Implementation and Analysis of a Key Generation Protocol for Wireless Sensor Networks

Matthias Stephan Wilhelm

Diploma Thesis

Implementation and Analysis of a Key Generation Protocol for Wireless Sensor Networks

vorgelegt von

Matthias Stephan Wilhelm

1. September 2009

Kaiserslautern University of Technology
Computer Science Department
- disco | Distributed Computer Systems Lab -
DA-00015

Supervisor: Prof. Dr.-Ing. Jens B. Schmitt
Tutor: Dr.-Ing. Ivan Martinovic

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben. Alle wörtlich oder sinngemäß übernommenen Zitate sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Kaiserslautern, den 1. September 2009

Matthias Stephan Wilhelm

Abstract

KEY management in wireless sensor networks faces the typical as well as several new challenges. The scale, resource limitations, and new threats such as node capture and compromise suggest the use of on-line key generation, where secret keys are generated by the nodes themselves after deployment. This approach can cope with changing and unknown topologies and scales well with the size of the network. Yet, the cost of such schemes is high since their secrecy is based on computational complexity. Recently, several research contributions justified that the wireless channel itself can be used to generate *information-theoretic* secure keys for two parties. By exchanging sampling messages between two network nodes during movement, a bit string can be derived using the physical properties of the wireless channel which can only be measured by the involved entities. But movement is not the only possibility to generate randomness, as the channel response is also strongly dependent on the frequency of the transmitted signal. In this thesis, the concept of key generation based on the *frequency-selectivity* of channel fading is introduced, and its applicability even for resource-limited devices is shown by implementation and empirical analysis in a real-world wireless sensor network. The great practical advantage of this approach is that the concept does not rely on node movement as a source of randomness. Thus, the common case of a sensor network with stationary motes is supported. Furthermore, the use of these stable channel properties enables us to mitigate the effects of measurement errors and other temporal effects, enabling a rate of successful key agreements of over 97% in the experiments. The analysis shows that the frequency-selectivity is strong enough to generate unpredictable random keys, and quantifies the trade-offs between secrecy and robustness, as well as the impact of the number of channels on the provided secrecy.

Contents

List of Figures

List of Tables

1 Introduction

The advances in computer technology enable new applications and uses of computer systems that were previously infeasible. The miniaturization of computing devices as well as advances in wireless networking enable new forms of computer networks, such as wireless sensor networks (WSNs). In this new class of network, a large number of low-cost sensing devices, the so-called *sensor motes*, collect environmental data such as temperature, pressure or movement in a distributed manner [1]. These devices combine sensing and computation, and acquired data can be processed by a sensor mote to achieve, for example, data aggregation or filtering based on the sensor readings. The data is transmitted to a central computer system, which analyzes the incoming information to infer the state of the monitored environment, using wireless communication. This transmission can require multiple hops, i.e., all sensor nodes must collaborate to achieve the goal of being a source of distributed information. The sensor motes themselves are generally not designed to do computational intensive tasks, therefore the data processing is centralized to a small number of special network nodes, the *data sinks*. By relying on wireless links, the deployment of sensor motes is simple, as the sensing devices can simply be placed at the best positions without worrying about the wiring of the network. This ease of deployment enables a number of new applications, both indoor and outdoor, such as home automatization, health care, or the inspection of the structural integrity of buildings. Outdoor applications for wireless sensor networks include battlefield surveillance and the remote monitoring of wildlife, traffic or even natural disasters such as forest fires. The low cost nature of sensor mote hardware drives the possibility to deploy such networks in a random fashion, that is, to form wireless networks with unknown topologies in an ad-hoc manner. Some common scenarios, such as forest fire monitoring, use uncontrolled deployment, for example, by simply throwing a large number of sensor motes out of a plane flying above the area of interest. This is only possible if the network is inherently redundant, i.e., separated sub-networks, destruction of sensor motes, or a loss of sensor readings must not be critical. Thus, the network must be able to build network topologies dynamically. And as the hardware is battery-powered in general, the possibility of node failures

must also be taken into account during the design phase of such a network to ensure a robust operation.

Security is a large concern in wireless sensor networks. The central entity must be able to ensure the authenticity and integrity of incoming transmissions to avoid an inconsistent view on the state of the environment and the network, influenced by an adversary. In case these transmissions are unchecked, it is easy for an adversary to insert incorrect sensor readings that benefit him, because the injection of messages into the broadcast wireless channel is easy. An injection can have severe consequences, e.g., large temperature readings that initiate emergency countermeasures (such as fire alarms), or in the case of motion detection networks used for battlefield surveillance, hiding the true movements in a mass of invalid readings. The sensor network provides a direct interface to the central computer system if no security countermeasures are in place. A second aspect to security are cases where the confidentiality of data must be enforced, i.e., if sensitive data that must not be disclosed to outsiders is transmitted. For example, a military motion detection network can provide information that also benefits the enemy, as every movement is monitored. Many security primitives and protocols have been developed to fulfill these requirements, but in the context of WSNs, most of these mechanisms are in need of an adjustment to adhere to the special requirements and constrains of such networks. Thus, wireless sensor network security established a branch in network security on its own right.

One of the major research problems on securing wireless sensor networks is *key management*. To enable legitimate notes to communicate securely, shared secret information is needed, which gives an advantage to the legitimate nodes over malicious eavesdroppers. Without such information, often referred to as *secret keys*, an attacker is unable to take the identity of members of the network, and cannot decipher messages encrypted with this key. The goal is therefore to provide this advantage to legitimate nodes in the network. However, the question is: How can secrets be established if an adversary can eavesdrop on every message exchange because the wireless channel is inherently public and broadcast in nature? Many solutions to this question are given, but most of them consider the case of wired networks, that is, networks consisting of computationally strong devices connected with links that can be physically controlled. One possibility of key management is the use of public key cryptography, as it makes distribution of keys easy. An eavesdropper cannot get an advantage from learning the public parameters, and therefore these can simply be transmitted publicly to other nodes to form secure communication links. Yet, for performance reasons, the use for symmetric cryptography is preferred in the context of WSNs, as public key protocols have a high demand with respect

to hardware resources. Therefore, key management is still an active field of research in the context of wireless sensor networks. This thesis proposes a novel key generation protocol for such networks, which can produce keys both securely and reliably, without depending on cryptographic algorithms. This protocol can serve as a new security primitive that makes key management easier in WSNs.

1.1 Key Management in Wireless Sensor Networks

Key management is concerned with the generation and distribution of secret keys, and their life-time administration, including key revocation and re-keying [5]. This topic is well researched in wired networks, and many protocols have been proposed, some of which are successfully applied in large-scale networks such as the Internet. For WSNs, no "one size fits all" approach has been found. A large number of protocols have been proposed that are finely tuned for certain scenarios. These approaches often rely on a large number of assumptions that limit their applicability, such as leveraging a certain factor that is only available in a small class of scenarios. An example of this are scenarios with fixed topologies, which eases the distribution of secret keys significantly. For surveys of current solutions and protocols with respect to WSNs, refer to [9, 56]. This abundance of available protocols exists because wireless sensor networks are highly heterogeneous in their scale, application requirements and needed security guarantees. Further, a number of additional factors make key management in wireless sensor networks particularly hard and incompatible to classic protocols from the wired world.

1.1.1 Particular Problems in Wireless Sensor Networks

Many aspects play a role when considering the problems and needs of key management in wireless sensor networks. The drive for low-cost manufacturing of sensor mote hardware requires the design to be as simple as possible. This drive also favors designs using standard components for general purpose devices instead of highly integrated solutions tailored for specific needs. Especially the CPU and memory configuration provide very limited resources (for example, the MICAz sensor mote platform used in the testbed has a 7.37 MHz micro-controller and 4 kB of RAM). This leads to resource utilization that is at their maximum capabilities even during normal operation, as a resource surplus for special tasks is wasted if the network is used differently. Unfortunately,

security services are not considered during the design phase of such networks, and are more likely introduced as an afterthought, which requires the security protocols to use a minimum of resources. A further aspect is that the sensor motes are battery-powered, and therefore complex computations should be avoided in any case, as those batteries cannot be replaced in most scenarios. Additionally, a limited battery life means a limited lifetime of the network as a whole, and as individual sensor motes can be targeted with power depletion attacks if they use power-consuming security protocols, such complex security protocols can also introduce new weaknesses. These new attack vectors against the performance of the system must also be considered to measure the overall security of a network. Flooding a single node of the network can lead to a separation of the network, which in turn can disrupt the complete operation. Thus, the protocol must be designed to avoid denial of service (DoS) vulnerabilities, i.e., the design must also be performance-aware.

Another major factor is the scale that such networks can reach. As the number of per-link keys that must be stored increases quadratically with the number of nodes, key management must find new ways to ensure secure connectivity. This is not an issue when the nodes are deployed in a planned way, but with random deployment, the topology is unknown *a priori* and in consequence, all nodes should be able to form secure connections with every other node. But the lack of fixed infrastructure is not the only concern that WSNs raise in this context. A second possibility is a changing topology, e.g., if a sensor mote moves from one to another region of the network. New connections have to be established in an ad-hoc fashion, and messages must take new routes to the data sinks. It is infeasible to guarantee that two arbitrary nodes always share a secret. To further increase the problems key management faces, the set of nodes is not fixed over the lifetime of the network as nodes leave due to hardware failure or battery depletion, and new nodes join when sleeping nodes become active or new nodes are deployed to restock the network. And finally, many of these networks are unattended, and an attacker can gain physical access to the hardware and read the physical memory of sensor motes, which results in the so-called node capture attack [21]. Alternatively, he can place his own sensor nodes to monitor the communication or inject messages, if no security countermeasures against impersonation are in place.

1.1.2 Key Management Protocols

The approaches to place secret keys on individual sensor nodes can be classified in two categories: key distribution and on-line key generation. In the first class, the keys are generated in a cryptographically secure manner by a central

entity, such as the manufacturer of the sensor motes, and stored directly in the memory of the devices. Some protocols store only a random subset of all possible keys on each sensor mote and ensure the availability of shared keys probabilistically. The next step in the protocols range from a direct usage of the pre-distributed keys (e.g., in case of a planned node deployment), to probabilistic methods where neighboring nodes have to agree on common keys from a set of pre-distributed keys [32, 58]. As the sensor motes are not able to store the keys for every possible connection as the number of keys grows too fast, only a subset of the keys is stored, and the protocol only guarantee that sensor nodes can find common keys with a sufficiently high probability. These random schemes can limit the impact of node captures because in this case only a subset of keys is leaked. Alternatively, the keys can be distributed after deployment using secure physical connections (e.g., in [28]), but this approach does not scale well with an increasing number of nodes.

The second class of key management protocols suits the needs of wireless sensor networks better: the on-line generation of secret keys after network deployment. As the topology is known locally to the network nodes at that time (the nodes can broadcast discovery messages to initiated contacts with the surrounding nodes), only the necessary keys must be generated. This promises to be a very efficient and scalable approach to wireless sensor network key management and is therefore explored further in this thesis.

1.1.3 On-line Key Generation

The need for a central key management authority can be avoided if the nodes themselves can create secret keys on-line and on-demand. This approach supports a large number of nodes easily, even in unknown topologies, as only the required keys must be generated for links to sensor motes in the neighborhood. The approach scales well with the number of nodes, as no central managing entity (which also constituted the single point of failure in the network) is necessary. Even if an attacker has physical access to the network, his attack will have only local impact because only the links involving the captured node are compromised. This is important, as, in general, WSNs are considered to be deployed in hostile environments.

Yet, common practice proves that the best way to provide sensor motes with secret keys is key (pre-) distribution, even with such extreme examples as the use of a single key to secure a whole network. This stems for the fact that the wireless channel is public, that is, all messages are broadcast, which makes the transmission of keys after deployment problematic. There is no authenticated channel available, because this again requires mutual secret information to be

available. Adversaries can therefore eavesdrop on the key agreement messages on the wireless link because both nodes must agree on the secret publicly and thus have to communicate. This leads to a leakage of secret information if there is no sufficient complexity hiding this information from an attacker. Some authors argue that eavesdropping on a large number of network nodes adds complexity by itself (for example, in [2], the keys are transmitted as plain text), but again, this applies only to limited scenarios.

