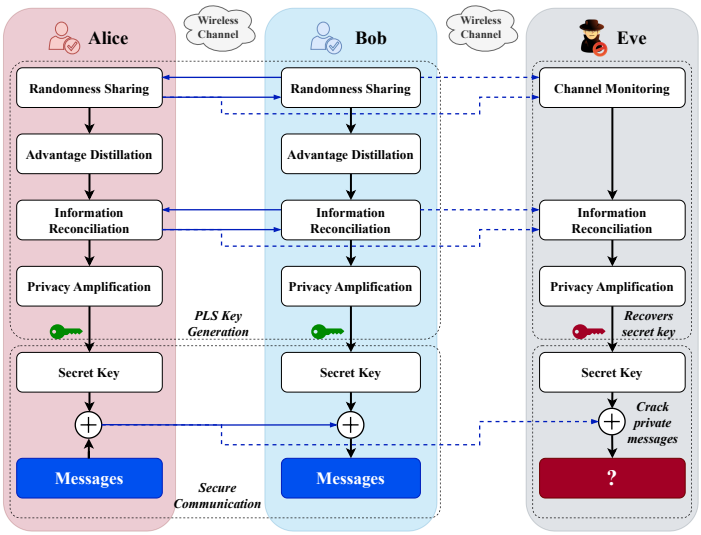


Figure 1. Physical-layer secret key extraction procedure.

Concerning the eavesdropper, it is assumed that Eve, who passively listens to Alice and Bob, has full knowledge of the algorithm used to generate the bit sequence, enabling her to replicate the key-generation process from the signals she eavesdrops.

This paper focuses on steps 1 (Randomness Sharing) and 3 (Information Reconciliation) of the key generation procedure outlined above as shown in Figure 2, with a detailed discussion provided in the following sections.

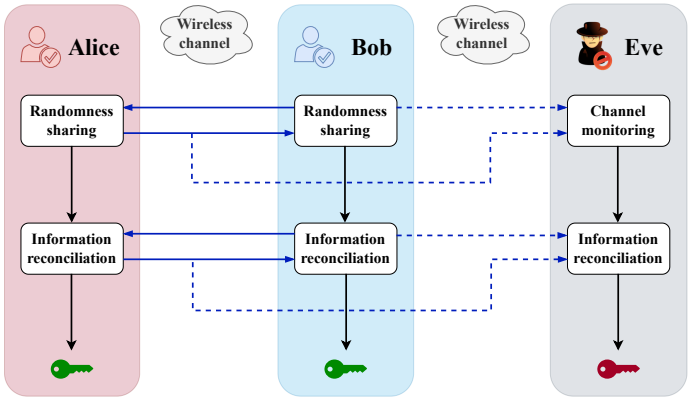


Figure 2. Illustration of the two core phases of the secret key extraction procedure addressed in this work: Randomness Sharing and Information Reconciliation.

3. Randomness Sharing

The Randomness Sharing step aims to separately obtain, at both Alice and Bob, correlated—ideally identical—observations of a channel-dependent feature, which each party can independently use as a common source of randomness for secret key generation. This process relies on the principle of channel reciprocity, which states that the wireless channel between the two parties exhibits, in principle, the same characteristics in both transmission directions, though in practice this symmetry may be imperfect.

From a practical perspective, to exploit channel reciprocity for secret key generation, the legitimate users take turns transmitting a predefined probing signal $p(t)$ with center frequency f_0 and bandwidth W, and observe the signal they receive from the other party. In formal terms, let $r_{xy}(t)$ denote the signal received by node $y \in \{\text{Alice}, \text{Bob}, \text{Eve}\} \backslash \{x\}$ when node $x \in \{\text{Alice}, \text{Bob}\}$ transmits the probing signal, with the symbol $\backslash$ denoting the

set difference, i.e., the set of elements that are in y but not in x. This received signal can be written as

$$r_{\mathrm{xy}}(t) = p(t) * h_{\mathrm{xy}}(t) + n_{\mathrm{y}}(t) = s_{\mathrm{xy}}(t) + n_{\mathrm{y}}(t) \quad (1)$$

where $*$ denotes the convolution operator, $h_{\mathrm{xy}}(t)$ is the impulse response of the channel between nodes x and y, $s_{\mathrm{xy}}(t)$ is the channel response to the probing signal $p(t)$, and $n_{\mathrm{y}}(t)$ is the additive white Gaussian noise at node y.

Under the assumption of channel reciprocity, i.e., $h_{\mathrm{xy}}(t) = h_{\mathrm{yx}}(t)$, the channel responses $s_{\mathrm{Alice\,Bob}}(t)$ and $s_{\mathrm{Bob\,Alice}}(t)$ observed by Bob and Alice are expected to be equal. In principle, this allows both parties to independently derive an identical secret key from their respective observations of the received signal, provided that the influence of noise is negligible. In practice, the channel responses are only approximately equal, i.e., $s_{\mathrm{Alice\,Bob}}(t) \approx s_{\mathrm{Bob\,Alice}}(t)$, with minor discrepancies (to be properly addressed) arising from non-simultaneous measurements and hardware mismatches in the RF front-ends of Alice's and Bob's devices.

In contrast, since Eve is located elsewhere, the channel response she observes—when the probing signal is transmitted by either Alice or Bob—is typically only loosely correlated, or even entirely uncorrelated, with those experienced by the two legitimate users. Obviously, the presence of noise $n_{\mathrm{y}}(t)$ may further amplify the discrepancies in the signals received by Alice, Bob, and Eve.

The last stage of the Randomness Sharing step requires both the legitimate users and Eve to generate discrete (i.e., sampled) representations of the time-continuous received signal $r_{xy}(t)$, each on their side. In this paper, we consider both the case where the discrete representation is obtained in the time-domain and the case where it is obtained in the frequency-domain via the Fourier transform of the received signal:

- *Time-domain representation:* The legitimate users and Eve independently sample the received signal $r_{\mathrm{xy}}(t)$, each on their side, extracting N_{xy} samples $\{r_{\mathrm{xy}}[k]\}_{k=1}^{N_{\mathrm{xy}}}$ with the same sampling frequency f_{s}, starting from an appropriately chosen time instant.
- *Frequency-domain representation:* The legitimate users and Eve, after sampling the received signal $r_{\mathrm{xy}}(t)$, each compute the corresponding fast Fourier transform, thereby obtaining their own vector $\{R_{xy}[k]\}_{k=1}^{N_{\mathrm{xy}}}$.

The details of these procedures will be provided in Sections 3.1 and 3.2. In both approaches, the Randomness Sharing step produces a numerical vector—real-valued in the time-domain case and complex-valued in the frequency-domain case—which serves as input to the subsequent steps of the key generation process.

3.1. Time-Domain Processing for Randomness Sharing

When adopting the time-domain approach, the legitimate users and Eve independently sample the received signal $r_{\mathrm{xy}}(t)$, each on their side, finally obtaining N_{xy} samples $\{r_{\mathrm{xy}}[k]\}_{k=1}^{N_{\mathrm{xy}}}$ to be used for key generation. As an illustration, Figure 3 shows an example of signal captured by Bob in response to a probe waveform transmitted by Alice, normalized to its maximum amplitude.

For a given sampling frequency f_{s}, determined by the bandwidth of the probing signal $p(t)$, the first challenge is identifying the optimal time instant at which Alice, Bob and Eve begin extracting meaningful samples of the received signal. To address this, we assume that when the key generation procedure is triggered, the signal at the receiver output (whether Alice's, Bob's, or Eve's) is continuously sampled and stored in a local memory, which is sufficiently large to accommodate samples over a time window longer than the expected duration of the channel response.

Among all stored samples, the meaningful ones are selected starting from the first time the signal exceeds a predefined threshold T_h, up to the final downward crossing below that same threshold. This allows for the removal of possible noise-only segments at the beginning of the received signal as well as scarcely significant tails. This procedure, carried out independently by Alice, Bob, and Eve, provides each of them with a local sequence $\{r_{xy}[k]\}_{k=1}^{N_{xy}}$ of N_{xy} samples that they leverage in the subsequent steps of the key generation algorithm.

A possible variant of this approach involves introducing an additional step that derives the envelope of the sequence $\{r_{xy}[k]\}_{k=1}^{N_{xy}}$. An example is shown in Figure 3, which presents a zoomed-in view of a sampled received signal (black markers) along with its corresponding envelope (red markers), both normalized to their peak value.

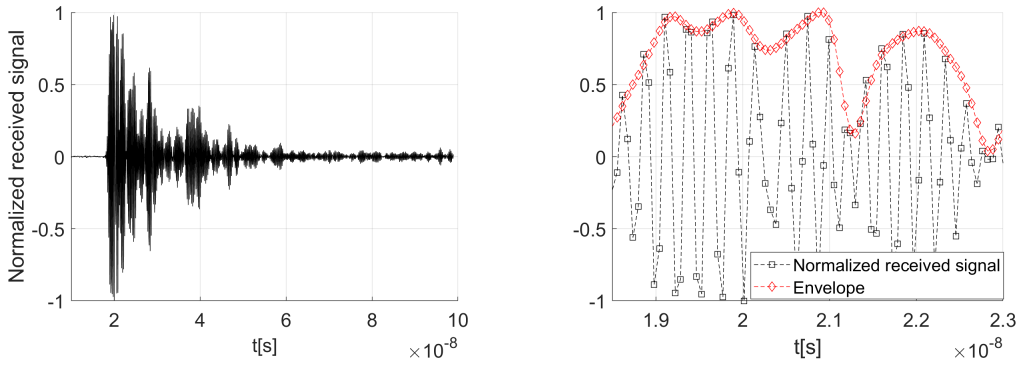


Figure 3. Time-domain representations. Example of received signal normalized to its peak value (**left** graphic). Zoomed view of a sampled received signal (black markers) and its corresponding envelope (red markers), both normalized to their respective peak values (**right** graphic).

The rationale behind this variant is to facilitate key agreement between Alice and Bob by leveraging the smoother shape of the envelope, which may reduce discrepancies caused by small-scale fluctuations in the original signal.

3.2. Frequency-Domain Processing for Randomness Sharing

An alternative strategy for *Randomness Sharing* operates in the frequency-domain. Specifically, after acquiring the time-domain samples of $r_{xy}(t)$, each node applies an fast Fourier transform to obtain the corresponding frequency-domain representation. To retain only the meaningful spectral components, each node selects the frequency-domain samples corresponding to a bandwidth $B \leq W$, centered at f_0, where B is a predefined value known to both Alice, Bob and Eve. This operation finally outputs the samples $\{R_{xy}[k]\}_{k=1}^{N_{xy}}$ used for key generation.

As will be discussed in the following, working in the frequency-domain can be beneficial, as it may render the key generation procedure less sensitive to time synchronization issues.