The cryptographic solution to this problem is public key cryptography, most notably the Diffie-Hellman key agreement protocols. By relying on computational complexity, it is infeasible for an adversary with bounded computational capabilities to break the secrecy of the protocol. One important assumption in this context is the *decisional Diffie-Hellman assumption* (DDH) [8], which gives a statement on the computational hardness of certain problems involving discrete logarithms in cyclic groups. In such protocols, both parties generate a private and a public part of a secret, and transmit only the public part to the other party. An adversary who has knowledge of the two public parts only cannot infer the shared secret, even though he has sufficient information, but he is unable to use this information because of its computational complexity. The required complexity against computationally strong attackers dictates long key lengths, which results in large complexities for the legitimate parties as well. And although there are efforts to adopt public key cryptographic protocols to the world of WSNs (e.g., TinyECC [31]), these adaptions often offer only degraded security or have a significant time and memory footprint [52]. As an example, TinyECC (with optimizations) requires roughly 20 kB of ROM and 1.7 kB of RAM, which is 15.6% respectively 42.5% of the overall available memory resources of MICAz sensor motes, and single operations require a computation time in the order of seconds.

Thus, the traditional approaches are not the best choice for efficient on-line generation of secret keys in WSNs. In the last years, the use of physical characteristics to generate and distribute secret information have been explored in other contexts. The idea of quantum key agreements that cannot be eavesdropped shows that the laws of physics can stand against an attacker [41]. Therein, one party generates a random number and sends it to the other party using a quantum channel that cannot be eavesdropped. The secrecy is therefore not based on unproven assumptions such as DDH, but is provided to *any* adversary, even a computationally unbounded one. This is referred to as *information-theoretic* security (as introduced by C. E. Shannon in [49]).

A further approach is to use physical sources of correlated information that are conditioned to physical parameters such as the position. These random sources form correlated random processes, which are shared between two par-

ties and de-correlate rapidly if the physical parameter is changed. In this way, the output of the random process is unknown to anyone at different physical positions, and can therefore be used as a shared secret. One such source is the wireless channel because physical effects such as multipath fading make its behavior hard to predict. This thesis explores the possibility to generate strong secret keys, even with the limited hardware capabilities of sensor motes, based on this principle. This way, simply by monitoring the state of the wireless channel, secrets can be derived and the need for complex calculations and cryptographic algorithms in memory can be avoided. Put differently, we can trade computation with communication, which is generally accessible for WSN hardware, it is even on-par with the communication capabilities available for laptops for the WSN standard. Because the resulting secret is information-theoretic secure, the gap between the hardware of adversaries and sensor motes is closed. This thesis sets out to achieve this goal by designing a key generation protocol based on wireless channel properties.

1.2 Outline

The thesis describes the way to a strong shared secret, based solemnly on the possibility to monitor the state of the wireless channel between two parties. The structure of this thesis is as follows:

Chapter 2: In the next chapter, the source for secure randomness we aim employ, the wireless channel, is discussed with its general characteristics. Especially the multipath fading behavior that enables this key generation protocol is presented in detail. A reliable access to the channel state, which results in a key generation concept that can be used on common hardware, is developed. Then, the unpredictability of the wireless channel, the source of secrecy, is discussed, and the usability of physical states for such an end is evaluated using an analytical, information theoretic framework to quantify the uncertainty that an eavesdropper experiences. A strong adversary model is developed to analyze the secrecy, and to motivate that the concept is capable to support strong secrecy requirements of security services.

Chapter 3: Here, the way from a collection of channel samples to a shared secret is presented in detail. The design of the building blocks of the protocol is described, along with a stochastic model that allows the analysis of the secrecy in the analytical framework. The protocol is designed to be usable even on the most resource-limited devices by shifting the

complexity from computation to communication. The robustness against sampling errors is the most important aspect when generating secret keys, as a single error makes the key unusable. By the use of error-correcting codes from information theory, this robustness can be achieved, to a degree that successful key agreements can even be guaranteed. The protocol, as well as the parameter choices that influence the security and robustness, are presented in this chapter.

Chapter 4: This chapter summarizes the quantitative results of the performance and secrecy that were acquired from a number of experiments using the MICAz wireless sensor platform. The implementation and testbed used for the experiments are also described. For the experimental analysis, the deviations between measurements of two parties are analyzed in order to show that the use of channel characteristics is justified, even on low-cost hardware which is not designed to measure the channel state with high precision. The measurements are accurate, with bounded errors, and a high rate of successful key agreements is shown by the experiments. This shows the applicability of the key generation protocol to wireless sensor mote hardware by experiments in real-world scenarios. Further, the secrecy of the resulting shared string from different experiments is quantified using several methods. The experiments show that strong secrets can indeed be generated on current sensor network platforms.

Chapter 5: Further analysis, which models the secrecy in a device independent manner, is presented in this chapter. An analytical model is proposed that captures both the secrecy capacity and the correlation structure of the empirical data. Thus, by relying on the underlying distributions of wireless propagation, a general statement can be made on the performance and secrecy of the proposed protocol in all scenarios. The model is validated against the empirical data, and is shown to enable predictions on the possible improvements when hardware capabilities are increased in the future.

Chapter 6: This chapter provides a discussion of the uses of the generated secrets to increase the security of WSNs and to support security services. Further, different technologies are evaluated for their properties with respect to the presented protocol. Experiments using the well-known IEEE 802.11 standard, as well as considerations for software-defined radios are presented in this chapter.

Chapter 7: Finally, the last chapter concludes this thesis and gives an outlook on future work on this topic.

1.2.1 Thesis Goals

The goal of this thesis is to design a key agreement protocol that uses the physical properties of the wireless channel to generate secret keys. This enables on-line key agreement in wireless sensor networks, which relieves the burden both in time (computation) and space (memory) from the sensor motes. The protocol must be lightweight enough to be supported by all kinds of current sensor mote hardware. Also, it must be usable in *all* WSN scenarios, especially in the common case, where the sensor motes are stationary after deployment. Of course, the generated keys must be unpredictable, even when an eavesdropper monitors the sampling messages and can estimate the positions of the participating nodes. Only then are the keys usable. Further, because the protocol relies on measurements of physical properties, the robustness against errors must be sufficiently high to ensure that keys are generated with a high probability, i.e., the key generation is successful. This means that the natural deviations in the wireless channel, as well as deviations in the measurements due to systematic errors, must be overcome reliably.

1.3 Related Work

The problem of key generation has been considered under different aspects. In this section, several different perspectives on the same problem complex are discussed. Quantum computing researchers were the first to consider the use of correlated random variables to derive secret strings in practical scenarios.

1.3.1 Quantum Key Distribution

By relying on the properties of quantum mechanics, two parties can generate a shared random bit string. By exchanging physical signals, such as photons that are polarized in two different modes, a string of bits can be transmitted (quantum based protocols are described, e.g., in [17, 6]). Any attempt to eavesdrop on this string during key distribution leads to a disturbance in the system, which can be detected by the legitimate parties with a high probability. The secrecy is therefore based on the physical principles of quantum mechanics which state that any measurement of an unknown quantum state alters this state involuntarily. In the photon example, if an eavesdropper applies a filter

with the wrong polarization, the photon is blocked and the man in the middle is revealed. Thus, as an eavesdropper must be able to predict the key in order to avoid detection, this becomes infeasible with an increasing number of bits, and he his actions will be detected.

Due to the nature of the quantum channel, deviations in the bit string between the two parties are possible. In this context, the notion of privacy amplification and information reconciliation is crucial [6]. Privacy amplification guarantees that even when an eavesdropper has partial information on the secret string (if, e.g., the act of eavesdropping remains undetected), the resulting string is fully secret with a high possibility. Constructions that can be used for privacy amplification are discussed in Chapter 6. Information reconciliation is used to repair errors in the shared bit strings of the legitimate parties by exchanging reconciliation messages, which offer a minimum of information to an adversary. This method can be used to design a large number of key agreement protocols that employ shared randomness. An overview of current directions to apply quantum key generation and distribution to a general setting can be found in [11].

1.3.2 Directly Related Work

Several publications explore the applicability of physical characteristics of wireless signal propagation to key generation. Directly related to the work in this thesis is the use of the wireless channels characteristics of *narrow-band systems*, which are currently the most common systems in computer networking. Examples for such systems are IEEE 802.11 networks (also referred to as wireless LAN), as well as personal area network (PAN) specifications such as Bluetooth and ZigBee (IEEE 802.15.4), which are also used for communication in wireless sensor networks.

Mathur *et al.* [37] propose a key generation protocol that uses the unpredictability of the wireless channel response, which is introduced by device mobility. The introduced "radio-telepathy" enables two parties to derive secret keys from the wireless channel. It is sampled with a high rate to form a sequence of channel state measurements. These measurements are correlated for the legitimate parties, but due to the physical effects described in detail in the next chapter, this sequence is unpredictable for an eavesdropper, even if he can monitor the sampling signals and knows the location of the nodes involved in the protocol and the propagation environment. This stems from the fact that the channel response de-correlates quickly with increasing distance. As this raw sequence of samples is unlikely to be exactly equal for the legitimate parties due to constrains in the measurement process and the instability of the

wireless channel, a level-crossing algorithm is used to ensure that both sides generate the same bit string. The proposed algorithm uses two thresholds, an upper threshold which is crossed if signals are stronger than average, and likewise for the lower threshold. When a certain number of consecutive samples in the sequence exceed one of those thresholds, a bit is appended to the resulting secret string, a "1" if the signal was strong for this period of time and exceeded the upper threshold, and a "0" if the lower threshold was traversed. Doing so over the complete sequence results in a shared secret generated from the public channel. This method can ensure that both strings are equal, as longer necessary excursions are be required, which is more likely to be equal for both parties.

The authors make an analytical argument for the secrecy of the protocol and show, by a real-world implementation using specialized IEEE 802.11 hardware as well as stock wireless devices in laptops, that this approach is usable. But some problematic aspects are not addressed in this work. The obtained secret strings are only evaluated for their randomness, but randomness does not imply secrecy. If an adversary can monitor the environment and is able to observe the pattern of movement, he might be able to predict bits, which then do not add to the secrecy of the combined secret. Although this fact does not destroy the security of the protocol as a whole if proper precautions are made, the estimated amount of secrecy can be too optimistic. The practical usefulness of the protocol is still limited as well, as a very high sampling rate must be used to extract the full randomness induced by movement from the wireless channel. This is because level-crossing is a rare event (in the order of Hz), especially with conservative parameter settings required for a high success ratio. This limits the rate of secret bits to a maximum value of 15 secret bits per second[1]. The implementation the authors describe is able to generate 1 secret bit per second, as errors in the bit string must be avoided at all cost. This leads to the problem that a large number of samples must be discarded, even if they are equal on both sides. This error correction mechanism requires both computational time and memory space, and the benefits of this approach are therefore reduced or even voided. As laptops can generate keys in the order of milliseconds, a protocol that requires movement for key generations with a duration of minutes, this protocol is of theoretical interest only. Still, this interesting concept of radio-telepathy is appealing if these problems can be overcome.

[1]This depends on the speed of movement, this example is for a typical indoor speed of 1 m/s.

Azimi-Sadjadi *et al.* [4] introduce a key generation protocol that is focused on robustness, that is, the goal is to generate secret keys that are equal for both legitimate parties with a very high probability. State information from the channel is gathered similarly to the protocol described previously, but here only negative excursions are considered. This reduces the impact of possible concurrent transmissions that disturb the measurements, and ensures that both parties are able to measure the same excursions. If the sampled sequence is above the threshold, a "1" is appended to the shared string for this sample, and if a deep-fade is detected, a "0" is appended. This leads to a string that consists of large groups of "1"s interleaved by smaller groups of "0"s. As the beginnings of these groups are likely to be shifted by a few samples (these samples are gathered with slight time shifts), the deviations are repaired by an error correction mechanism. The resulting string is not usable as keying material directly because most parts are easily predictable by an adversary. Not every bit in the bit string carries a bit of secret information, but only a small fraction of a bit. In order to generate strings suitable as secret keys, randomness extraction is used. This information theoretic tool generates shorter strings, which cannot be distinguished from strings coming from a uniform random source, on the input of long strings from arbitrary distributions. Yet, this property comes with a steep cost in terms of bits that are lost, and the authors are not addressing this issue. A description of randomness extractors is given in Chapter 6.

The authors provide an analysis for Rayleigh fading channels, and a simulation based on this channel model. The work lacks quantitative results to make the protocols performance comparable. But as deep fades are rare events, the number of secret bits is likely to be bounded to a few secret bits per second. Together with the use of randomness extractors and error correction, the number of secret bits per second is approximately only a fraction of a bit. By relying on deep-fades only, the largest share of secret information is discarded.

As pointed out, some important aspects still remain to be solved regarding on-line key generation. The use of low-power and low-cost hardware that benefits most from on-line key generation is not evaluated, and the protocols are not designed for this important purpose. The need for movement or special hardware must also be overcome to enable realistic scenarios and applications for such key generation protocols. Currently, no contribution has been made to apply the approach of physical generation of secrets to wireless sensor networks.

1.3.3 Related Work using Different Technologies

Narrow-band systems are the most common in current computer networks, but upcoming technologies such as ultra-wideband (UWB) or the use of special-purpose hardware enables researchers to get better access to channel state information and design new types of key generation protocols based on physical properties.

Aono *et al.* [3] make use of an electronically steerable parasitic array radiation (ESPAR) antenna with controllable transmission characteristics. By the use of beam forming, the fluctuation in the channel characteristics can be increased to generate a larger amount of randomness. The authors propose a RSSI based key generation scheme, which uses a single threshold for binary quantization. The special antenna used make this approach unusable in wireless sensor networks, as it is not needed in this context and only increases the hardware cost.

Wilson *et al.* [53] extract secret strings from UWB channel pulses. These pulses have durations in the order of nanoseconds, which makes it feasible to detect echos reflected by objects in the path of the signal. By leveraging the ability to distinguish individual multipath components, the number of possible secret bits in a single probing is quite high. The authors only provide a simulation based on experimental data and show that such a scheme is feasible, but the extraction process is prone to errors and requires very strong signals to work reliably.

An early contribution addressing the use of physical properties is given in [22]. The authors suggest the use of phase shifts to generate secret keys. The drawback of this approach is that this information is hard to acquire. Current systems report the strength of signals and quality of links, but phase information requires a precise sampling of the waveform, which is not necessary in normal applications and is not likely to be provided by vendors of sensor mote hardware in the near future.

1.3.4 Theoretic Results

The theoretic foundation to the practical results presented above can be found in information theory literature.

Maurer considered the usability of correlated randomness to allow key agreements based on common information on public channels [38, 39, 40]. The notion of correlated random variables is used to describe the mutual information between legitimate parties, but also information leakage to an eavesdropper. His work includes the proof that such schemes can be used to generate unbreak-

able keys, the bounds and limits of such generation protocols and theoretic constructions that can provide secrecy. No practical implementations are considered, only the principles of advantage distillation, information reconciliation and privacy amplification are introduced to construct such systems.

A different approach was suggested by Wyner, with generalizations to the broadcast channel by Csiszár and Körner: the wiretap channel [54, 15]. Two legitimate parties communicate over a public channel with given bit error rate (BER). The results show that even if an attacker has access to the channel and can eavesdrop on the messages, it is possible to achieve perfect secrecy for the legitimate nodes if the attacker has only access to a degraded version of the channel, i.e., his BER is higher. The rate of the secret communication depends on the channel quality of the adversary, which is difficult to quantify correctly, as an attacker can increase his quality of reception, for example, by the use of high gain antennas.

Finally, the work of Dodis *et al.* [16] provides tools to derive secret information from strings that are partially known to an adversary, known as strong fuzzy extractors. This is useful in the context of biometric data, as, e.g., fingerprints are not completely unpredictable. This work provide a means to produce equal secret keys, even when the input varies slightly. This is also important because the measurement process for biometric data not always results in the same values, but values with slight deviations.

2 Concept

THIS chapter provides an introduction to the concept of key generation using wireless channel properties. It describes the nature of wireless channel propagation and its properties that generate uncertainty to an adversary. The secrecy of such schemes is explored and a possibility to quantify the amount of uncertainty in the channel is presented, relating to a realistic attacker model.

2.1 Wireless Channel Properties

Wireless communication uses electromagnetic waves to transmit information to other network nodes. In the narrow-band systems considered in this work, the signal representation of a message is band-limited, for example, to a bandwidth of 2 MHz in the IEEE 802.15.4 standard. Messages are encoded and then modulated onto a sinusoid carrier wave of a chosen frequency [46]. The receiver also tunes to this frequency, demodulates and decodes the transmitted signal to recover the original message. The main characteristics of such waves are the amplitude, the phase and the frequency. The frequency must be known to both parties that want to communicate, and therefore must be defined in a communication standard. In the case of most sensor networks, the ISM band in the 2.4 GHz range is used, but several other frequency bands are open for public use as well. The amplitude directly corresponds to the strength of the signal that is transmitted, and the larger the amplitude, the easier it is for a receiver to detect and demodulate the signal. The majority of the signal's power is contained in the carrier wave for the common phase shift modulations, that is, the received signal strength can be assumed to be independent from the message content in the WSN context. The phase of the signal refers to the shift of the sinusoid wave from its initial position.

When signals are transmitted over a wireless channel, these characteristics are affected, depending on the properties of the channel response. An example for the effects of wireless signal propagation is given in Figure 2.1. The sender transmits a cosine wave with amplitude A and phase shift $\theta = 0$, using a carrier frequency of f_c, that is, $s(t) = A\cos(2\pi f_c t)$. After transmission over the wireless channel, a changed version of this signal is received. The signal is

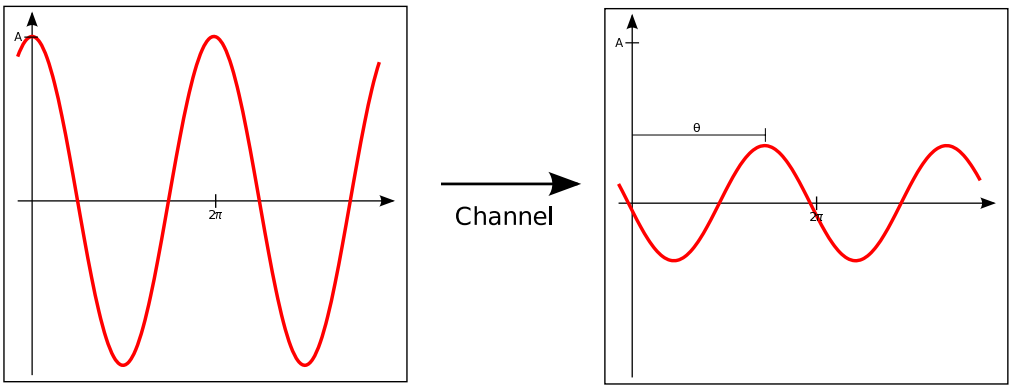


Figure 2.1: Signal transformations by the wireless channel, resulting in attenuation and phase shift at the receiver.

attenuated, i.e., the amplitude of the signal is reduced, resulting in a weaker received signal strength at the receiver. The phase is affected as well by the transmission, the signal is shifted by the effects in the channel to a different phase offset $\theta = \arg h(t)$, with $h(t)$ being the complex channel response. The resulting signal has the form $\hat{s}(t) = |h(t)|\, A \cos(2\pi f_c t + \theta)$, which shows the impact on the amplitude and the phase. Both of these effects are strongly depending on the environment the signal travels through. The understanding of these effects is important to apply its properties to security purposes.

2.1.1 Attenuation of Wireless Signals

As we have seen, the amplitude of the transmitted signal is altered as it travels through physical space. Normally the signal is attenuated, and we refer to this as *fading*. Three main factors have been identified to contribute to fading in wireless transmission systems: path loss, shadowing, and multipath fading [46].

Path loss depends on the distance between transmitter and receiver. As the transmitted energy is distributed across the surface of a sphere in omnidirectional communication, the larger the distance and therefore the radius of this sphere, the larger is the surface over which the energy is distributed. Therefore the energy density drops quadratically, which leads to a loss in signal strength, that is, this phenomenon contributes to the reduction of the signal's amplitude. The received signal strength, based on the distance d between two nodes, can be described with the formula (measured in decibel):

$$PL(d) = PL_{d_0} + 10\eta \log_{10}\left(\frac{d}{d_0}\right).$$

d_o denotes the reference distance, this model is only applicable if the distance d is greater than this value. The path loss of the considered system for the reference distance d_0, PL_{d_0}, must be determined empirically. The path loss exponent η determines the amount of signal diminishment. This exponent depends on the environment and must also be determined experimentally. Common values for this exponent are in the range of 2 to 5 for indoor scenarios. As the only parameter of this model is the distance d, a simple logarithmic relationship between the distance and the received signal strength is the result, independent from the nature and layout of the environment. This is clearly not the case for real-world propagation of wireless channels, therefore, this model only provides a very coarse approximation, for example, to evaluate transmission ranges in free space.

Shadowing (also known as the physical phenomenon of *refraction*) refers to the loss in signal strength when solid objects in the path of the signal are passed through. Depending on the material properties and form of the object, the signal is strongly attenuated. If no objects are in the direct path between sender and receiver, we refer to this as a *line-of-sight* (LOS) *connection*, otherwise as a *non-LOS connection*. If the attenuation is measured at different positions using the same distance d between both parties, then the attenuation is random if shadowing is considered. Empirical studies proposed a model named Log-Normal shadowing [46] to capture both the path loss and shadowing component, using the formula

$$LN(d) = PL(d) + \xi,$$

with the zero-mean Gaussian (Normal) random variable ξ with the variance σ^2. σ is typically chosen as 5 or 8, depending on the amount of shadowing in the environment. The presence of shadowing therefore introduces some uncertainty in the attenuation experienced by two nodes.

But the most interesting channel property in the context of key generation using physical properties is *multipath fading* as a source of randomness.

2.1.2 Multipath Fading

Electromagnetic waves are affected by the laws of geometric optics. A signal can travel through multiple paths and arrive at the same position from different directions. To simplify the analysis of such behavior, the waves are abstracted to linear rays that travel through space. As with light, the direction and intensity of these rays can be influenced by effects such as reflection, diffraction and scattering, which affect the paths of these rays.

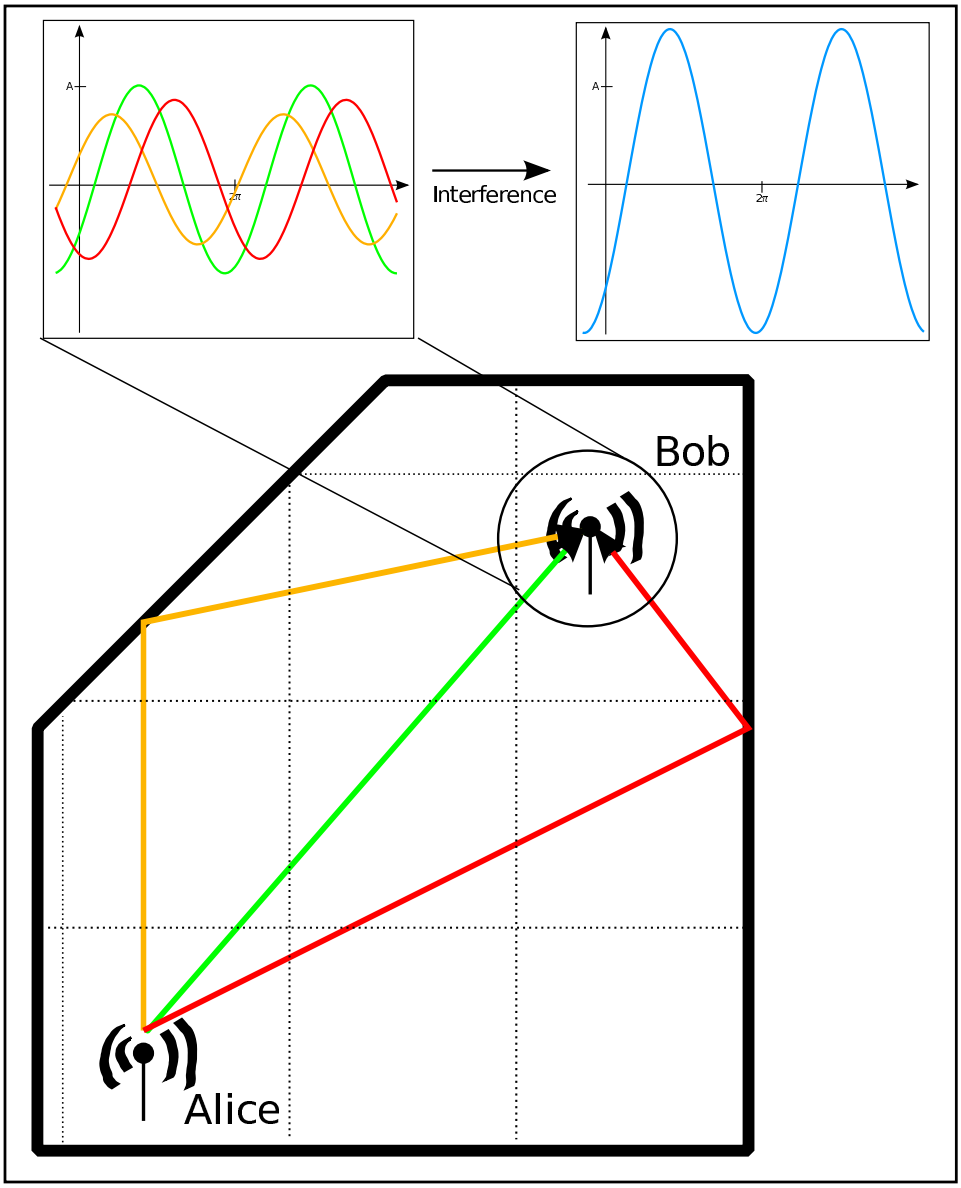


Figure 2.2: Simplified example of multipath effects. The signal takes three different paths and arrives at the receiver from three different directions, each affected by different channel effects. Both the signal amplitude and phase are affected. At the receiver, the waves interfere, resulting in a combined signal, the superposition of all multipaths. In this example, this leads to constructive interference, generating a larger amplitude. Different phase shifts, a changed signal frequency or different paths due to movement lead to different interference characteristics.