Remark 1. *As is evident, both procedures described above result in numerical vectors. When adopting the time-domain approach, each element of* $\{r_{xy}[k]\}_{k=1}^{N_{xy}}$ *is a real-valued number represented using* $n_b^{(ADC)}$ *bits, where* $n_b^{(ADC)}$ *depends on the resolution of the analog-to-digital converter used to sample the received signal. Typical values for* $n_b^{(ADC)}$ *range from 8 to 16 bits.*

In contrast, when adopting the frequency-domain approach, each element of the fast Fourier transform $\{R_{xy}[k]\}_{k=1}^{N_{xy}}$ *is a complex-valued number, whose real and imaginary parts require a bit-width that exceeds* $n_b^{(ADC)}$*. This increase is due to the bit growth that occurs during the fast Fourier transform computation.*

In both cases, however, the resolution is too high for Alice and Bob to obtain perfectly matching vectors, whether in the time-domain $\{r_{xy}[k]\}_{k=1}^{N_{xy}}$ *or in the frequency-domain* $\{R_{xy}[k]\}_{k=1}^{N_{xy}}$*. In fact, such fine precision makes the process sensitive to non-ideal factors like noise and hardware imperfections, which inevitably disrupt the channel reciprocity. Therefore, to ensure both parties derive an identical secret key, the output of the Randomness Sharing step must be refined through a key reconciliation step, which is discussed below.*

4. Information Reconciliation

Upon completion of the Randomness Sharing step, Alice and Bob independently obtain their respective sets of samples, which may lie in either the time or frequency-domain, depending on the selected approach. In the key generation procedure considered in this paper, the optional Advantage Distillation stage, typically following the Randomness Sharing step, is omitted. Instead, Alice and Bob proceed directly to the Information Reconciliation step, whose operation differs depending on whether it processes time-domain or frequency-domain samples. Both cases are discussed in the following subsections.

4.1. Information Reconciliation in the Time-Domain

The Information Reconciliation in the time-domain unfolds through the following sequence of operations:

1. The set of time-domain samples undergoes an additional quantization step, referred to as re-quantization in the following, which reduces the number of bits for representing each sample to $n_\mathrm{b} < n_\mathrm{b}^{(\mathrm{ADC})}$. As a result, the representation of each sample becomes less accurate, increasing the likelihood that Alice and Bob will obtain matching quantized values, despite small discrepancies in their original measurements.
   To account for variations in signal strength due to hardware mismatches, each legitimate node—Alice and Bob—independently adjusts the dynamic range of its additional quantizer to match that of the received signal. As a result, even though both parties use the same number of quantization intervals (2^{n_b}), differences in the observed signal amplitudes may lead to different dynamic ranges being covered. Consequently, the quantization step width, denoted by q, may differ between Alice and Bob.
   A similar procedure is applied when the envelope of the received signals is used for key generation instead of the raw time-domain samples. In this case, the additional quantizer is configured based on the dynamic range of the envelope signal, ensuring consistent adaptation to signal strength variations across both nodes.
   Regardless of the specific choice, re-quantization effectively mitigates the impact of front-end gain mismatches, such as those caused by amplifier gain variations or connector-induced attenuation, which are common in practical systems.
2. To further reduce the likelihood of discrepancies between the samples obtained by Alice and Bob after the additional quantization step, exclusion zones—referred to as censored regions—can be defined around the new quantization thresholds (those corresponding to n_b bits). Specifically, for each threshold, a symmetric region of total width Δ is excluded, extending by $\Delta/2$ above and below the threshold. When this strategy is employed, each party discards, prior to re-quantization, the original samples (represented with $n_\mathrm{b}^{(\mathrm{ADC})}$ bits) that fall within these regions and shares the corresponding indices with the other party, who discards the same samples accordingly.
   Figure 4 illustrates this approach in a scenario where the envelope of the received signal is leveraged for secret key generation. The envelope, initially sampled with a resolution of $n_\mathrm{b}^{(\mathrm{ADC})} = 12$ bits, is re-quantized to $n_\mathrm{b} = 3$ bits, resulting in 8 distinct quantization intervals. The original samples shown in the figure are thus mapped to

one of these intervals—unless they fall within the censored regions, highlighted with gray boxes, in which case they are discarded.

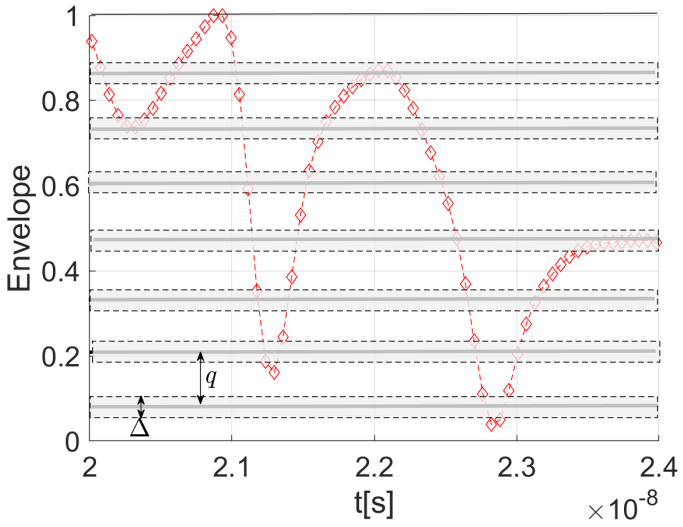


Figure 4. Example of the received signal envelope, normalized to its peak value, with quantization intervals and censored regions highlighted.

This process has two primary effects: it increases the likelihood of key agreement by eliminating ambiguous values near the re-quantization thresholds, while also decreasing the amount of usable data for key generation, ultimately resulting in shorter secret keys. This procedure applies to both the time-domain samples and their envelope, depending on the adopted approach.

3. After the re-quantization process, each node independently applies Gray coding to the resulting samples, whether derived from the signal itself or from its envelope. This encoding technique is chosen to minimize the number of bit errors caused by small quantization mismatches between Alice and Bob. The result is the generation of two preliminary bit sequences, one per legitimate node, which serve as raw keys.
4. The raw keys are then aligned through an exchange of messages between the two parties over the public channel between them. The public phase of the reconciliation strategy adopted in this work follows the method proposed in [23], which leverages linear block coding principles. Specifically, the procedure is based on a publicly known (n, k) linear block code $\mathcal{C}$ and its associated standard array. In particular, the standard array for an (n, k) linear code, with $n > k$, is a table with 2^{n-k} rows and 2^k columns, used to systematically list all 2^n possible binary sequences of length n. It is structured as follows:
    - The first row of the table contains all the 2^k valid codewords of the code.
    - Each remaining row begins with a binary sequence called coset leader, a sequence not yet listed in the array, chosen to have the smallest Hamming weight (i.e., the fewest number of 1 s).
    - The other entries in each row are formed by adding the coset leader (bitwise modulo 2) to each codeword in the first row.

    As a result, each row contains a unique set of sequences, called coset, and all 2^n binary sequences appear exactly once in the table. For reference, Table 1 shows the standard array of the (7, 4) Hamming code used in the reconciliation phase of the key generation algorithm behind the numerical results in this paper.
    Specifically, in Table 1, the coset leaders (highlighted with a dark gray background) represent all possible single-bit error patterns that can affect the valid codewords,

which appear in the first row and are marked with a light gray background. By design, each white cell in a given row and column contains the result of combining (bitwise modulo 2) the valid codeword at the top of that column with the single-error pattern defined by the coset leader of that row. Since the (7, 4) Hamming code can correct any single error, every invalid codeword in a column (i.e, each white cell) uniquely maps to the valid codeword at the top of that column.

Once the (n, k) linear block code $\mathcal{C}$ is chosen and its associated standard array is available to each party, the Information Reconciliation process proceeds as follows. Each of the legitimate parties partitions their respective raw keys into fragments of n bits. One party, say Alice, determines the coset to which each fragment belongs and communicates its index (i.e., the row index) to the other party, Bob, via a public channel observable by Eve. Simultaneously, Alice records the associated column index in the standard array of the code $\mathcal{C}$ without disclosing it. Upon receiving each coset index, Bob searches the corresponding row of the standard array to find the codeword that minimizes the Hamming distance to his own n-bit fragment. Both parties then substitute their original n-bit fragments with the resulting k-bit column indices. This procedure yields a reconciled key that is shorter than the raw key generated during the Randomness Sharing step, as each n-bit fragment is replaced by a k-bit fragment, with $k < n$.

Although Eve can eavesdrop on the public channel and determine which coset the n-bit fragments belong to, she remains unaware of the specific element within that coset. As a result, the information accessible to her is limited to the row associated to that coset.

Table 1. Standard array of the Hamming $(7, 4)$ code. The first row (light grey background) contains all valid codewords, while the first column (C1) shows the *coset leaders*.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
Coset 0	0000000	0001111	0010110	0011001	0100101	0101010	0110011	0111100	1000011	1001100	1010101	1011010	1100110	1101001	1110000	1111111
Coset 1	1000000	1001111	1010110	1011001	1100101	1101010	1110011	1111100	0000011	0001100	0010101	0011010	0100110	0101001	0110000	0111111
Coset 2	0100000	0101111	0110110	0111001	0000101	0001010	0010011	0011100	1100011	1101100	1110101	1111010	1000110	1001001	1010000	1011111
Coset 3	0010000	0011111	0000110	0001001	0110101	0111010	0100011	0101100	1010011	1011100	1000101	1001010	1110110	1111001	1100000	1101111
Coset 4	0001000	0000111	0011110	0010001	0101101	0100010	0111011	0110100	1001011	1000100	1011101	1010010	1101110	1100001	1111000	1110111
Coset 5	0000100	0001011	0010010	0011101	0100001	0101110	0110111	0111000	1000111	1001000	1010001	1011110	1100010	1101101	1110100	1111011
Coset 6	0000010	0001101	0010100	0011011	0100111	0101000	0110001	0111110	1000001	1001110	1010111	1011000	1100100	1101011	1110010	1111101
Coset 7	0000001	0001110	0010111	0011000	0100100	0101011	0110010	0111101	1000010	1001101	1010100	1011011	1100111	1101000	1110001	1111110

Commentary: The procedures described above for key reconciliation at Alice and Bob can be implemented in computationally efficient ways that avoid exhaustive searches over the standard array. Detailed algorithms are provided in Appendices A and B.

4.2. Information Reconciliation in the Frequency-Domain

As will be discussed later, a key challenge in generating secret keys from time-domain features of received signals lies in achieving precise time synchronization between the legitimate users. This synchronization is crucial to ensure that both parties acquire identical samples, and becomes especially critical when ultra wideband signals are used, due to their extremely rapid time-domain fluctuations. While this issue is often overlooked in key generation algorithms proposed in the literature, it becomes critical in real-world implementations—such as the one addressed in this paper. Indeed, to effectively exploit channel reciprocity, any practical key generation system must incorporate mechanisms to mitigate synchronization mismatches.

To address this issue, ref. [37] proposes a strategy in which the key is generated by analyzing the received signal in the frequency-domain. Specifically, consider a scenario where the sample sets $\{r_{xy}[k]\}_{k=1}^{N_{xy}}$ obtained by Alice and Bob are time-shifted versions of one another, due to one of the legitimate parties including one or more meaningful samples

at the beginning of the received signal that the other party discarded. This misalignment, exemplified in Figure 5, potentially prevents Alice and Bob from deriving identical keys.

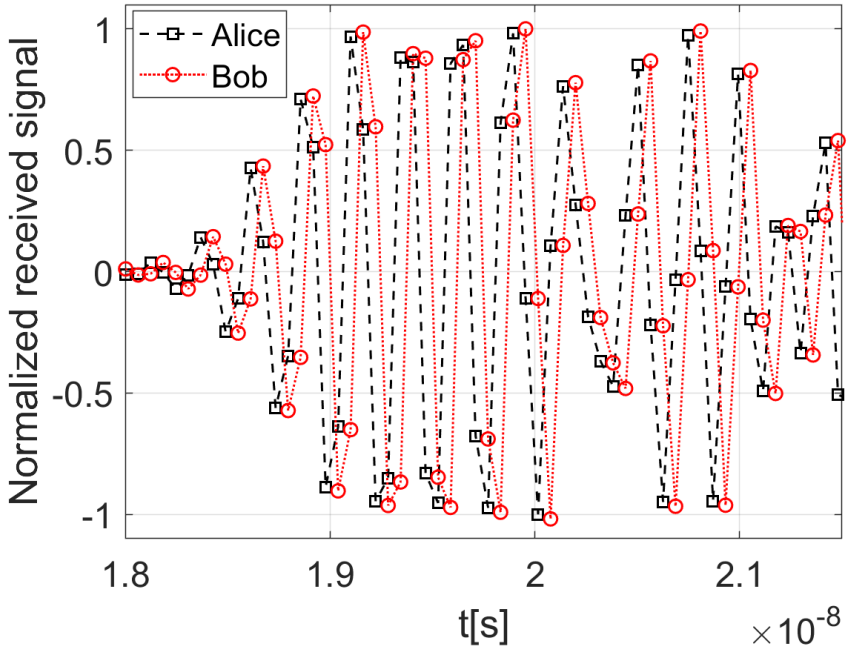


Figure 5. Time-shifted received signals of Alice and Bob.

It is well known, however, that a time delay affects only the phase component of a signal's frequency spectrum, while leaving the amplitude spectrum unchanged. Therefore, by having Alice and Bob operate in the frequency-domain and rely exclusively on amplitude information to generate the key, the process becomes inherently more robust to timing misalignment.

Specifically, as outlined in Section 3.2, this technique can be implemented at each receiver by sampling the incoming waveform, computing the fast Fourier transform of the sampled data, and extracting the magnitude of each frequency-domain component in a properly chosen bandwidth B centered at f_0. This process yields the sequence $\{|R_{xy}[k]|\}_{k=1}^{N_{xy}}$, which represents the amplitude spectrum of the discrete-time received signal. Being invariant to time shifts, it can serve as a reliable source of shared randomness for secret key generation, even in the presence of synchronization mismatches between the legitimate users.

Starting from their respective amplitude spectra $\{|R_{xy}[k]|\}_{k=1}^{N_{xy}}$, Alice and Bob re-quantize the retained samples using n_b bits. Optionally, prior to this step, they may discard ambiguous samples that fall within censored regions around the new quantization thresholds. These operations correspond to steps 1 and 2 described in Section 4.1 for the time-domain processing.

As an example, Figure 6 displays the amplitude spectra of two signals: one received by Bob from Alice, and the other received by Alice immediately after from Bob. The figure also marks, using the $\circ$ symbol, the re-quantized samples derived by Alice, and the $\times$ symbol for those collected by Bob, assuming a representation with $n_b = 3$ bits. The horizontal lines represent the boundaries (thresholds) of the 2^{n_b} re-quantization intervals for Alice and Bob. In the figure, the samples discarded by Alice and Bob, due to their original values lying within the censored regions (which are omitted in Figure 6 for clarity), are indicated by the symbols $\triangleright$ and $\triangleleft$, respectively.

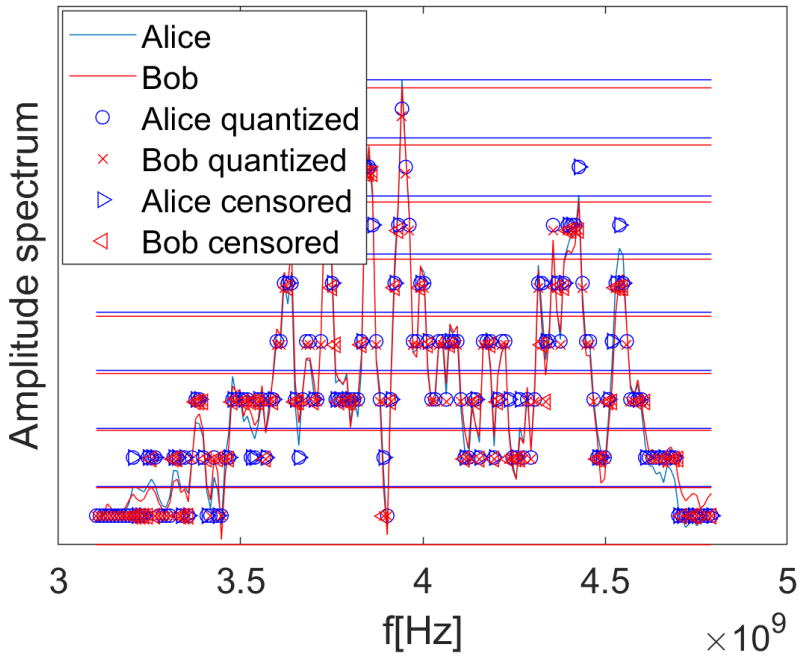


Figure 6. Amplitude spectra of the signals received by Alice and Bob, together with the re-quantized samples. Circles represent Alice's retained samples, crosses represent Bob's retained samples, and the discarded samples appear as triangular markers. The horizontal lines indicate the quantization thresholds for $n_b = 3$ bits.

Once the amplitude spectrum samples have been re-quantized and the ambiguous ones discarded, the Information Reconciliation procedure proceeds with steps 3 and 4 outlined in Section 4.1, which remain unchanged.

The Privacy Amplification step, which typically follows the Information Reconciliation stage, is not considered in this paper. Privacy Amplification is a standard step in physical-layer key generation protocols, typically applied after Information Reconciliation to eliminate any residual information that an eavesdropper might infer. It involves applying a cryptographic hash function to the reconciled key, producing a shorter but more secure key. This ensures that even small discrepancies between Eve's observations and the legitimate key result in completely different outputs. In this study, Privacy Amplification was deliberately omitted. This decision is supported by the experimental results, which consistently show that Eve is unable to reconstruct the key, even when located less than one meter from Bob. The industrial environment, characterized by dense multipath and metallic clutter, ensures low spatial correlation between channel responses, making the key generation process inherently resilient to passive eavesdropping. Nonetheless, in less cluttered or mobile environments, or in scenarios with reduced channel diversity, Eve may observe signals with higher correlation to those of the legitimate users. In such cases, Privacy Amplification becomes essential to guarantee information-theoretic security. Moreover, lightweight hash functions such as those used in Ascon or other NIST-approved algorithms [38] can be employed to maintain efficiency in resource-constrained IIoT devices. Future work may explore the integration of Privacy Amplification in dynamic or outdoor industrial settings, where the assumptions of spatial decorrelation and channel unpredictability may not hold as strongly.

Remark 2. *The length of the secret key produced after the Randomness Sharing and Information Reconciliation steps will most likely vary from one execution to the next. We emphasize that this variability does not weaken the security of the proposed physical-layer key generation scheme; on*

the contrary, it introduces an additional source of uncertainty for a passive eavesdropper. In the absence of any side information about the final key length, an eavesdropper must not only attempt to infer the key bits from observations of a channel different from that of Alice and Bob, but also implicitly determine the correct key length, thus enlarging the effective search space. By contrast, our experimental testbed demonstrates that, when appropriate strategies are adopted, Alice and Bob converge to identical keys with high probability, without any prior agreement on the key length.

5. Measurement Campaign

The measurement campaign was carried out at Top Finish, a galvanic plating facility located in Tuscany, Italy (the company is now part of Metalplus srl, www.metalplus.it, accessed on 19 September 2025). The site is divided into two distinct zones: an automated area for plating low-cost metals, and a non-automated area dedicated to high-value materials such as gold and platinum. Measurements were taken in the non-automated section, shown in Figure 7, where human operators manually immerse fashion accessories into various galvanic baths. This scenario, whose layout is depicted in Figure 8, is characterized by the predominance of metallic objects and surfaces, including a grated floor in the central corridor, and spans an area of 10.5×5.5 m^2.

Figure 7. Picture of the galvanic plating zone where the measurement campaign was performed.

In this setting, a series of experiments was performed using Time Domain PulsON 440 devices [39], a class of UWB transceivers specifically designed for high-precision ranging and positioning. These devices support peer-to-peer communication and real-time location systems applications, offering sub-nanosecond timing resolution through the ultra wideband waveform depicted in Figure 9 [40]. Operating in the 3.1–4.8 GHz band, they feature built-in antennas, a high-speed analog-to-digital converter, and USB connectivity for easy integration and data logging.

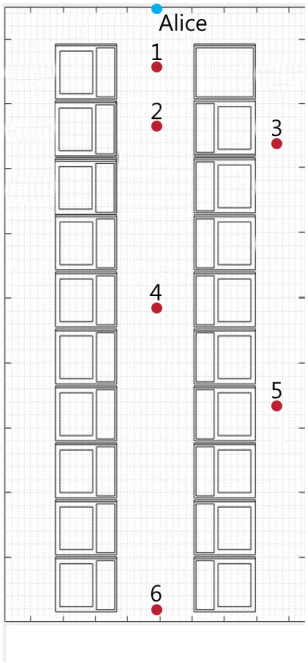


Figure 8. Layout of the galvanic plating area where the measurement campaign was carried out. The blue point indicates Alice's location, while the red points represent the possible positions of Bob and Eve.

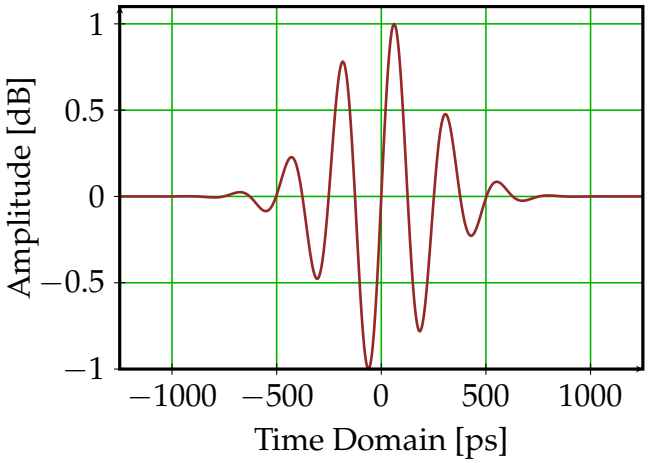


Figure 9. ultra wideband waveform employed by the PulsON 440 transceiver as shown in [40].