The most common effect is *wave reflection*. It occurs when radio waves hit a flat surface that is relatively large compared to the signal wavelength (in case of 2.4 GHz transmissions, the wavelength λ is approximately 12.5 cm). Moreover, different properties of the surface and the angle of incidence determine the attenuation of the resulting reflected ray. The attenuation is not strong with this effect, and most of the signal's power is preserved in general. Another phenomenon, which occurs when a signal interacts with objects along its propagation path, is *diffraction*. Diffraction usually takes place when a wave hits the edge of an obstacle. The edge will then act as a transmitter, effectively "bending" the signal around the obstacle. This effect is introduced by the nature of electromagnetic waves and results in a strong attenuation, it can be viewed as a partial shadowing. Similarly, a radio wave is affected by *scattering* if it hits rough surfaces or objects that are small in relation to the wavelength. As a result, the wave is split up into many different waves, reflected at different angles and phases. With the ray abstraction, this leads to a large number of strongly attenuated rays traveling in different directions.

In real-world scenarios, radio propagation will be subject to a combination of the aforementioned phenomena, resulting in transfer functions depending on many variables. This is because the transmitted signal at the receiver does not arrive as a single electromagnetic signal, but as a superposition of a large number of signals. By traveling along several paths, over different distances and with different phase offsets that result in constructive or destructive interference. The received signal strength may be weakened or amplified in contrast to the single path propagation case. Additionally, each one of the signal paths is affected by path loss and shadowing differently, which adds to the randomness of the resulting signal. Due to the high complexity and unknown parameters, such real-world propagation can often not be analyzed. The proposed propagation models contain several random variables, and exact values require a perfect knowledge of the hardware and antenna properties, as well as the physical environment. Thus, the simplified propagation models can at most approximate the amount of expected signal change, but the absolute received signal strength is still random and hard to predict.

To visualize the effects of multipath propagation, a simple scenario with three contributing paths is shown in Figure 2.2[1]. The green path, that is, the line of sight component, arrives with the largest amplitude at the receiver. The other two rays are reflected by walls and arrive from different directions at Bob's antenna. Due to these interaction with the environment, the two

[1]In accordance to conventions in security literature, the two legitimate parties in the protocol, as well as pairs of transmitters and receivers, are referred to as Alice and Bob.

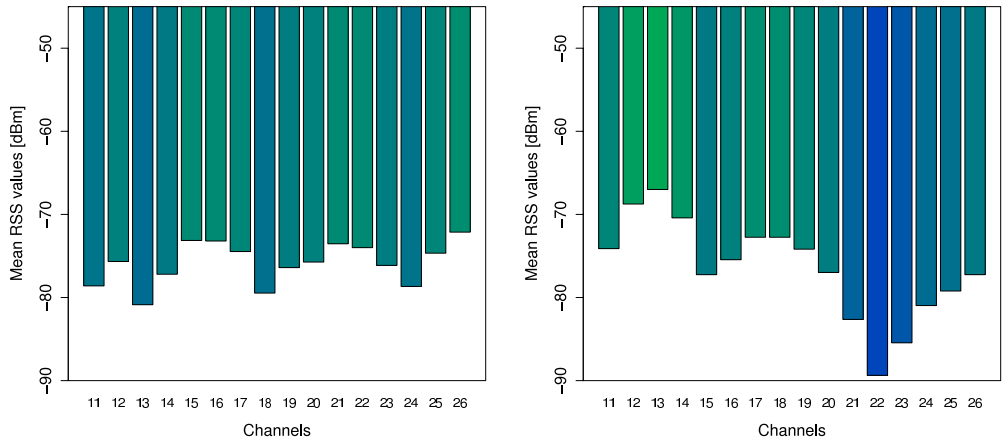
rays are attenuated, which leads to a reduced amplitude, as well as phase shifts, because of the longer paths and phase changes caused by reflections. Yet, Bob receives a signal with a higher amplitude as any of the individual multipath components, because the phases of the waves result in a constructive interference in this example. But if the red ray experienced a slightly different phase shift, the resulting signal is very weak because the green and red paths experience destructive interference in this case.

2.1.2.1 Frequency-selectivity of Multipath Fading

The phase differences and amplitudes at the receiver depend on the length of the multipaths, the environment and the frequency of the signal. Thus, constructive interference on one given frequency can turn into complete destruction of the signal simply by using a different frequency. By measuring the channel response over several different channels, the specific multipaths increase or decrease the resulting signal amplitude at the receiver unpredictably. Thus, changes in the carrier frequency lead to uncertain outcomes in the measurements. This fact enables the design of a key generation scheme that also works in static scenarios, and still has access to a rich source of unpredictable information. By acquiring the channel state on a number of channels, we can exploit the random interactions between multiple signal paths without the need for changing environments. This mitigates the shortcomings of the protocols proposed in Section 1.3, as strong secrets can then be generated without relying on device mobility. The shadowing component is also frequency-selective because of changing interaction with the traversed materials, but the multipath effects are much stronger and form the foundation of the proposed key generation protocol.

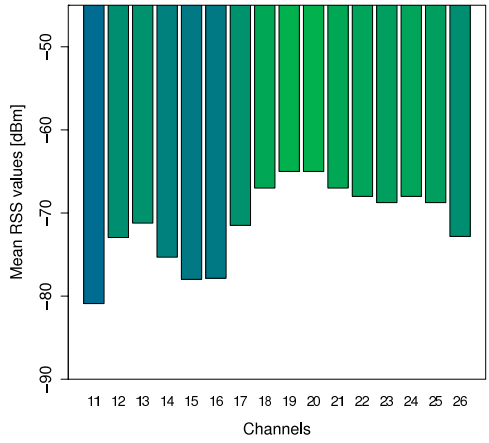
2.1.2.2 Sensor Mote Measurements

The described multipath effects can also be observed in measurement from our wireless sensor testbed. This section introduces experimental results for explanatory purposes, details on the acquisition of these results are presented in Chapter 4. To get a visual impression of the magnitude of the effects, Figure 2.3 shows measurements of four different positions of sender and receiver. 16 different channels (frequency bands) in the ISM band are available to the MICAz platform for probing, with a separation between the center frequencies of 5 MHz. One bar in the barplot represents the magnitude of the received signal strength in decibel, relative to a reference power of 1 mW. As this measure is logarithmic, a difference of 10 dB means a difference of factor 10 in signal

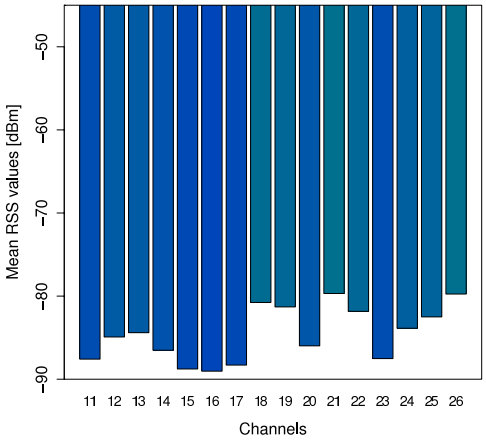


(a) Line-of-sight connection

(b) Strong destructive interference on some channels

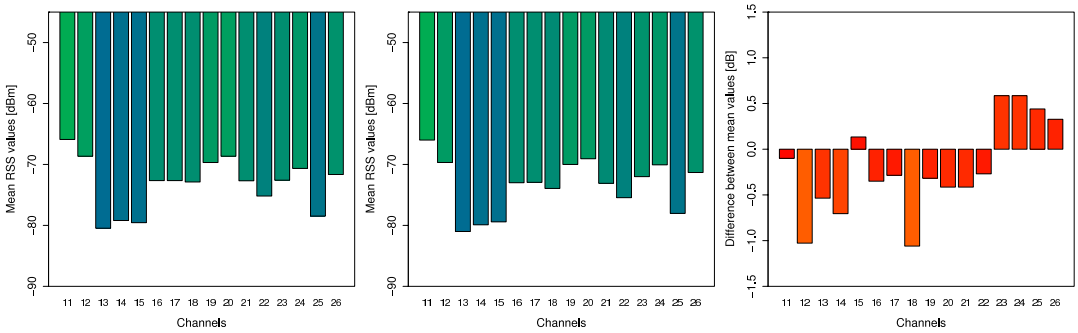


(c) Short distance between Alice and Bob



(d) Long distance with great path loss effects

Figure 2.3: Example of different attenuation effects in a real-world WSN environment.



(a) Alice's view (b) Bob's View (c) Differences

Figure 2.4: Example WSN measurement showing the reciprocity of the wireless channel. The first two bar plots show the view that the sampling parties have of the channel, the last figure shows the deviations in each channel.

strengths. Constructive interference of two equal waves can therefore lead to an increase of +3 dB (doubling the signal strength), while destructive interference can produce arbitrarily small dBm values, as a received power of 0 W equals $-\infty$ dBm.

Figure 2.3a shows several channels with relatively flat fading, that is, no large deviations in the signal strength are experienced when measuring on different frequencies. The channel effects lead to an attenuation of approximately 10 dB. Figure 2.3b shows a strong deep fade, that is, the signal is weakened significantly such that no signal is detected on channel 22 by the receiver. This happens if the signal strength is below the smallest signal power that the receiver can recognize. In this example, the most powerful signal is more than 100 times stronger than the weakest signal. A short distance results in large signal strengths on all channels, as the path loss is not strong in such settings. A large distance, on the other hand, results in a large amount of path loss. These effects are visible in Figures 2.3c and 2.3d.

2.1.3 Reciprocity of the Wireless Channel

One important aspect is the randomness of the channel characteristics, but the fact that this randomness is accessible by both of the two participating parties, even if they are distributed, allows for a key generation protocol. A correlated random variable shared between Alice and Bob is needed. The principle of reciprocity states that two transceivers will experience the same channel properties when the role of sender and receiver is switched, given that the time gap

between the two signals is small enough. The duration of the maximum time gap, called the channel coherence time, refers to the length of the time interval in which the same wave propagation characteristics can be assumed. This reciprocity can only be observed directly if the transceiver hardware, antennas and transmitted signal strengths are equal. Otherwise, a calibration between the two parties is needed to arrive at the same measurement results.

An example measurement of reciprocity is given in Figure 2.4. Both parties experience nearly the same signal strength values on all channels, with bounded deviations. This information can be used as an advantage over an adversary, as he has to guess the amount of attenuation that Alice and Bob can simply measure by sampling on the wireless channel. The experimental analysis in Chapter 4 showed that the reciprocity is strong enough to guarantee successful key agreements, independent from the characteristics of the positions. This is the case even in environments with numerous multipaths, or with moving devices and antennas.

2.2 Secrecy Considerations

We have seen that the wireless channel has several desirable properties that be can used to provide secrecy. In "real" wireless transmissions, a combination of all the effects described in the previous section is experienced, making it highly unpredictable and very sensitive to position and frequency. A changing position leads to signals traveling on different paths, which in turn leads to altered resulting signal strengths because of multipath interference and shadowing. By continuous movement, an arbitrary number of secret bits can be produced, but the second source of randomness, the frequency-selectivity, can leverage the uncertainty that multipath fading offers better by resolving the individual paths and using the resulting information.

This unpredictability, even with small deviations in the position, in shown in Figure 2.5. In this experiment, the sensor mote serving as Alice is located on a desk. Bob is placed in the adjacent room, that is, the line of sight connection is obstructed by several attenuating objects. A wooden desk, a concrete wall and a metal shelf generate a part of the amount of fading that the signal experiences. This scenario is typical for indoor applications of wireless sensor networks. Bob is placed on a small table and a measurement of the channel state is taken. Then, Bob is placed on several positions located on a circle with a radius of 10 cm around his initial position, and new measurements are taken. A total of 12 positions on the circle are considered, with an equal distance between the positions. The results show that even small changes in the position have a

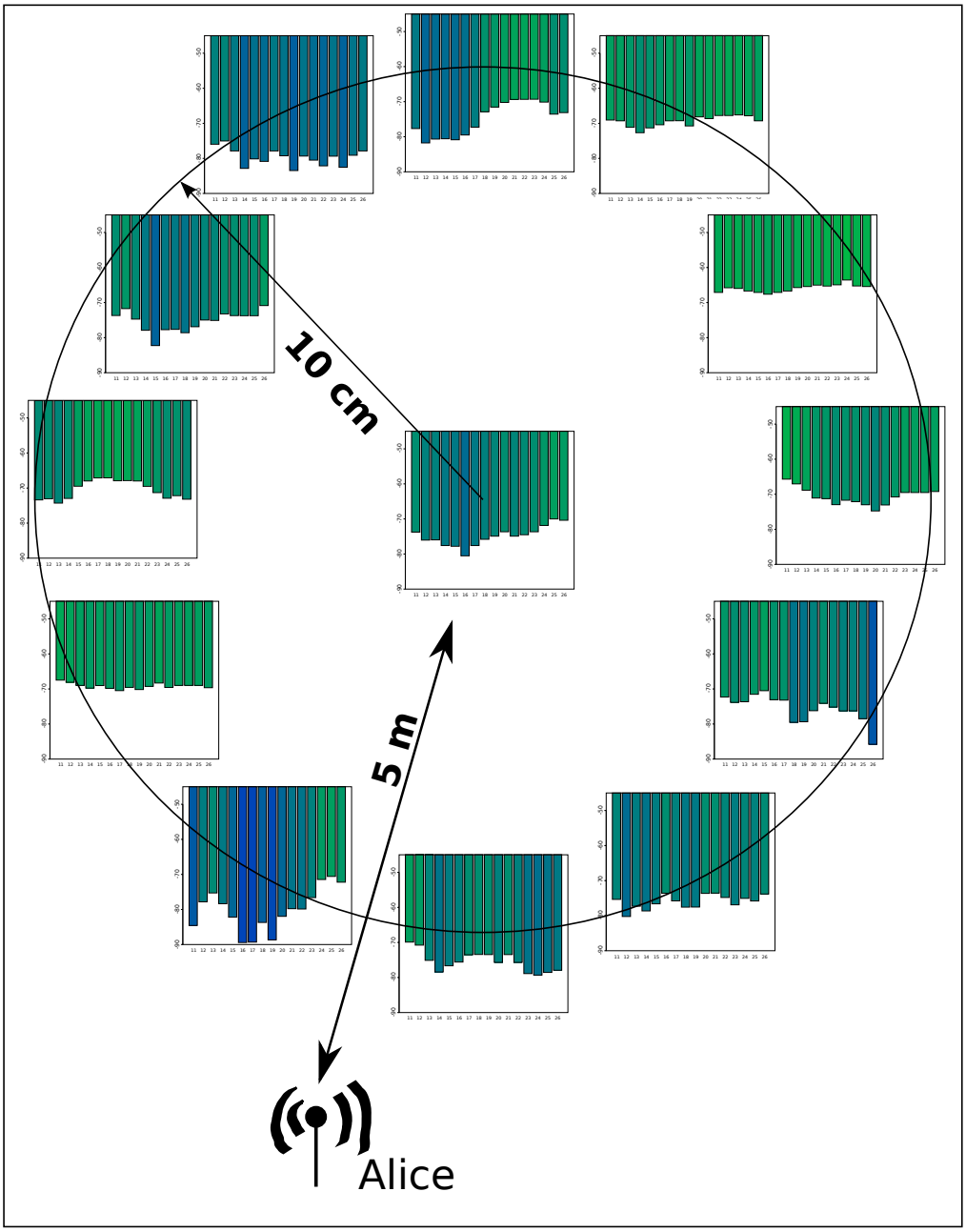


Figure 2.5: Spatial selectivity of the channel state observed in measurements on a circle with 10 cm radius. The position of Alice is fixed and place 5 meters away from Bob, and the line of sight between Alice and Bob is obstructed by a desk, a wall and a metal shelf, all of which influence the propagation characteristics of the wireless channel. The position of Bob is shifted in a circle with a radius of 10 cm during the experiment, and signal strength measurements are taken for each position.

large impact, ranging from very strong signals on all available channels to very strong attenuation. Even if an adversary can estimate Bob's position with a precision of 5 cm, he is still ignorant of the true measurements. All of these positions show a strong reciprocity, i.e., Alice experiences the same channel characteristics as Bob does.

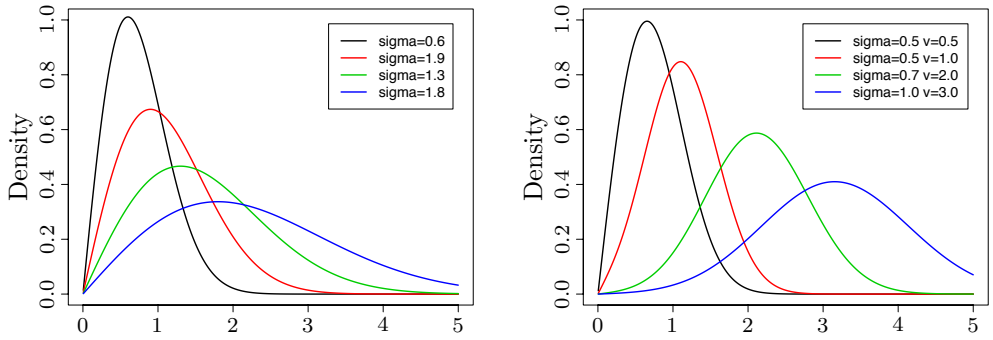
In conclusion, the resulting signal strength is hard to predict, as an exact model of the antennas and hardware, as well as the environment with its surfaces and obstacles must be available. Therefore, the signal strength value introduces uncertainty for adversaries who are not at same physical positions as the legitimate parties. But how can this uncertainty be quantified? An important step to answer this question is the modeling of a strong adversary.

2.2.1 Adversarial Model

One important aspect for the quantification of secrecy of such a scheme is to define what an adversary is able to do, in the same way as it is necessary when evaluating cryptographic security. A computationally unbounded attacker can break Diffie-Hellman key agreements with ease because such an attacker can solve any problem that relies on computation complexity. Similarly, an attacker who can take the same physical positions as the legitimate sensor nodes can destroy the benefit of this key generation protocol. But with realistic constrains on an attacker, the security of the protocol can be analyzed, and realistic scenarios can be considered. The goal of this thesis is to show how the security of this protocol can be evaluated given realistic propagation models, as well as empirical data, when considering the adversary defined in this section.

The adversary has several options to attack the execution of the key generation protocol. He can eavesdrop on the wireless channel and observe both the content of the messages and the signal strengths that he can experience at his position. As the content of the messages carries no information and the signal strength de-correlates rapidly in space, this gives him very little information on the channel state between Alice and Bob, thus, eavesdropping does not give an advantage to the adversary. This was shown empirically in [37]. With his presence, he can only prevent Alice and Bob from sending the secret in plain text over the wireless channel.

Because the eavesdropping gives only little to no gain in information, the best attack vector is to model the multipath channel between Alice and Bob, taking into consideration the hardware and environment, and then determine the signal strength values. Knowledge to aid him in this modeling might come from plans of the building for indoor scenarios or from observations of the environment, or from the positions of Alice and Bob by observation of the sensor



(a) Rayleigh distribution for different parameter choices

(b) Ricean distribution for different parameter choices

Figure 2.6: Wireless channel propagation models.

motes or via positioning methods using wireless signals such as triangulation. While the effects of path loss and shadowing on the direct connection between the two nodes are predictable (e.g., using ray-tracing methods [19]), the resolution of the multipath components is very challenging. To refine his model, the adversary is allowed to do measurements with similar hardware somewhere off-site. The only limitation in this is that the attacker cannot measure at the same positions of the legitimate sensors during operation, because this is equivalent with a node capture that discloses the key directly.

Given this information, we can model the knowledge of the adversary by limiting the possible signal strengths to the distribution of signal strengths of similar positions. This leads to a possibility to quantify the amount of uncertainty that an attacker experiences. This can be achieved by using the distribution of signal strength values from channel propagation models.

2.2.2 Predictability of Wireless Propagation

A multitude of channel propagation models have been proposed to capture the most important aspects of the wireless characteristics. This ranges from simple models, such as the free-space model which only considers path loss, models that try to capture additional forms of randomization by introducing random variables such as log-normal shadowing, to models that explicitly formulate the physical aspects of interest such as the two-ray model, where the interference of the line of sight and of a reflected ray are calculated geometrically. These models enable the analysis of networks, but with a loss of information due to generalization, as the real amount of fading is lost when considering only the

distribution of signal strength values. The true wireless channel's properties are still hard to predict for a certain position, yet, the distribution of signal strengths can be captured accurately by channel propagation models in the literature [46]. Thus, by using distributions, the individual characteristics of a position are lost, and only a probability is given on the state of channel parameters, e.g., the amplitude of the transfer function. This can be used to quantify the uncertainty in the channel state that an adversary experiences, even when he has a precise model that perfectly describes the wireless channel between Alice and Bob.

2.2.2.1 Channel Propagation Models

There are several distributions in the literature which model the received signal strength well in common scenarios and consider mentioned attenuation effects. A common continuous distribution with a single degree of freedom is the Rayleigh distribution, which can be used to describe the effects of several multipath components whose phase shifts follow a Normal distribution. Therefore, this distribution is used if there is no strong line of sight component. The Ricean distribution is a generalized Rayleigh distribution with two degrees of freedom, and the second parameter is able to control the strength of the LOS component. Thus, this distribution is mainly used to describe signal strength measurements with a non-negligible direct connection between two sensor motes. The following formulas describe the probability density functions (pdfs) of these distributions.

$$\text{Rayleigh} \quad p(x,\sigma) = \frac{x}{\sigma^2} \cdot \exp\left(-\frac{x^2}{2\sigma^2}\right) \qquad \text{rms } \sigma > 0$$

$$\text{Rice} \quad p(x,\sigma,v) = \frac{x}{\sigma^2} \cdot \exp\left(-\frac{x^2+v^2}{2\sigma^2}\right) \cdot I_0\left(\frac{x \cdot v}{\sigma^2}\right) \qquad \text{rms } \sigma > 0, \text{ peak } v \geq 0$$

Figure 2.6 shows plots of these functions. Both distributions have a part that is shaped similar to the Normal distribution, and a tail part that represent rare events, which are still not negligible. The parameter σ controls the shape of the peak, and the second parameter in the Ricean distribution, ν, can be used to shift the center of the distribution. $I_0(z)$ in the formula is the modified Bessel function of the first kind with order zero. In wireless channels where the received signal strengths are distributed according to these models, the interpretation is as follows: the majority of expected signal strengths are in one part, and strong attenuation events such as deep fades are located in the tail part of the distribution.

2.2.3 Uncertainty of Wireless Propagation

In accordance to the adversarial model in Section 2.2.1, we can conclude that the uncertainty that an adversary experiences only depends on the distribution of signal strengths that can be experienced by Alice and Bob. If we assume that the attacker has perfect knowledge of this distribution, then how much shared secrecy is left between the users of the key generation protocol? Again, information theory offers a notion that fits well in this context: the notion of *information entropy*.