In our setup, one transceiver was fixed at the end of the corridor, as shown in Figure 8 (blue point), while additional transceivers were placed at increasing distances from it, corresponding to the red points numbered 1 to 6 in the figure, which refer to the following distances and propagation conditions:

(1) Line-of-sight conditions, at 1014 mm from the fixed device;
(2) Line-of-sight conditions, at 2000 mm from the fixed device;
(3) Non-line-of-sight conditions, at 3058 mm from the fixed device;
(4) Line-of-sight conditions, at 5074 mm from the fixed device;
(5) Non-line-of-sight conditions, at 6846 mm from the fixed device;
(6) Line-of-sight conditions, at 10,100 mm.

Peer-to-peer communications were established between the fixed node and the nomadic nodes, and the received signals, affected by the propagation environment, were collected, sampled at $f_s = 16.3$ Gsamples/s, and stored locally, enabling the acquisition

of a complete set of measurements for each position of the nomadic nodes relative to the fixed node.

These measurements enabled direct observation of how increasing distance and the presence of obstructions affect the time delay and multipath structure of the received signal. Most importantly, for the purposes of this paper, the sets of collected signals were used to evaluate the previously introduced key generation algorithms and to derive the results presented in the following section. Specifically, in this setup, the fixed node represents Alice, while the nomadic nodes act as Bob or Eve. When Bob occupies one of the six possible positions, Eve can be placed in any of the remaining five locations to attempt to capture the secret key by eavesdropping on the transmission from Alice.

The UWB waveforms provided by the PulsON 440 devices represent the output of a proprietary front-end that includes internal filtering, calibration and signal conditioning stages. As a result, the acquired signals do not expose the physical channel in a form that permits a reliable extraction of parameters such as the number of multipath components, NLOS ratios, amplitude attenuation due to metallic obstructions, or channel coherence times. The present study focuses on the direct use of the measured waveforms as provided by the hardware, without attempting a parametric characterization of the industrial channel. The performance results thus reflect the aggregate effect of the multipath richness, obstruction dynamics and temporal variability intrinsic to the industrial environment, even though these factors cannot be independently quantified using the measurement platform.

6. Numerical Results

In this section, we evaluate the performance of the key generation algorithms described in Sections 3 and 4, using the measurements collected as outlined in Section 5. The investigation was carried out across multiple distances between Alice and Bob, who independently generate secret keys based on the respective sets of signals acquired at each distance. Specifically, for each considered distance, a number of received signals was collected, ranging from 136 to 171 depending on the distance, and stored as digital samples with a resolution of $n_{\mathrm{b}}^{(\mathrm{ADC})} = 16$ bits. These samples were then used as input to the key generation algorithms at both legitimate ends and at Eve's side, producing local keys that may or may not match. After processing all the available signals for a given distance, four performance metrics were computed to evaluate the effectiveness of the key agreement process:

- *KeyHitRatio*, defined as the ratio of key matches between Alice and Bob to the total number of generated keys;
- *EveHitRatio*, defined as the ratio of Eve's key matches to the total number of all keys successfully matched between Alice and Bob;
- *AvKeyLength*, representing the average length of the keys, calculated over all successfully matched keys.
- *KDR*, denoting the key disagreement rate, defined as the ratio between the number of mismatched bits in the keys generated by the two parties before Information Reconciliation and the total key length.

In addition, we also analyzed the number of repetitions required by the key agreement procedure to successfully establish a common key.

As a complementary analysis, we also evaluated the computational complexity of the algorithm that achieved the best performance. Specifically, we derived:

- *Time complexity*, which quantifies the number of elementary operations (or an asymptotic upper bound thereof) required by the algorithm as a function of the input size p.

This is reported using Big-O notation, e.g., $O(p)$, $O(p \log p)$, or $O(p^2)$, and refers to the worst-case unless otherwise stated.

- *Memory requirement*, which denotes the amount of memory needed to store inputs, outputs, and intermediate data structures.

Various configurations were explored, taking into account the parameters of the key generation algorithm, the ability of Alice and Bob to synchronize their channel observations, the degree of channel reciprocity, and the distance between Eve and Bob.

It is important to note that the results presented in this section are entirely derived from real UWB measurements. The PulsON 440 devices used in the measurement campaign do not provide a direct estimate of the SNR, and the received signal quality varies naturally with distance, multipath richness, LOS/NLOS transitions, and local metallic clutter. As a consequence, SNR is not a controlled parameter in our setup and cannot be consistently reported in the figures. The performance results therefore inherently reflect the real-world SNR conditions encountered during the acquisition campaign.

In particular, with respect to the measurement setup and the factors influencing the key generation algorithm, the performance was evaluated under various conditions in terms of:

- Distance d between the legitimate users Alice and Bob, constrained by the discrete positions available to Bob as defined in the layout of Figure 8, i.e., $d \in \{1.014\text{ m}, 2\text{ m}, 3.058\text{ m}, 5.074\text{ m}, 6.846\text{ m}, 10.1\text{ m}\}$. For each distance, the Randomness Sharing and Information Reconciliation procedures were applied to the complete set of measured received signals, resulting in the generation of the corresponding secret keys at each party.
- Eve's distance from Bob, denoted as d_{Eve}. For each position of Bob, the secret key that Eve could attempt to infer from the signal transmitted by Alice, when placed in any of the remaining five positions, was derived and compared with the key established between Alice and Bob.
- Number of bits n_{b} used for re-quantization of either the received signal (in the time-domain approach) or its amplitude spectrum (in the frequency-domain approach). Results are reported for $n_{\text{b}} = 1, 2$, and 3 bits.
- Width Δ of the censored regions. All performance metrics are analyzed as a function of the ratio $\frac{\Delta}{q}$, where q denotes the width of the quantization interval.
- Time misalignment, denoted by $SyncDelay$, between the signals used by Alice and Bob to generate the secret key. This parameter models synchronization mismatches as an integer number of sampling intervals. Numerical results are presented for $SyncDelay \in 0, 1, 2$, where $SyncDelay = 0$ corresponds to perfect synchronization.

Each setting is investigated in the following, in a sequence that progressively removes ideal assumptions, thereby highlighting the gap between what is theoretically attainable and what is practically achievable.

Basic settings and terminology: Unless otherwise specified, the numerical results presented below are obtained assuming $n_{\text{b}} = 2$ bits and $\frac{\Delta}{q} = 0.3$. Furthermore, for the sake of notational simplicity, and with a slight abuse of notation, the actual distances between Alice and Bob, given by $d \in \{1.014\text{ m}, 2\text{ m}, 3.058\text{ m}, 5.074\text{ m}, 6.846\text{ m}, 10.1\text{ m}\}$ will be denoted as $d \in \{1\text{ m}, 2\text{ m}, 3\text{ m}, 5\text{ m}, 7\text{ m}, 10\text{ m}\}$ in the tables, figure captions, and discussion. This choice is purely for readability and does not imply any actual approximation.

In the following, we use the term "perfect reciprocity" to refer to the ideal case in which Alice and Bob generate the key from identical received signals. Conversely, we use the term "imperfect reciprocity" or "non-ideal reciprocity" to describe the more realistic scenario in which the key is generated from signals acquired sequentially—as occurs in practice,

with Alice and Bob taking turns transmitting the UWB probe waveform. Clearly, in this case, channel reciprocity may not be perfect due to possible changes in the propagation environment that occurred meanwhile. Furthermore, to avoid repetitive phrasing, the terms "channel response" and "received signal" will be used interchangeably in the following.

In all cases investigated hereafter, the Information Reconciliation procedure relies on the standard array shown in Table 1, which corresponds to the well-known (7,4) Hamming code.

6.1. Key Generation with Perfect Channel Reciprocity

First, we assume that Alice and Bob observe identical signals and obtain exactly the same N samples, that is, $\{r_{\text{AliceBob}}[k]\}_{k=1}^{N} = \{r_{\text{BobAlice}}[k]\}_{k=1}^{N}$. This implies perfect channel reciprocity and full synchronization between them.

To derive the performance metrics under these ideal conditions, the key generation algorithms at both Alice and Bob were, at each iteration, provided with the same sampled received signal, actually collected by Bob, from the measurement set corresponding to the considered distance d. Given the perfect synchronization and channel reciprocity, a key match was observed at each iteration, and the length of the corresponding key was assessed. Afterward, a new received signal from the same set was provided to Alice and Bob, and the procedure was repeated until the entire set for the given distance was processed. Finally, the *KeyHitRatio* and *AvKeyLength* metrics were computed.

Next, still assuming perfect channel reciprocity, meaning that the same received signal is provided to Alice and Bob, we relaxed the assumption of perfect time synchronization and explored the case where Bob starts sampling the channel response with a delay of 1 or 2 samples relative to Alice, while still collecting the same number of samples.

The results of these investigations are presented in Figure 10, which shows the *KeyHitRatio* as a function of the distance between Alice and Bob. The results are shown for perfect time synchronization (*SyncDelay* = 0) and increasing time mismatches (*SyncDelay* = 1 and *SyncDelay* = 2), with each sample quantized using $n_{\text{b}} = 2$ bits. In all cases, we assumed $\frac{\Delta}{q} = 0.3$. As expected, under ideal conditions, namely perfect channel reciprocity and perfect time synchronization (*SyncDelay* = 0), *KeyHitRatio* is 1 for all distances. In contrast, even a delay of just one sampling interval proves catastrophic, as key matching is never achieved. This is because the channel response varies so rapidly over time that even a delay of one sampling interval results in a weak correlation between the channel responses processed by Alice and Bob. Interestingly, a slight improvement is observed when a delay of two sampling intervals is introduced. This indicates that the received signal exhibits a higher autocorrelation at a lag of two sampling intervals than at a lag of one. Nonetheless, the overall correlation remains insufficient to support reliable key generation in the presence of synchronization offsets, highlighting the critical importance of precise time alignment in practical implementations.

Although not explicitly shown in the figure, we also observed that, remarkably, Eve is consistently unable to reconstruct the correct secret key by eavesdropping on Alice's transmission—even when positioned just 0.986 m away from Bob, corresponding to Bob positioned at location 1 and Eve at location 2. The rapid fluctuations of the adopted ultra wideband signal, together with the complex propagation characteristics of the industrial environment, result in channel responses that exhibit low spatial correlation even at such short distances.

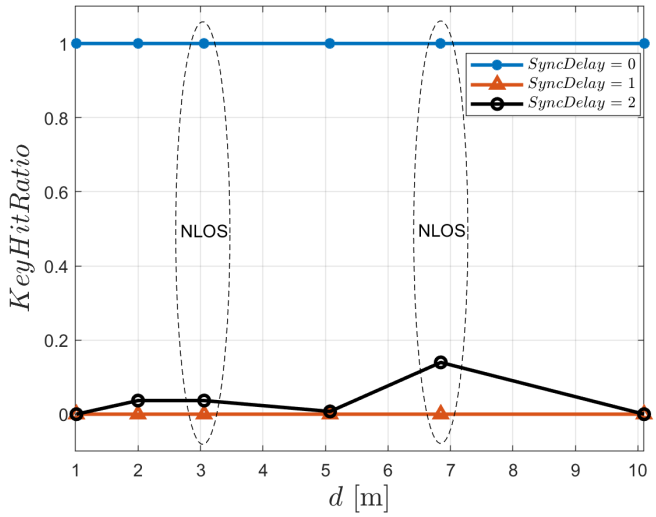


Figure 10. Time-domain key generation. *KeyHitRatio* with perfect channel reciprocity as a function of Alice-Bob distance. Configuration: $n_b = 2$ bits, $\Delta/q = 0.3$, $SyncDelay = 0, 1, 2$. Ellipses denote results obtained under non-line-of-sight conditions.