2.2.3.1 Entropy, Mutual Information

In information theory, (Shannon) entropy is a measure of the uncertainty associated with a random variable [48, 49]. It quantifies the average amount of information that is learned on observation of the outcome of the random variable, and is usually measured in the unit of bit[2]. If p denotes the probability mass function of X, then the entropy for this discrete variable can be written as

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p\left(x_i\right).$$

Consider we want to generate a bit string using the random variable C_{fair} which models a fair coin, that is, outcomes of coin tosses are uniformly distributed, the probability of the two events $\{H, T\}$, denoting the outcome "heads" or "tails", are both $\frac{1}{2}$. If the coin comes up with heads, then a "1" is appended to the bit string, "0" otherwise. After n tosses, we have generated a random string from the set $\{0,1\}^n$. The entropy of a single bit in this string, created using the fair coin as the source of randomness, is

$$H\left(C_{\text{fair}}\right) = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) = 1.$$

This means every bit is unknown, and thus also carries an uncertainty of one bit. This is of course the maximum amount of entropy, and is was shown that uniform distributions are always the maximum entropy distributions if the range of possible values is limited.

When considering the entropy of our channel state measurements, we already observed that the outcome is not uniformly distributed, some values are

[2]For this thesis, entropy values are always in the unit of bit if not explicitly stated otherwise, and all logarithms are to the base 2. As the conversion factor from natural units (nats) to bit, the value 1.442 was used when necessary.

more likely than others. If a coin is not fair but biased, we can observe a similar influence on the resulting entropy. The biased coin has a probability for heads of $p\left(C_{\text{bias}} = H\right) = \frac{1}{16}$, and the only other possible outcome has the probability $p\left(C_{\text{bias}} = T\right) = \frac{15}{16}$. The information carried by one bit in the resulting bit string, the entropy, is given by

$$H\left(C_{\text{bias}}\right) = -\left(\frac{1}{16}\log\frac{1}{16} + \frac{15}{16}\log\frac{15}{16}\right) = 0.234,$$

that is, a bit in the resulting string represents only 0.234 bit of new information. Using one complete bit for the encoding of the outcome "tails" is mostly redundant, as this outcome can be easily guessed. It is sufficient to encode the positions of the "1"s, or, equivalently, the occurrences of "heads" in the sequence of coin tosses, as such an event carries much more information. The key generation scheme that we consider can be analyzed similarly. The wireless channel is a source of randomness with non-uniform distributions of outcomes, similar to the biased coin. Alice and Bob can measure the outcome of these "coin tosses" in the wireless channel, but the adversary can only estimate the distribution of measurements, not the exact values themselves. This uncertainty is also captured by the entropy, which is therefore the best measure for our purposes.

One difference to the coin example is that the legitimate parties only have a degraded access to the channel state information, as measurement errors or differences in their views can lead to differing measurement results. Both parties see a slightly different outcome from the random variable. The amount of information that both parties share after observing this variables is given by the *mutual information* $\mathrm{I}(X, Y)$. This information should be as high as possible to achieve a large secret between the two legitimate nodes. As a maximum, the entropy of the random variable can be used as mutual information. And as Eve also has access to a degraded version of the two correlated random variables X', Y', there is also a flow of information between Alice and Eve and between Bob and Eve, given by the mutual information $\mathrm{I}(X, X')$ and $\mathrm{I}(Y, Y')$, respectively. This leakage of secrecy should be as small as possible. As mentioned before, the wireless channel properties are known to de-correlate quickly, both in time and space, and this eavesdropping on the channel state is not the best attack vector. Full knowledge of the underlying distribution is the best way for an adversary to infer information on the shared secret.

Several different entropy measures are proposed in the literature. For discrete random variables, the notion of Shannon entropy and min-entropy are considered. The Shannon entropy describes the complete amount of information available in the random variable while the min-entropy is a metric for the

minimum amount of secret information in a variable, or put differently, the *predictability* of a random variable [40]. It is defined for a discrete random variable A with $\text{supp}(A) = \mathcal{A}$ as

$$H_\infty(A) = -\log_2 \left(\max_{a \in \mathcal{A}} \Pr[A = a] \right).$$

The available min-entropy is maximal in case A is uniformly distributed, i.e., in our context this would mean no preference for some RSS measurements over others is present.

The differential Shannon entropy tries to extend the concept of the discrete Shannon entropy to continuous random distributions. Let X be a continuous random variable with probability density function f, whose support is the set $\mathcal{X}$. The differential entropy $h(X)$ is defined as

$$h(X) = -\int_{\mathcal{X}} f(x) \log f(x) \mathrm{d}x.$$

However, there are some problematic aspects of the differential entropy, most notably that it can be negative as the pdf can be larger than one. To get comparable values for the entropy, this case must be checked. We will use the differential entropy later for secrecy analysis the protocol.

The underlying distributions are considered to be known to evaluate the entropy. The channel models presented in the previous section can be used to make general statements, but to evaluate the experiments, the underlying distributions are unknown. How this problem can be mitigated is presented in Chapter 4.

2.2.3.2 Joint Entropy, Conditional Entropy

In the case of a joint system with several random variables, the notion of *joint entropy* describes the total amount of entropy in the resulting random vector. In the case of independent random variables, the joint case is straightforward: the sum of the entropies of the random variables is the joint entropy. Considering the bit string produced by the fair coin, this is the case, each additional coin flip adds one bit of entropy.

But this is not the case in settings with dependent variables, as these variables are predictable if another correlated variable is known, and the total amount of entropy is lowered. As an example, consider a "magic" coin that has the same probability for heads and tails on the first toss, but then shows the same outcome in every following toss. Clearly, no additional information is conveyed after one bit is known. But this in an extreme example, and

the nature of interdependence between individual channels is hard to quantify from experimental data, leading to unknown amounts of entropy loss, as the underlying distributions are unknown. Yet, as an example, a deep fade on one channel makes low signal strengths on adjacent channels more likely. The problem of quantification of such effects is considered in the analytical part of this work.

2.2.4 Discussion

These considerations take a very powerful adversary into account, especially as it is assumed that the attacker knows the underlying signal strength distribution exactly. In real-world scenarios, this is only approximately known to an adversary, and therefore the secrecy of this concept is generally higher, as the entropy is a conservative metric for secrecy. The strength of the keys can thus be considered the be higher in most uses. Additionally, the entropy is based on models of the wireless channel, but the nature is even more random and erratic.

2.3 Conclusion

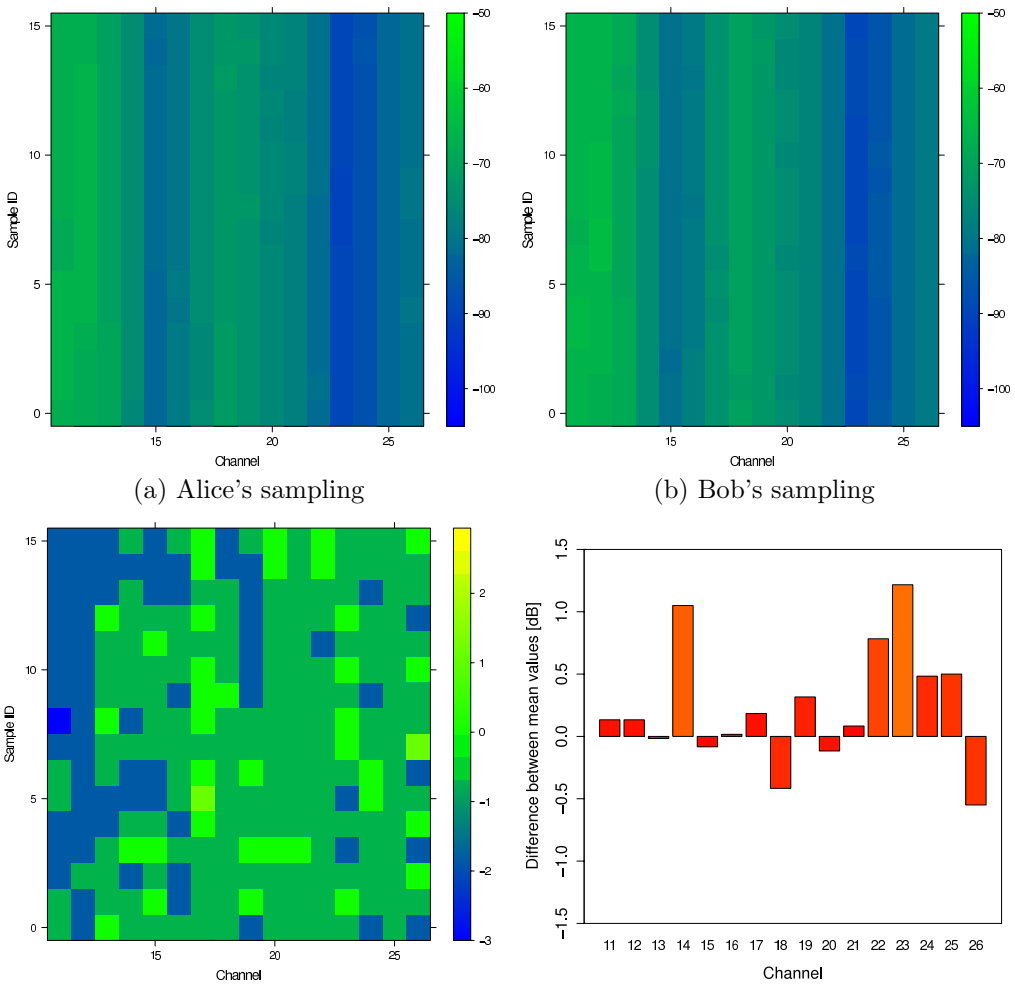
This concludes the concept of the key generation protocol. The origin of randomness was explored, as well as consideration to quantify the security of such a scheme. The channels state seems to be a good candidate to base the key generation protocol upon, and the next chapter will show how this value can be accessed on WSN hardware platforms, and how to correct the deviations in those measurements.

3 Protocol Design

This chapter discusses the key generation protocol design, focusing on implementation efficiency and practical considerations of wireless sensor networks. Starting from the acquisition of the physical channel state, the process of turning analog information into a digital secret is described. The proposed protocol uses several phases, and the structure of this chapter reflects this design. The first section considers the sampling process regarding the wireless state information that current hardware platforms offer by the indication of received signal strength. An important goal of the protocol is to generate keys reliably, that is, the probability that both parties arrive at the same key must be high. This requires a capability to detect and repair mismatches, also referred to as reconciliation. For this end, an error correction mechanism based on *error-correcting codes* (ECCs), a notion from communication theory, is presented. Finally, the complete protocol is discussed, which, despite its straight-forward design, is capable to produce strong secret keys reliably. But first, a way for the sensor motes to access wireless channel information has to be identified.

3.1 Secrets from the Wireless Channel

The first question to answer is: How are the channel state parameters accessible on current sensor mote hardware? Almost all transceiver platforms report some form of indication of the received signal strength of a packet. This information can be used to choose the best next hop in ad-hoc routing protocols, as the strongest signals normally also indicate the best connection, that is, the connection with the lowest rate of bit errors. The most common metric for this strength measurement is the *received signal strength indicator* (RSSI). It is a dimensionless value, which means the hardware manufacturers can choose freely how RSSI values relate to physical power measurements in milliwatt (mW) or decibel relative to 1 mW (dBm). This measure is not required to have a linear or logarithmic relationship to the true signal strength values, the only requirement is that higher RSSI values indicate higher signal strengths. Despite these weak regulations, most hardware devices provide quite accurate



(a) Alice's sampling

(b) Bob's sampling

(c) Sampling differences

(d) Differences of mean values

Figure 3.1: The sampling process using wireless sensor motes. The samples are collected one channel at a time, until 16 samples are taken, and until all channels are probed. Sub-figure 3.1c shows the differences between two sample measurements. The protocol uses mean values to overcome these short-term deviations.

measurements, and the RSSI values can directly be converted to correct signal power values (RSS values), measured in mW. Chapter 4 shows, based on extensive experiments, that the measurements are stable, even on low cost hardware and precise enough to show a strong reciprocity.

Thus, with every message received, a RSS value is report, which represents the power on the wireless channel which is perceived during the reception of the packet. The value refers to the average power that the signal transports, but also to the power from concurrent transmissions and other sources of electromagnetic radiation, which can lead to inaccurate measurements. By sampling the channel and collecting RSS information of the channel state, we can infer the magnitude of the transfer function, as the received signal strength metric is computed using the integral over the squared amplitude of the signal. As the sensor hardware supports multiple frequencies to allow coexistence and interference avoidance, we can sample the channels with several frequencies to produce vectors of channel state information.

The wireless channel is stationary during the channel coherence time. As this time depends on the speed of movement, fixed positions of sensor nodes result in stable RSS readings on each of the channels. This means that the sampling messages will show only small deviations, and an increased number of samples results in a larger precision in the estimates of signal strength values.

The acquisition process for WSN hardware is shown as an example in Figure 3.1. 16 channels are used, and 16 sampling messages are exchanged on each channel. Each column represents the measurements on a single channel and is divided into tiles, their color representing the received strength of a single sample signal. Due to the reciprocity, the corresponding tiles in the measurements of Alice and Bob are very similar, as the time between the two measurement is small. The stability of the channel response can be seen by the uniformity of each column, the number of outliers is small in general. In order to get a precise picture of the state of the channel, mean values should be used by both parties, because the impact of short-term effects such as rapid movements or cross-talk can be reduced. This effect can be seen in Figure 3.1c. While single samples show strong deviations, the differences in the resulting mean values are small (Figure 3.1d).

Formally, we collect k samples for each of the n channels that the hardware can access. The results of the RSS measurements, $\mathbf{m}_i = \left\{ m_i^{(1)}, \ldots, m_i^{(k)} \right\}$ with $i = 1, \ldots, n$, can be combined to form the vector of measured mean values $\mu_i = \sum_{j=1}^{k} m_i^{(j)}$ for all channels $i = 1 \ldots n$. The central limit theorem states that the samples for each channel follow a Normal distribution with this mean μ_i if the number of samples is large enough. Each mean value can be viewed as drawn

from a distribution that describes the channel propagation characteristics (e.g., it follows a Rayleigh distribution).

In the following, we assume that we can conduct measurements by sampling RSS values on a set of n different frequencies $\mathcal{F} = \{f_1, \ldots, f_n\}$ (also referred to as *channels*). We view the mean of these samples taken on an individual channel f_i as a random variable M_i, and the means μ_i of all n channels as the random vector $\mathbf{M} = (M_1, \ldots, M_n)$. A realization, the outcome of our measurements is $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$, with $\mu_i \in \mathcal{M} = [\mu_{\min}, \mu_{\max}]$, the range of mean values that can be measured by the hardware platform. We assume that $\mathcal{M}$ is a finite subset of $\mathbb{R}$, that is, only a finite precision in the measurements is achieved, and use properties of $\mathbb{R}$ such as ordering and relations when discussing dependencies of elements in $\mathcal{M}$. As an example for this set, in our wireless sensor network measurements we used $\mathcal{M} = [\text{-}104, \text{-}40]\,\text{dBm}$, with a precision depending on the number of samples taken, since each RSS sample is integer valued. We associate $\mathcal{M}$ with the distance function $dis : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ defined as $dis\,(\mu, \mu') := |\mu - \mu'|$, which is the difference in dB in our case. Thus, $\mathcal{M}$ together with this distance function constitutes a *metric space*.

3.1.1 Sources of Errors

Taking several samples during the RSS measurements increases the precision of the means, but it is still unlikely that both Alice and Bob measure exactly the same values. The RSSI values that are reported by the hardware are integer valued, although the signal strength itself is a continuous measure. This introduces errors in the order of 0.5 dBm on current hardware platforms. As the RSS indicator is not designed to be a precise source of channel state information, two different devices can also introduce different views on the wireless channel even if the physical characteristics are equal. Another source of errors is the inevitable noise in the measurement circuits. Again, the hardware was manufactured with cost-efficiency in mind, not to deliver a maximum of precision in such measurements. A large factor is imperfect reciprocity, which can result from differences in antennas or transceiver hardware, but also from asymmetries in the environment that can cause different multipath effects at the two sensor motes.

Another problem, which can be accounted to hardware limitations, is the limitation to simplex transmissions. Current hardware is only able to either send or receive at a given moment in time, and during the sampling process, the resulting time gaps can lead to different views on the channel state. And as sensor motes need time to do a turn-around from sending to receiving both in hardware and software, this gap limits the sampling rate severely. The use

of mean values in the proposed key generation protocol mitigates this problem. This is an important design aspect that makes the protocol applicable for low-cost devices.

One of the major problems for signal strength estimation is cross-traffic and interference from other sources. These additional waves can add to or decrease the received strength by interfering with the received signals. The sensor motes have a simple channel assessment mechanism that tries to ensure that the channel is unoccupied when the sensor motes start transmitting. Yet, deviations due to other senders in the vicinity can not fully be eliminated with this mechanism. Especially sources of electromagnetic radiation such as microwave ovens and other devices that radiate on the free ISM band are irrespective of other users of the wireless channel.

As the errors are inherent in the system and cannot be avoided in most cases, the system must be able to cope with errors in the measurements. We show in the next section that all of these factors can be controlled, as long as the deviations are bounded.

3.2 Error Correction

Given the values $\boldsymbol{\mu}, \boldsymbol{\mu}' \in \mathcal{M}$ measured by Alice and Bob, the goal is to obtain a shared value without revealing any information to Eve. To reliably reconcile information, error correction is crucial because brute force approaches based on testing all possible combinations are infeasible in the context of resource-limited devices. Also, the different combinations must be compared to each other by public communication with the other party, again without revealing information, which requires additional computational effort to hide the real secrets, for example, by applying a hash function on each of the possibilities. Clearly, this scheme does not translate well to the context of WSNs.

Coding theory provides a useful framework to describe error correcting codes [33]. This powerful tool enables us to repair deviations and reach a shared binary string with a minimum of communication used for reconciliation. The codes considered here are applied to the mean values μ_i of each channel, i.e., the elements in the random vector can be corrected using different codes. The design focus of the code is the possibility of a performance-aware implementation. The reconciliation process should require a minimum number of messages and computation time, and the memory footprint must be small. This section describes the definition of a family of codes which is designed to cope with the kind of bounded deviations experienced in the experiments.

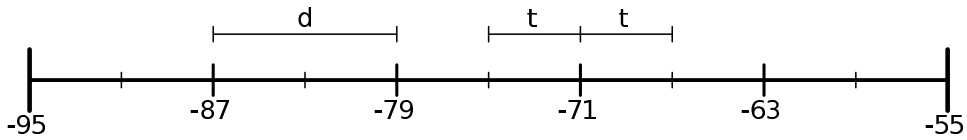


Figure 3.2: Codewords in linear space, this is an example with a tolerance value t of 4 dB. This code has 6 codewords and the distance between the codewords is $d = 8$ dB.

3.2.1 Error Correcting Codes

In general, a code $\mathcal{C}$ is a subset of metric space $\mathcal{M}$, $\mathcal{C} = \{c_1, \ldots, c_K\} \subseteq \mathcal{M}$, with a total of K elements, called codewords. The map from $\mathcal{M}$ to $\mathcal{C}$ is called *encoding*, denoted as *enc*. In the encoding process information is lost, as several elements of $\mathcal{M}$ are encoded to the same value in $\mathcal{C}$ in general. The only exception is the case $\mathcal{M} = \mathcal{C}$, where *enc* is a bijective function. The *minimum distance* d between the codewords in $\mathcal{C}$, that is, the smallest $d > 0$ such that for all $i \neq j$ we have $dis\,(c_i, c_j) \leq d$, is a measure for the smallest deviation in an $m \in \mathcal{M}$ that can be detected. The closer the codewords are together, the more sensitive is the code to changes in the input values. The most important property of a code for our application is the *error-correcting distance* t of $\mathcal{C}$. This is the smallest distance for which an $m \in \mathcal{M}$ is encoded uniquely, i.e., for all $m \in \mathcal{M}$, we have $dis\,(m, c) < t$ for at most one $c \in \mathcal{C}$. Therefore, all values m, m' are encoded to c given their distance to c is small enough. We refer to this value of t as the *tolerance* of the code.

Common codes such as Hamming and Reed-Solomon codes operate on the Hamming distance metric and therefore lead to undesirable tolerance characteristics, as this metric is unaware of the magnitude of deviations between measurements. As an element $\mu \in \mathcal{M}$ represents a number in $\mathbb{R}$, the bits in the binary representation of μ have different significance. This leads to the problem that a deviation of 32 dB is not different to a difference of 2 dB for the code, as in both cases only one bit is flipped. Thus, we need to construct a code that considers our special distance function; only deviations that are smaller than a maximal distance t must be corrected, with this distance being the absolute distance measure in $\mathbb{R}$.

3.2.1.1 Code Construction

The construction is as follows: we choose $K = 2^p$ elements of $\mathcal{M}$ such that we have the same distance d between all codewords, where p is the number of bits that are needed to identify a codeword. This equidistance en-

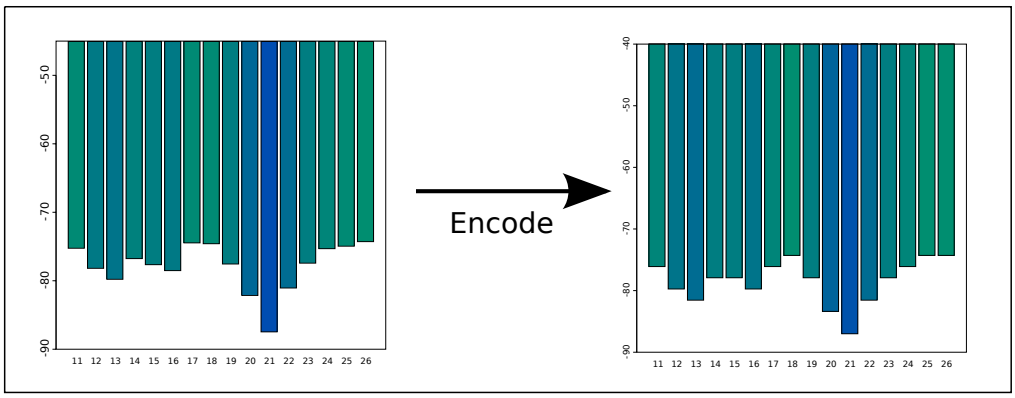


Figure 3.3: Quantization of the measured values.

sures that the tolerance is the same for all values in $\mathcal{M}$. We denote this code as $\mathcal{C}_p = \{c_1, \ldots, c_{2^p}\}$, the bijective mapping to the binary representation as $bin : \mathcal{C}_p \to \{0,1\}^p$, which maps codewords to binary strings. Since $\mu_{\min}$ and $\mu_{\max}$ are both fixed values, the distance d between neighboring codewords is reduced as the number of codewords increases. The relation is given by $d = \frac{|\mu_{\max} - \mu_{\min}|}{2^p - 1}$. The tolerance of such a code is given by $t = \frac{d}{2}$, since all codewords are evenly spaced. The number of codewords therefore directly affects the tolerance of the code, when fewer codewords are considered, the more deviations can be repaired. The process of encoding maps the value μ to the codeword c with the minimal distance in $\mathbb{R}$, which can be viewed as a *quantization* of the measured value. As an example, we consider the code $\mathcal{C}_5 = \{\text{-}104, \text{-}101, \ldots, \text{-}42, \text{-}40\}$ with 32 codewords and tolerance $t = 1$ for our metric space $\mathcal{M}$. For this code, the measured value $\mu = \text{-}71.424\,\text{dBm}$ is encoded to the codeword $c = \text{-}72$. A simplified representation of these properties is given in Figure 3.2. All values in the region of length t around a codeword are encoded to it. If a mean value μ_x is exactly in the middle between two codewords, that is, $dis\,(\mu_x, c_i) = dis\,(\mu_x, c_j) = t$ with $i \neq j$, then the larger codeword with respect to the relations in $\mathbb{R}$ is chosen. This is the only case when the encoding function was not well-defined.

An example how this applies to the experimental values is shown in Figure 3.3. The continuous measurements on the left side are rounded to the next quantization levels of the code for each channel. In this example the tolerance is $t = 2$, i.e., the codewords are spaced 4 dB apart and deviations of up to 2 dB are ignored. Mean values which are close together such as the measurements on channel 25 and 26 are thus encoded to the same codeword.

3.2.1.2 Tolerance Properties of the Code

The amount of uncertainty is reduced in this process as some values become impossible, but at the same time the tolerance for deviations is increased. Thus, we can trade robustness vs. secrecy by choosing a code $\mathcal{C}_p$ with suitable parameter $p \in \mathbb{N}$, which is able to correct errors in measurements given $dis\,(\mu, \mu') < t$. Similarly to the distance function, the tolerance can be described as acceptable measurement deviations, such as ± 1 dB in received signal strength. Figure 3.4 shows how an increase of tolerance changes the resulting codewords. The measured values to the left show bounded deviations, on some channels the measurements are almost equal, some deviations are over 1 dB. Encoding these sample mean values with a tolerance value of $t = 1$ corrects most of the errors, only channels 19 and 24 are disagreeing, as the deviations on those channels is too high for this tolerance value to be repaired. An increase in tolerance to a value of $t = 2$ repairs all deviations, but the price for this is also visible: the reduced number of codewords leads to many measurements that are encoded to the same codewords. This fact makes these values easier to predict for an adversary. In this example, 9 different quantization levels are possible for $t = 1$, and for $t = 2$ only 4 levels are remaining. The RSS bar plot on the right shows a tolerance value of $t = 4$, i.e., the codewords are separated by 8 dB. Here only 3 levels are remaining, with a strong precedence of the -82 codeword. Care must be taken to choose a code that guarantees a successful reconciliation, but at the same time does not remove too much entropy in the process.