Focusing exclusively on ideal conditions—where key agreement is always successful—Table 2 reports the values of *AvKeyLength* for all the considered distances, ranging from approximately 260 to 780 bits. This variability is primarily influenced by the duration of the channel response and the number of censored samples, both of which may vary depending on the experienced propagation conditions.

Table 2. *AvKeyLength* with perfect channel reciprocity and time synchronization ($SyncDelay = 0$).

	1 m	**2 m**	**3 m**	**5 m**	**7 m**	**10 m**
$SyncDelay = 0$	339.5	260.3	433.3	264.4	778.0	441.9

Since these results are obtained under ideal conditions—namely, perfect channel reciprocity and time synchronization—they are particularly significant, as they represent the maximum average key lengths achievable with the measured channel responses using the key generation algorithm discussed herein. As such, they serve as a benchmark for the results presented in the following sections, which consider more realistic, non-ideal conditions and incorporate variants of the key generation algorithm specifically designed to address the challenges posed by channel asymmetry and synchronization errors.

Remark 3. *In practical experiments, it is indeed not possible to guarantee that the timing misalignment between Alice and Bob is an exact integer multiple of the sampling interval. Real systems inevitably experience fractional delays. However, this limitation does not compromise the validity of our results. Fractional delays simply produce performance levels that lie between those observed in the idealised cases where the delay is an integer number of sampling intervals. In other words, the integer-delay simulations reported in this paper should be interpreted as bounding cases, with fractional delays yielding intermediate behaviour.*

General Comment on Eve's Performance

In our setup, Eve was assumed to operate under highly favorable conditions: she used hardware identical to that of Alice and Bob, had full knowledge of the key-generation algorithm, and observed all public reconciliation messages exchanged between the legitimate users. Even under these assumptions, she consistently failed to reconstruct the

secret key. Specifically, *EveHitRatio* remained zero across all tested scenarios, including configurations that ranged from ideal conditions to realistic settings involving imperfect channel reciprocity and synchronization mismatches.

This outcome is primarily driven by the rapid spatial decorrelation of UWB channel responses in the industrial environment, where dense multipath and metallic obstructions greatly reduce spatial correlation. As a result, even when Eve was positioned as close as 0.986 m to Bob, she observed substantially different channels, rendering the physical-layer randomness exploited by Alice and Bob inaccessible to a passive eavesdropper.

It is worth emphasizing that further strengthening Eve, for instance by equipping her with additional computational power or multiple antennas, would not alter this conclusion. The limiting factor is not Eve's processing capability, but the fundamental spatial properties of the wideband radio channel in the considered environment.

Conversely, reducing Eve's distance from either Alice or Bob to less than one meter would require placing two PulsON 440 devices only a few centimeters apart. Such a configuration cannot be reliably reproduced in our setup, because at these extremely small separations the antennas operate in each other's near field, and mutual coupling and device housing effects dominate the measured response. These effects would not represent a realistic eavesdropping condition and would make the measured signals dependent on hardware interaction rather than on the propagation channel. Furthermore, the industrial workspace imposes practical constraints that prevent stable placement of two devices at centimeter-level spacing.

Since Eve proved unable to recover the correct key under all tested conditions in our industrial setup, her performance is no longer reported in the following sections.

6.2. Key Generation with Perfect Channel Reciprocity: Addressing Time Synchronization Issues

As shown in Figure 10, the lack of perfect time synchronization significantly impairs the ability of Alice and Bob to agree on the same key. This presents a significant challenge, as achieving the required level of time synchronization, given the rapid variation of channel responses with such wideband signals, is highly critical. A potential solution explored in this section involves extracting the key from the envelope of the channel response, which evolves more slowly than the channel response itself (see Section 3.1 and Figure 3).

The effectiveness of this strategy is illustrated in Figure 11, which shows how key generation performance improves despite time mismatches, under the same conditions as those considered in Figure 10. Specifically, now it can be observed that key matching becomes possible when $SyncDelay = 1$, with *KeyHitRatio* approaching 1 as the Alice–Bob distance increases beyond 3 m. However, the figure also reveals that when $SyncDelay = 2$, key matching becomes very difficult, indicating that the envelope-based key generation strategy provides only limited robustness against time-synchronization errors.

It is worth noting that, under the presence of a time offset, NLOS conditions sometimes exhibit higher key agreement than LOS. This behavior may appear counter-intuitive but is explained by the temporal structure of UWB channel responses. In LOS links, the channel response contains a strong, sharply localized first path; even a one-sample timing misalignment shifts the sampling window over a rapidly varying region of the waveform, substantially reducing the correlation between Alice's and Bob's observations. Conversely, in NLOS links the received energy is distributed over multiple diffuse multipath components, resulting in a smoother envelope with slower temporal variations. As a consequence, small timing errors have a less disruptive impact, and the two observations remain more correlated. This leads to a higher *KeyHitRatio* in NLOS scenarios when $SyncDelay \neq 0$.

For the sake of completeness, Table 3 reports the average key length achieved when $SyncDelay = 0$ and $SyncDelay = 1$.

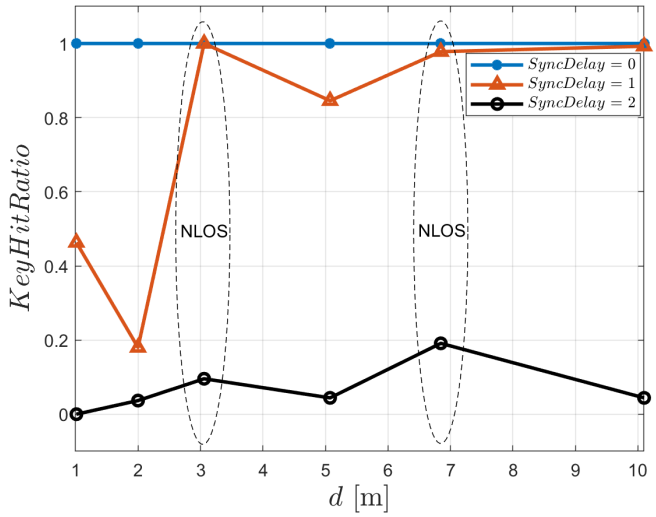


Figure 11. Envelope-based key generation. *KeyHitRatio* with perfect channel reciprocity as a function of Alice-Bob distance. Configuration: $n_b = 2$ bits, $\Delta/q = 0.3$, $SyncDelay = 0, 1, 2$. Ellipses denote results obtained under non-line-of-sight conditions.

Table 3. *AvKeyLength* with perfect channel reciprocity, time synchronization with $SyncDelay = 0$ and $SyncDelay = 1$.

	1 m	**2 m**	**3 m**	**5 m**	**7 m**	**10 m**
$SyncDelay = 0$	312.6	249.1	404.2	246.0	801.2	511.7
$SyncDelay = 1$	281.7	232.5	376.9	220.9	762.9	465.0

6.3. Key Generation with Non-Ideal Channel Reciprocity

Continuing our investigation into the performance of key generation based on measured channel responses, we now relax the assumption of perfect channel reciprocity. In our measurement setup, the signals exchanged over a given Alice-Bob link were collected sequentially, one after the other. This emulates a realistic scenario in which Alice first transmits a probe signal to Bob, who then responds by sending his own probe signal back to Alice. Although these two channel responses are likely to be well-correlated, they are not guaranteed to be identical. Environmental changes occurring between the two transmissions, even if subtle and imperceptible to human senses, can significantly affect the high-frequency and broadband signals used. As a result, the observed channel responses may differ, leading to a violation of the reciprocity condition. This is what we refer to in the following as the case of imperfect or non-ideal channel reciprocity.

Figure 12, when compared to Figure 11, highlights the degradation in *KeyHitRatio* that arises when channel reciprocity is no longer ensured. Still employing the envelope-based approach, the figure shows that, under perfect time synchronization ($SyncDelay = 0$), *KeyHitRatio* decreases by a factor ranging from approximately 50% to over 80% compared to the ideal case (Figure 11), as the distance increases. The figure also shows that, even under these non-ideal conditions, leveraging the envelope of the channel responses allows matched keys to be derived despite a time synchronization mismatch of one sampling interval ($SyncDelay = 1$). However, *KeyHitRatio* is reduced by approximately 70% in the best case ($d \approx 3$ m), and by more than 80% at all other distances, compared to the corresponding curve in Figure 11. As expected, a synchronization error greater than 1 sample renders the key generation algorithm ineffective, preventing any reliable key agreement. Table 4 shows the average key lengths achieved in this case when

$SyncDelay = 0$ and $SyncDelay = 1$. A reduction is evident in nearly all instances compared to those achieved under perfect channel reciprocity, as shown in Table 3.

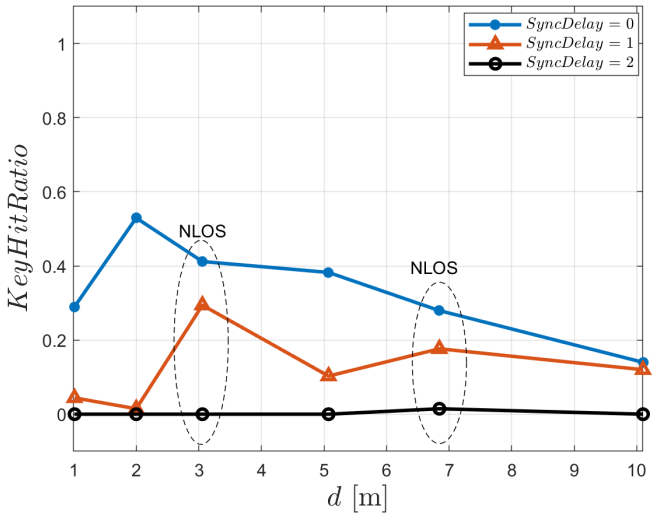


Figure 12. Envelope-based key generation. *KeyHitRatio* with non-ideal channel reciprocity as a function of Alice–Bob distance. Configuration: $n_b = 2$ bits, $\Delta/q = 0.3$, $SyncDelay = 0, 1, 2$. Ellipses denote results obtained under non-line-of-sight conditions.

Table 4. *AvKeyLength* with non-ideal channel reciprocity, time synchronization with $SyncDelay = 0$ and $SyncDelay = 1$.

	1 m	**2 m**	**3 m**	**5 m**	**7 m**	**10 m**
$SyncDelay = 0$	297.7	229.1	307.5	206.7	713.9	400.0
$SyncDelay = 1$	289.1	225.6	301.6	202.3	708.7	394.6

6.4. Addressing Time Synchronization Issues by Leveraging the Frequency-Domain

The results presented above indicate that, even when the key is generated from the envelope of the channel response, the key generation algorithm's performance deteriorates significantly once the timing mismatch between Alice and Bob exceeds a single sampling interval, rendering the algorithm practically ineffective for real-world applications. As discussed in Section 4.2, this issue can be addressed working in the frequency-domain, by deriving the secret key from the amplitude spectrum of the received signals.

The results shown in Figure 13 were obtained using this approach, assuming non-ideal channel reciprocity and synchronization mismatches of 0, 1, and 2 sampling intervals. As expected, *KeyHitRatio* is not affected by synchronization issues and the key generation process consistently achieves a success rate exceeding 80% across all considered distances. This confirms the robustness of the amplitude-spectrum-based approach to timing mismatches between Alice and Bob.

It is important to note that, under perfect synchronization, the frequency-domain approach yields lower key lengths and slightly reduced *KeyHitRatio* compared to the time-domain method. This is expected: the time-domain waveform (or its envelope) retains the full temporal structure of the UWB channel response, including sharp multipath components that provide high entropy when Alice and Bob observe identical signals. In contrast, the frequency-domain method relies solely on the magnitude of the FFT coefficients, discarding phase information and compressing the temporal richness of the channel response into a smoother frequency representation. As a result, less randomness is available for key extraction, even when synchronization is ideal. The benefit of this

approach becomes apparent only when time misalignment is present, where its robustness significantly outweighs the loss of entropy.

Table 5 reports the average key length for the considered approach, which remains nearly constant, ranging from 94.5 to 102.1 bits, as the distance between Alice and Bob increases. In this regard, it is interesting to compare Table 5 with Table 2, which serves as a benchmark since it reflects perfect channel reciprocity and ideal time synchronization. The comparison clearly shows that the fact that key generation is based solely on the amplitude spectrum, thus disregarding the phase information embedded in the received signal, results in a substantial reduction in the average key length. Nonetheless, it is worth noting that average key lengths on the order of 100 bits remain adequate for many practical communication systems.

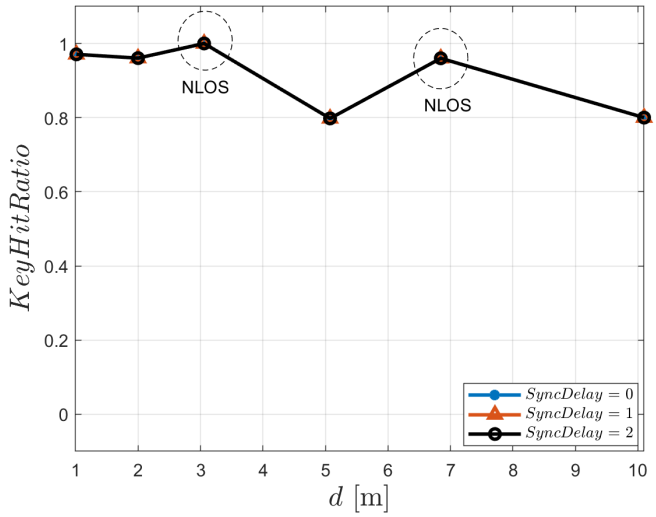


Figure 13. Frequency-domain key generation. *KeyHitRatio* with non-ideal channel reciprocity as a function of Alice–Bob distance. Configuration: $n_b = 2$ bits, $\Delta/q = 0.3$, $SyncDelay = 0, 1, 2$. Circles denote results obtained under non-line-of-sight conditions.

Table 5. Average secret key length obtained using the frequency-domain key generation algorithm under non-ideal channel reciprocity, for different Alice–Bob distances. Configuration: $n_b = 2$ bits and $\Delta/q = 0.3$.

1 m	2 m	3 m	5 m	7 m	10 m
102.1	97.6	100.8	94.5	98.0	95.3

6.5. The Impact of Re-Quantization on Frequency-Domain Key Generation

Still focusing on the frequency-domain approach outlined in Section 4.2, Figure 14 illustrates the impact of the quantization resolution parameter n_b on the key agreement performance, as a function of the distance between Alice and Bob, under non-ideal channel reciprocity conditions. Three configurations are considered: $n_b = 1$, $n_b = 2$ and $n_b = 3$.

As expected, the plot reveals that lower quantization resolutions result in more reliable key matching across all distances. Specifically, the case $n_b = 1$ achieves near-perfect key agreement ($KeyHitRatio \approx 1$) for all distances, indicating high robustness to non-ideal channel reciprocity. For $n_b = 2$, the key agreement remains generally high, although some degradation is observed, particularly at intermediate distances (e.g., 5 m). In contrast, the setting with $n_b = 3$ exhibits significantly lower and more variable *KeyHitRatio* values, dropping below 0.4 at larger distances.

These results should be interpreted in conjunction with the average key lengths achieved in the three considered configurations, as reported in Table 6. This joint evaluation

highlights that $n_b = 2$ offers a favorable trade-off between key agreement success rate and key length. While reducing n_b improves the likelihood of successful key matching, as lower resolution mitigates the impact of small mismatches between Alice and Bob's observations, it also limits the length of the generated key. Conversely, higher values of n_b increase key length but significantly reduce the success rate of key agreement, especially over longer distances. Therefore, the choice $n_b = 2$ balances robustness against channel imperfections with acceptable key generation efficiency.

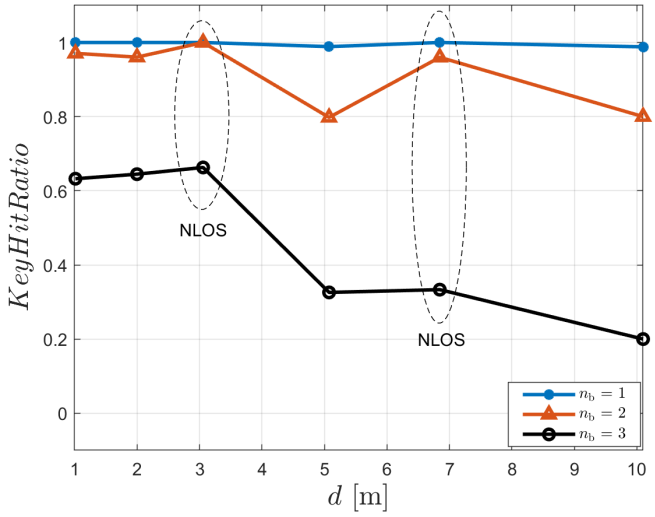


Figure 14. Frequency-domain key generation. *KeyHitRatio* with non-ideal channel reciprocity as a function of Alice–Bob distance. Configuration: n_b = 1, 2 and 3 bits, $\frac{\Delta}{q} = 0.3$.

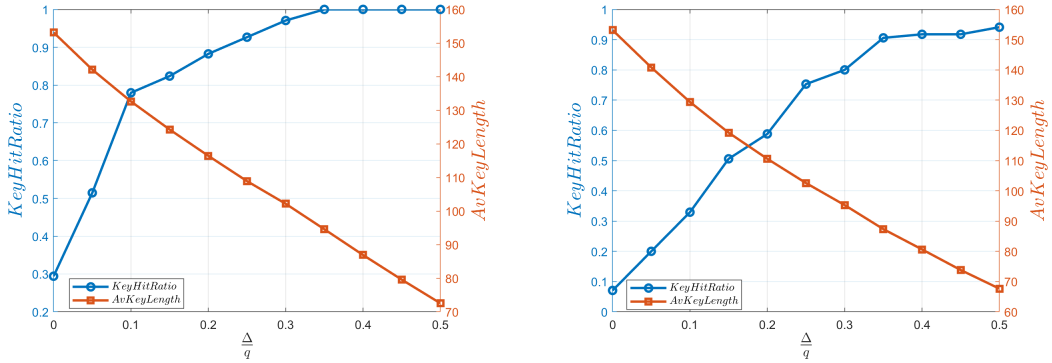
Table 6. Average secret key length obtained using the frequency-domain key generation algorithm for different quantization resolutions n_b, under non-ideal channel reciprocity. Results are reported for all Alice–Bob distances.

	1 m	**2 m**	**3 m**	**5 m**	**7 m**	**10 m**
$n_b = 1$	62.0	63.4	57.4	58.1	64.1	58.8
$n_b = 2$	102.1	97.6	100.8	94.5	98.0	95.3
$n_b = 3$	124.2	128.4	125.2	122.4	126.5	123.0

6.6. The Impact of Censored Region Width on Frequency-Domain Key Generation

Figure 15a illustrates the impact of the censored region relative width, expressed as $\frac{\Delta}{q}$, on the key agreement ratio and the average length of the generated secret key. The results refer to the frequency-domain approach and to a distance of 1 m between Alice and Bob. The two performance metrics are shown on dual axes: *KeyHitRatio* is plotted against the left y-axis, while *AvKeyLength* refers to the right y-axis.

As expected, increasing the relative width of the censored region leads to an improvement in *KeyHitRatio*. Starting from a relatively low value of approximately 0.3 at $\frac{\Delta}{q} = 0$, *KeyHitRatio* increases rapidly, exceeding 0.8 at $\frac{\Delta}{q} \approx 0.1$, and eventually saturates at 1.0 for $\frac{\Delta}{q} \geq 0.35$. This trend confirms that censoring values close to the re-quantization thresholds effectively reduces bit mismatches caused by minor discrepancies in the signals observed by Alice and Bob, thereby improving the reliability of key agreement.



(**a**) Configuration: $n_b = 2$ bits, $d = 1$ m. (**b**) Configuration: $n_b = 2$ bits, $d = 10$ m.

Figure 15. Frequency-domain key generation: effect of the censored-region relative width Δ/q on *KeyHitRatio* (left axis) and *AvKeyLength* (right axis). Results refer to non-ideal channel reciprocity with $n_b = 2$ bits, for Alice–Bob distances of (**a**) 1 m and (**b**) 10 m.

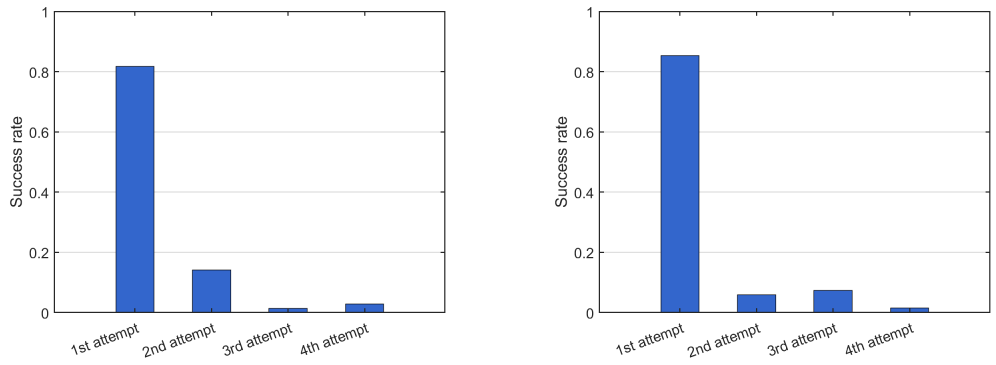
On the other hand, *AvKeyLength* exhibits a monotonic decrease with increasing $\frac{\Delta}{q}$. Starting from a peak value of approximately 155 bits when no censoring is applied ($\frac{\Delta}{q} = 0$), the average key length steadily declines, reaching about 70 bits at $\frac{\Delta}{q} = 0.5$. This behavior is consistent with the exclusion of an increasing number of signal samples that fall within the censored regions. As these samples are discarded from the re-quantization process, the amount of usable information decreases, leading to shorter secret keys.

Figure 15b, which corresponds to a distance of 10 m, shows similar trends. In this case, *KeyHitRatio* exceeds 0.8 for $\frac{\Delta}{q} > 0.3$, a threshold higher than that observed at $d = 1$ m.

These results highlight an inherent trade-off in physical-layer key generation: increasing key agreement reliability through more aggressive censoring reduces the effective key length. In the scenario considered, setting $\frac{\Delta}{q}$ around 0.3 emerges as a balanced compromise, offering high *KeyHitRatio* values while still maintaining a sufficiently long key for practical cryptographic use.

6.7. Key Agreement Delay and Key Disagreement Rate

Focusing on the case with $n_b = 2$ and $\frac{\Delta}{q} = 0.3$, we further analyzed how many times the procedure must be repeated before a common key is successfully established, following key agreement failures in previous attempts. In particular, we considered the scenarios where Alice and Bob are separated by $d = 5$ m and $d = 10$ m, which, as shown in Figure 14 (black plot), represent the most critical conditions for successful key establishment. In this context, Figure 16a, referring to the case $d = 5$ m, shows the success rate of the key agreement procedure as a function of the number of attempts. It is evident that key agreement is achieved on the first attempt in more than 80% of the cases, and in over 95% of the cases after in at most two attempts. Similar results were obtained for the case $d = 10$ m, as illustrated in Figure 16b: the success rate exceeds 85% on the first attempt and reaches 91% after at most two attempts. This behavior confirms that, although occasional agreement failures occur, the proposed procedure reliably achieves a common key within a limited number of iterations.



(**a**) Configuration: $n_{\mathrm{b}} = 2$ bits, $d = 5$ m. (**b**) Configuration: $n_{\mathrm{b}} = 2$ bits, $d = 10$ m.

Figure 16. Success probability of the key-agreement procedure as a function of the number of attempts. Results obtained for $n_b = 2$ and $\Delta/q = 0.3$, at distances of 5 m (**a**) and 10 m (**b**).

Finally, Figure 17 shows the key disagreement rate (KDR) as a function of Bob's distance from Alice. This metric is defined as the average ratio between the number of erroneous bits in a key and the total key length. It is evaluated prior to the *Information Reconciliation* step, thereby providing a baseline indicator of the performance required by the mismatch correction procedure adopted in this phase. Furthermore, the KDR reflects the reliability of the key generation process, since lower values correspond to a higher probability that Alice and Bob obtain identical keys. The results show that the KDR remains below 10^{-2} for all distances, except for $d = 5$ m and $d = 10$ m, where slightly higher values are observed. This confirms that, under most operating conditions, the key generation procedure exhibits a high level of reliability, requiring only limited error correction during reconciliation.

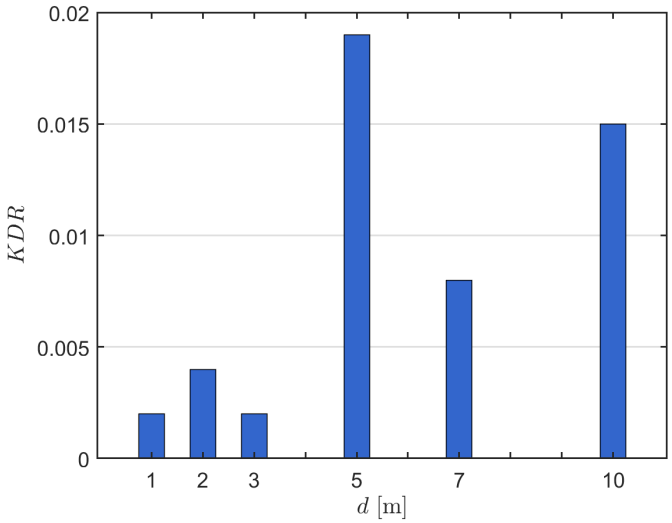


Figure 17. Key disagreement rate. Configuration: $n_{\mathrm{b}} = 2$ bits, $\Delta/q = 0.3$.

6.8. Computational Complexity

Focusing on the frequency-domain key-generation method, the most effective among the strategies considered, Table 7 summarizes the per-step time complexity and memory requirements for Alice and Bob. The main processing stages and their associated time complexities and memory requirements are outlined below:

- *ADC sampling and buffering*. These operations require linear-time processing of N_{xy} probe samples, with memory usage proportional to $N_{\mathrm{xy}} \cdot n_b^{(\mathrm{ADC})}$ bits [41].

- *fast Fourier transform computation.* The time complexity is $O(N_{xy} \log N_{xy})$, while data storage scales as $O(N_{xy})$, i.e., $O(N_{xy}\, n_b^{(ADC)})$ bits for $n_b^{(ADC)}$-bit samples [42].
- *Re-quantization and censoring.* These operations involve simple threshold comparisons and selective retention, resulting in a time complexity of $O(N_{xy})$. The memory footprint is dominated by the re-quantized stream, $O(N_{xy}\, n_b)$ bits.
- *Gray mapping.* This step involves a single XOR and bit shift per sample, yielding a time complexity of $O(N_{xy})$. The memory requirement is limited to storing input and output sequences, i.e., $O(N_{xy}\, n_b)$ [43].
- *Reconciliation at Alice's side.* For an (n, k) Hamming code, syndrome decoding provides an efficient means to determine, for each n-bit fragment, its coset and associated standard-array column (as requested by step 4, Section 4.1); the procedure is detailed in Appendix A. Among the steps described therein, the dominant computational cost is syndrome evaluation, which requires $O(n(n-k))$ bit operations per fragment. The memory footprint is likewise dominated by the storage of the parity-check matrix $\mathbf{H}$, resulting in a memory complexity of $O(n(n-k))$ bits.
- *Reconciliation at Bob's side.* The procedure executed by Bob in Step 4-Section 4.1 can be implemented efficiently using the algorithm described in Appendix B. In this workflow, the dominant runtime arises from fragment alignment, which is a single length-n bitwise operation and thus costs $O(n)$ per fragment. The memory footprint is likewise $O(n)$, driven by per-fragment working storage (e.g., the local fragment, the coset leader, and the aligned result).

Table 7. Time-complexity and memory-requirement summary for the main processing steps of the frequency-domain key-generation algorithm.

Operation	Time Complexity	Memory Requirement (bits)
Sampling & buffering	$O(N_{xy})$	$O(N_{xy}\, n_b^{(ADC)})$
fast Fourier transform	$O(N_{xy} \log N_{xy})$	$O(N_{xy}\, n_b^{(ADC)})$
Re-quantization & censoring	$O(N_{xy})$	$O(N_{xy}\, n_b)$
Gray mapping	$O(N_{xy})$	$O(N_{xy}\, n_b)$
Reconciliation at Alice	$O(n(n-k))$	$O(n(n-k))$
Reconciliation at Bob	$O(n)$	$O(n)$

6.9. Final Remarks

The results we obtained indicate that transitioning from ideal conditions—namely, perfect channel reciprocity and precise time synchronization—to more realistic scenarios leads to a reduction in the success probability of shared secret key generation, which can, however, be maintained over 80% when suitable strategies and parameter settings are adopted. This decrease is also accompanied by a notable reduction in the average key length, which, however, remains adequate for practical security applications. In particular, with a 80% *KeyHitRatio*, Alice and Bob can obtain an average secret key length of about 130 bits under non-ideal channel reciprocity at a short distance of $d = 1$ m (Figure 15a), and about 95 bits at $d = 10$ m (Figure 15b).

It is important to emphasize that the primary aim of this article was not to derive the maximum performance achievable by physical-layer key generation algorithms. In fact, we did not attempt to fine-tune the parameters of the algorithms we considered. Rather, the objective of this work was to highlight the inherent challenges and critical issues that arise when cryptographic key generation is applied to real-world signals. Despite these

difficulties, our aim was to demonstrate the practical feasibility of secure key generation at the physical layer, provided that appropriate strategies and countermeasures are adopted.

The industrial scenario was deliberately chosen due to the increasing importance of communication security in such environments. In particular, many industrial applications rely on low-cost, resource-constrained devices, such as sensors and embedded nodes, for which traditional cryptographic techniques may be impractical or inefficient due to their limited computational capabilities. In this context, physical-layer security mechanisms, including key generation from the wireless channel, offer a promising alternative, especially when tailored to the constraints and variability of real-world environments.

7. Conclusions

This paper presented an experimental evaluation of a physical-layer key generation scheme based on wireless channel measurements in an industrial environment, with a focus on real-world factors that affect its reliability and performance. By analyzing channel responses collected at various distances and under non-ideal conditions, such as imperfect channel reciprocity and time synchronization mismatches, we assessed the feasibility of generating symmetric cryptographic keys between two legitimate users while ensuring that an eavesdropper remains unable to infer them.

Our results showed that, when relying on time-domain observations of the received signals, even minor deviations from ideal conditions can substantially degrade key generation performance. In particular, the loss of channel reciprocity and synchronization mismatches were found to significantly reduce the key agreement rate. While envelope-based processing in the time-domain offers partial mitigation, it alone is insufficient to fully address these limitations. To overcome these challenges, we investigated a frequency-domain approach in which the secret key is extracted from the amplitude spectrum of the received signals. This strategy demonstrated significant improvements, enabling reliable key agreement even under adverse conditions. A detailed analysis across different parameter settings showed that average key lengths of about 100 bits can be achieved with a high success rate, making the technique viable for practical deployment.

In addition, the complexity analysis revealed no critical issues from a practical implementation standpoint. In summary, the results highlight the potential of physical-layer key generation as a lightweight and secure method for establishing symmetric keys in industrial internet-of-things wireless systems deployed in harsh environments.

Author Contributions: Conceptualization, L.M.A., S.C., L.M. and G.P.; methodology, L.M.A., S.C., L.M. and G.P.; software, L.M.A., S.C., L.M. and G.P.; validation, L.M.A., S.C., L.M. and G.P.; formal analysis, L.M.A. and G.P.; investigation, L.M.A., S.C., L.M. and G.P.; resources, L.M.A., S.C., L.M. and G.P.; data curation, S.C. and L.M.; writing—original draft preparation, L.M.A., S.C., L.M. and G.P.; writing—review and editing, L.M.A., S.C., L.M. and G.P.; visualization, L.M.A., S.C., L.M. and G.P.; supervision, L.M. and G.P.; project administration, L.M. and G.P.; funding acquisition, L.M. and G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the European Union—Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP J33C22002880001, partnership on "Telecommunications of the Future" (PE00000001-program "RESTART").

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to privacy restrictions of the company where the measurements have been set.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Syndrome-Based Information Reconciliation at Alice

During Step 4 of the Information Reconciliation procedure described in Section 4.1, Alice partitions the raw key into n-bit fragments and, for each fragment, determines the corresponding coset and column in the standard array of the (n, k) code $\mathcal{C}$. Rather than exhaustively searching the standard array, an equivalent and computationally efficient method leverages the syndrome calculation. The main steps are as follows:

- *Syndrome calculation.* Given an n-bit fragment r, Alice computes the syndrome $s = r\,\mathbf{H}^{\mathsf{T}}$, where $\mathbf{H} \in \{0,1\}^{(n-k)\times n}$ is the parity-check matrix of the (n, k) Hamming code; the syndrome s uniquely identifies the coset because all vectors in the same coset share the same syndrome.
- *Coset leader identification.* The syndrome s indexes a precomputed syndrome table that maps each syndrome to its coset leader e, chosen as a minimum-weight error pattern for that coset.
- *Codeword retrieval.* With e known, the corresponding codeword is obtained as $c = r - e \pmod 2$; this is the codeword at the head of the identified column in the standard array.
- *Key extraction.* The k information bits are extracted from c and appended to the reconciled key that Alice is constructing.

Appendix B. Information Reconciliation at Bob

Similarly to Alice, Bob also partitions his raw key into n-bit fragments. He then refines each fragment using the coset information broadcast by Alice. In this case as well, a computationally efficient procedure can be devised, avoiding any exhaustive search within the standard array. The description below assumes at most one bit of disagreement between Alice's and Bob's fragments, consistent with the error-correction capability of the considered Hamming code.

- *Coset leader acquisition.* Bob receives the coset index from Alice, which uniquely identifies the corresponding coset leader e.
- *Fragment alignment.* Let r' denote Bob's local n-bit fragment. Bob computes $r'' = r' - e \pmod 2$. Provided that Alice's fragment r and Bob's fragment r' differ by at most one bit, r'' lies in the same column of the standard array as r.
- *Key extraction.* Since r'' and r belong to the same column of the standard array, they correspond to the same reference codeword c. Bob extracts the k information bits associated with c and appends them to the reconciled key that he is constructing.

References

1. Behnke, I.; Austad, H. Real-time performance of industrial IoT communication technologies: A review. *IEEE Internet Things J.* **2023**, *11*, 7399–7410. [CrossRef]
2. Longhi, N.; Amorosa, L.M.; Cavallero, S.; Buracchini, E.; Verdone, R. 5G Architectures Enabling Remaining Useful Life Estimation for Industrial IoT: A Holistic Study. *IEEE Open J. Commun. Soc.* **2025**, *6*, 1016–1029. [CrossRef]
3. Avila, B.Y.L.; Vazquez, C.A.G.; Baluja, O.P.; Cotfas, D.T.; Cotfas, P.A. Energy harvesting techniques for wireless sensor networks: A systematic literature review. *Energy Strategy Rev.* **2025**, *57*, 101617. [CrossRef]
4. Radhakrishnan, I.; Jadon, S.; Honnavalli, P.B. Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices. *Sensors* **2024**, *24*, 4008. [CrossRef] [PubMed]
5. Diffie, W.; Hellman, M.E. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
6. Iera, A.; Floerkemeier, C.; Mitsugi, J.; Morabito, G. The Internet of things [guest editorial]. *IEEE Wirel. Commun.* **2010**, *17*, 8–9. [CrossRef]
7. Bernstein, D.J. Introduction to Post-Quantum Cryptography. In *Post-Quantum Cryptography*; Bernstein, D.J., Buchmann, J., Dahmen, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5299, pp. 1–14. [CrossRef]
8. Liang, Y.; Poor, H.V.; Shamai, S. Information theoretic security. *Found. Trends Commun. Inf. Theory* **2008**, *5*, 355–580. [CrossRef]
9. Shannon, C.E. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
10. Gamal, A.E.; Kim, Y.H. *Network Information Theory*; Cambridge University Press: Cambridge, UK, 2011.

11. Wyner, A.D. The wire-tap channel. *Bell Syst. Tech. J.* **1975**, *54*, 1334–1387. [CrossRef]
12. Ahlswede, R.; Csiszár, I. Common randomness in information theory and cryptography. Part I: Secret sharing. *IEEE Trans. Inf. Theory* **1993**, *39*, 1121–1132. [CrossRef]
13. Li, J.; Petropulu, A.P. On ergodic secrecy rate for Gaussian MISO wiretap channels. *IEEE Trans. Wirel. Commun.* **2011**, *10*, 1176–1187. [CrossRef]
14. Rabbachin, A.; Conti, A.; Win, M.Z. Wireless Network Intrinsic Secrecy. *IEEE/ACM Trans. Netw.* **2015**, *23*, 56–69. [CrossRef]
15. Ren, K.; Su, H.; Wang, Q. Secret key generation exploiting channel characteristics in wireless communications. *IEEE Wirel. Commun.* **2011**, *18*, 6–12. [CrossRef]
16. Maurer, U.M. Secret key agreement by public discussion from common information. *IEEE Trans. Inf. Theory* **1993**, *39*, 733–742. [CrossRef]
17. Hershey, J.E.; Hassan, A.A.; Yarlagadda, R. Unconventional cryptographic keying variable management. *IEEE Trans. Commun.* **1995**, *43*, 3–6. [CrossRef]
18. Pasolini, G.; Dardari, D. Secret Information of Wireless Multi-Dimensional Gaussian Channels. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 3429–3442. [CrossRef]
19. Pasolini, G.; Dardari, D.; Abreu, G.; Severi, S. The effect of channel spatial correlation on physical layer security in multi-antenna scenarios. In Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–6 November 2013; pp. 298–302. [CrossRef]
20. Patwari, N.; Croft, J.; Jana, S.; Kasera, S.K. High-Rate Uncorrelated Bit Extraction for Shared Secret Key Generation from Channel Measurements. *IEEE Trans. Mobile Comput.* **2010**, *9*, 17–30. [CrossRef]
21. Severi, S.; Abreu, G.; Pasolini, G.; Dardari, D. A Secret Key Exchange Scheme for Near Field Communication. In Proceedings of the IEEE Wireless Communications and Networking Conference, Istanbul, Turkey, 4–9 May 2014.
22. Shehadeh, Y.E.H.; Alfandi, O.; Hogrefe, D. Towards Robust Key Extraction from Multipath Wireless Channels. *J. Commun. Netw.* **2012**, *14*, 385–395. [CrossRef]
23. Wilson, R.; Tse, D.; Scholtz, R.A. Channel identification: Secret sharing using reciprocity in ultra-wideband channels. *IEEE Trans. Inf. Forens. Secur.* **2007**, *2*, 364–375. [CrossRef]
24. Madiseh, M.G.; He, S.; McGuire, M.L.; Neville, S.W.; Dong, X. Verification of secret key generation from UWB channel observations. In Proceedings of the IEEE International Conference on Communications , Dresden, Germany, 14–18 June 2009; pp. 1–5.
25. Marino, F.; Paolini, E.; Chiani, M. Secret key extraction from a UWB channel: Analysis in a real environment. In Proceedings of the 2014 IEEE International Conference on Ultra-WideBand (ICUWB), Paris, France, 1–3 September 2014; pp. 80–85. [CrossRef]
26. Zhang, J.; Rajendran, S.; Sun, Z.; Woods, R.; Hanzo, L. Physical Layer Security for the Internet of Things: Authentication and Key Generation. *IEEE Wirel. Commun.* **2019**, *26*, 92–98. [CrossRef]
27. Zheng, Y.; Dang, F.; Yang, Z.; Jiang, J.; Wang, X.; Wang, L.; Liu, K.; Chen, X.; Liu, Y. BlueKey: Exploiting Bluetooth Low Energy for Enhanced Physical-Layer Key Generation. In Proceedings of the IEEE INFOCOM 2024—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 20–23 May 2024; pp. 711–720. [CrossRef]
28. Wu, J.; Wu, H.; Chen, Y. A Robust Physical Layer Key Generation Algorithm for Ultra-Wideband Ranging Process. *IEEE Commun. Lett.* **2024**, *28*, 2478–2482. [CrossRef]
29. Boshoff, D.; Zhou, M.M.; Nkrow, R.E.; Silva, B.; Hancke, G.P. UWB Physical Layer Key Sharing Using the Frequency Domain CIR Magnitude. *IEEE Trans. Ind. Inform.* **2025**, *21*, 2000–2009. [CrossRef]
30. Kreiser, D.; Dyka, Z.; Kornemann, S.; Wittke, C.; Kabin, I.; Stecklina, O.; Langendorfer, P. On Wireless Channel Parameters for Key Generation in Industrial Environments. *IEEE Access* **2018**, *6*, 79010–79025. [CrossRef]
31. Dehmollaian, E.; Etzlinger, B.; Springer, A. A Lightweight CIR-Based Physical Layer Key Generation Scheme for UWB. In Proceedings of the 2024 IEEE Wireless Communications and Networking Conference (WCNC), Dubai, United Arab Emirates, 20–24 April 2024; pp. 1–6. [CrossRef]
32. Zhang, J.; Woods, R.; Duong, T.Q.; Marshall, A.; Ding, Y.; Huang, Y.; Xu, Q. Experimental Study on Key Generation for Physical Layer Security in Wireless Communications. *IEEE Access* **2016**, *4*, 4464–4477. [CrossRef]
33. Wan, Z.; Chu, Z.; Mi, D.; Wang, H.M.; Jin, L.; Huang, K. STAR-RIS-Assisted Physical-Layer Key Generation. *IEEE Trans. Veh. Technol.* **2024**, *73*, 9165–9170. [CrossRef]
34. Yang, X.; He, Y.; Liao, R.; Cai, Y.; Dai, W. Mission reliability-centered opportunistic maintenance approach for multistate manufacturing systems. *Reliab. Eng. Syst. Saf.* **2024**, *241*, 109693. [CrossRef]
35. Liao, R.; He, Y.; Feng, T.; Yang, X.; Dai, W.; Zhang, W. Mission reliability-driven risk-based predictive maintenance approach of multistate manufacturing system. *Reliab. Eng. Syst. Saf.* **2023**, *236*, 109273. [CrossRef]
36. Bloch, M.; Barros, J. *Physical-Layer Security. From Information Theory to Security Engineering*; Cambridge University Press: Cambridge, UK, 2011.

37. Pasolini, G.; Paolini, E.; Dardari, D.; Chiani, M. Experimental Results on Secret-Key Extraction from Unsynchronized UWB Channel Observations. In *Physical and Data-Link Security Techniques for Future Communication Systems*; Baldi, M., Tomasin, S., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 111–124.
38. Sonmez Turan, M. *Ascon-Based Lightweight Cryptography Standards for Constrained Devices*; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2025. [CrossRef]
39. Time Domain. *PulsON 440 Data Sheet & User Guide*; Time Domain, Inc., Cummings Research Park: Huntsville, AL, USA, 2015; Draft revision 320-0317A.
40. Time Domain. *Operation Description/ Theory of Operation*; Time Domain, Inc., Cummings Research Park: Huntsville, AL, USA, 2015; Draft Version 320-0317A.
41. Kester, W. *Data Conversion Handbook*; Newnes (Elsevier Imprint): Amsterdam, The Netherlands, 2005.
42. Rockmore, D.N. The FFT: An algorithm the whole family can use. *Comput. Sci. Eng.* **2002**, *2*, 60–64. [CrossRef]
43. Waggener, B.; Waggener, W.N. *Pulse Code Modulation Techniques*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1995.