3.2.1.3 Error Reconciliation

With this construction, we are usually able to reconcile many deviations between μ and μ' given $dis\,(\mu, \mu') < t$. Still, some constellations are possible such that μ and μ' are encoded to two different codewords (e.g., given $\mathcal{C}_5$, μ = -70.9 dBm and μ' = -71.1 dBm are encoded as -70 and -72, respectively). To correct these error patterns, we need to send a public piece of information P that helps Bob to reconcile his measurement and recover the same codeword as Alice. Of course, at the same time P should reveal a minimum of new information to Eve.

Our construction is straightforward: Alice calculates $P = enc(\mu) - \mu$, the shift that is necessary from μ to the corresponding codeword $c = enc(\mu)$, and uses c as her secret information. This is similar to syndrome constructions that represent the necessary "shifts" to reach a codeword, but does not reveal information on the codeword itself. This shift is always smaller than or equal to t, and therefore reveals only information that is discarded by Alice and Bob any-

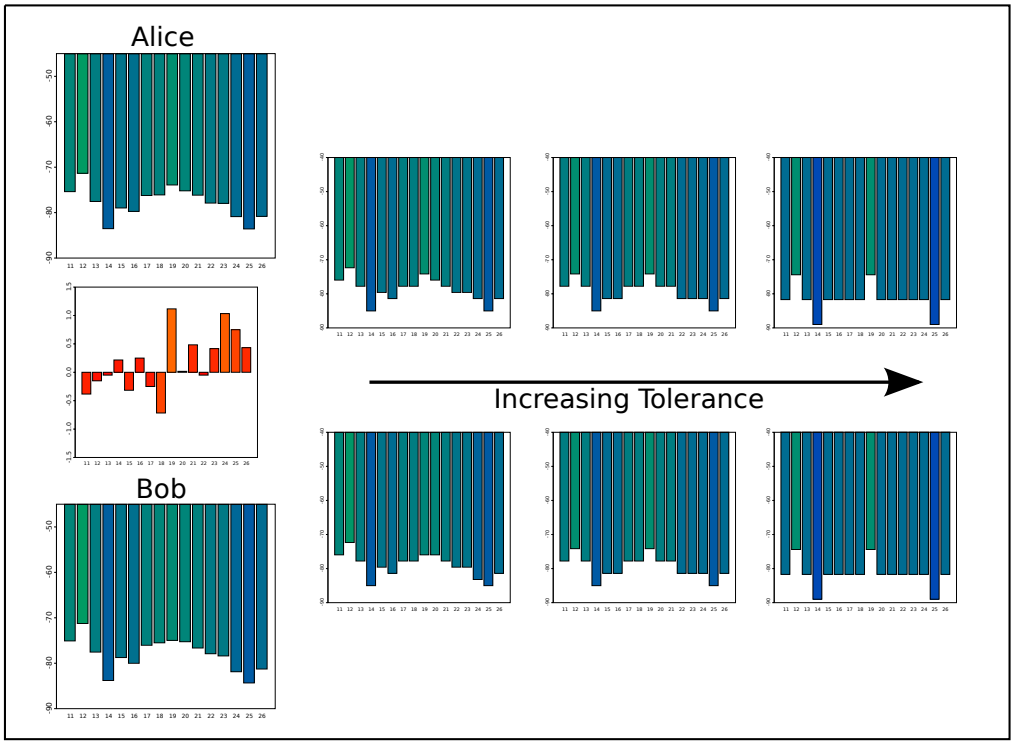


Figure 3.4: Quantizing of errors – impact of tolerances. Larger tolerance values are able to overcome larger deviations in the channel measurements, at the cost of the number of possible values after quantization.

way due to the quantization property of the code. Alice then sends P via a public channel to Bob, who uses P to generate the same codeword c using his measurement μ' (given $dis\,(\mu, \mu') < t$) by calculating $c = enc\,(\mu' + P)$. To prove the correctness, consider that when $dis\,(\mu, \mu') < t$, then $dis\,(\mu + P, \mu' + P) < t$, and thus $dis\,(c, \mu' + P) < t$. Finally, since the error-correcting distance of the code is t, $\mu' + P$ is encoded to c by Bob as well.

At this point, both parties share a secret bit string that can be used to support security services.

3.3 Protocol Specification

From these building blocks, we can now specify a key generation protocol suitable even for low-cost wireless sensor motes. Using the capabilities available on current wireless sensor platforms, the protocol describes the procedure starting from the sampling the wireless channel, and resulting in a secret bit string only available to Alice and Bob.

3.3.1 Protocol Phases

The protocol operates in three phases. In the sampling phase, the channel state is acquired, and due to the reciprocity of the wireless channel, state information with bounded deviations is collected by the two legitimate parties in the protocol. In the key generation phase, these deviations are corrected, resulting in a secret bit string that is guaranteed to be equal if the experienced deviations are bounded and a suitable correction code is used. The final phase ensures that the secret is equal on both side. If this is not the case, then the protocol can be resumed in the second phase, that is, the channel measurements can be reused for a new key generation attempt. When the secret is found be equal, then the protocol has finished successfully. The complete protocol is shown as pseudo code in Protocol 1.

3.3.1.1 Sampling Phase

In the *sampling phase*, the signal strengths on all n channels are estimated by taking the means of the measured RSS samples $m_i^{(j)}$, resulting in the vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ for Alice and $\boldsymbol{\mu'} = (\mu'_1, \ldots, \mu'_n)$ for Bob. These vectors capture the channel state between two sensor motes at this point of time. Due to constraints in the hardware, the sampling messages must be exchanged in an interleaved manner, which can lead to deviations in the measurements,

Protocol 1 Key Agreement

Sampling Phase: k probes are exchanged on each channel in $\mathcal{F}$ to estimate the means of the received signal strength μ_i.

Key Generation Phase:

1. For each channel $f_i \in \mathcal{F}$:
   a) Alice chooses an appropriate error-correcting code $\mathcal{C}_{p_i}$.
   b) Alice uses error correction on the mean μ_i and produces the codeword c_i and a public string P_i for this channel.
2. Alice sends the collection of reconciliation strings $\mathbf{P} = (P_1, \ldots, P_n)$ to Bob.
3. Bob repairs his measurements μ' to encode to the same codewords $\mathbf{c}$ as Alice.
4. Both parties use the seed $s = bin(c_1) || \ldots || bin(c_n)$, the concatenation of the binary representations of the codewords.

Acceptance Phase: A challenge-response scheme is used to validate the secret.

1. In case of error: Further error correction can be used to repair the secret, so both parties start another key generation phase.
2. In case of success: Alice and Bob accept the secret.

along with additional sources of errors listed in the previous sections. Thus, a set of samples must be gathered to reduce the impact of temporal effects on the measurements. Special care must be taken that the two parties are synchronized during the sampling phase, as they cannot communicate if they reside on different channels. The implementation of the sampling behavior is described in the next chapter. After this phase, Alice and Bob each possesses a mean vector, $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ respectively. The vector of deviations $\boldsymbol{\mu} - \boldsymbol{\mu}'$ determines if the key generation in the next phase will be successful. If this is the zero vector, than an agreement has already been reached at this point. But this is very unlikely, and error correction steps must be taken in the next phase to guarantee a shared secret.

3.3.1.2 Key Generation Phase

In the *key generation phase*, a suitable code $\mathcal{C}_{p_i}$ with tolerance t_i is chosen to extract the maximum amount of entropy for each channel. Put differently, the code must successfully reconcile common information, while preserving all the available uncertainty in the outcome of the measurements. The tolerance must be larger than the deviation in the measurements for this channel to generate matching codewords, i.e., $|\mu_i - \mu_i'| < t_i$. It is possible to use a different parameter p_i for each channel f_i, depending on the expected error in the measurements. The errors can be predicted, as they are bounded (the results for this are presented in the next chapter), and some sources for greater deviations are known from practical experience. A very short distance between sensor motes, for example, leads to inhomogeneous electromagnetic fields, which in turn harm the reciprocity.

As all mean values are available to both parties, the correct codes can be chosen independently on each side, but identifiers of the codes can also be transmitted over the public channel to avoid different codes choices by Alice and Bob at this stage. After choosing the appropriate codes, Alice encodes each of her μ_i using $\mathcal{C}_{p_i}$ to create the tuple of codewords $\mathbf{c} = (c_1, \ldots, c_n)$, and creates the string $\mathbf{P} = (P_1, \ldots, P_n)$ to send it to Bob via a public channel. Given $dis\,(\mu_i, \mu_i') < t_i$ for all $i = 1, \ldots, n$, he can now recreate the same vector of codewords $\mathbf{c}$ by applying error correction and encoding his measurements $\boldsymbol{\mu}' = (\mu_1', \ldots, \mu_n')$. Both Alice and Bob can now calculate the same secret seed s by converting their codewords into a single bit string with length $|s| = \sum_{i=1,\ldots,n} p_i$. This is done with the function *bin* that uniquely maps codewords to bit strings, on which both parties agreed upon publicly before. This mapping can also be static and stored on the sensor motes at the time of manufacturing. The length of the secret directly depends on the number of channels and the chosen tolerance of the code. But increasing the number of channels is the better method to generate longer secret strings, as smaller tolerances require more precise measurements, which can not be influenced easily.

3.3.1.3 Acceptance Phase

Finally, in the *acceptance phase*, a challenge-response scheme ensures that the secret key has been created successfully. Alice sends a hashed version of her secret seed s, which ensures that no information is conveyed to an eavesdropper as hash functions are one-way functions that cannot be easily inverted. Bob uses the same hash function on his version of the secret to verify it by comparing the hashes. The only feedback given is if the key generation was successful,

but in case of error, the source of this error cannot be inferred. Most likely, the error happened in a limited number of channels. In case of failure, Alice can attempt to alter the tolerances by modifying some of the p_i in order to increase the odds that the next run will be successful. This must be done heuristically, as Alice and Bob cannot reconcile on the errors without revealing parts of their secrets. This brute-force approach should only be used if the number of different possible combinations is limited.

3.4 Conclusion

After a protocol run, both parties share a secret bit string that is unpredictable for an attacker. This advantage that Alice and Bob share can now be used to enable security services. By using error-correcting codes, the protocol ensures that an agreement can be reached. But the question remains on how much advantage this bit string offers, i.e., how many bits are completely unknown to the attacker and how many can be guessed. The second question addresses the robustness of the protocol: are the deviations bounded in real-world scenarios with interference from other devices and changing environments? The next chapter presents empirical results that answers these questions.

4 Experimental Analysis

After the definition of the key generation protocol, the next interesting aspect is how this protocol performs in real-world environments, and how large the achievable security can be, given realistic propagation properties. With several experiments, these properties are explored in detail in this chapter. It is also shown that the concept is applicable on resource-constrained devices and with realistic deviations in the wireless channel. The first part is focused on the robustness and performance of the protocol, and in the second part, the secrecy is quantified empirically. These insights are used as a basis and justification for the analytical model that is developed in the next chapter.

4.1 Testbed

This section describes the specifications of the used hardware, which represents the capabilities of common WSN sensor motes. The test environment where the experiments took place is also presented in this section.

4.1.1 Sensor Mote Hardware

The hardware platform used in the experiments is the MICAz sensor mote from Crossbow [14], shown in Figure 4.1. This platform is equipped with a Texas Instruments CC2420 transceiver chip compliant to the IEEE 802.15.4 standard, that is, it is designed for low-speed communication in the license-free ISM band (2400–2483.5 MHz). The radio supports a maximum data rate of 250 kbps, and uses DSSS with O-QPSK modulation[1]. Different output power levels are available, ranging from -25 dBm to 0 dBm. In initial experiments, which examined the possibilities of such key generation protocols, the effects of these power levels on channel propagation characteristics were evaluated. The experiments showed that the impact on the secrecy is negligible, as both parties must agree on the output power levels in order to ensure reciprocity. And in the end, an output power level of -25 dBm resulted in a RSS measurement reduced by 25 dB, the hypothesis that a weaker line of sight component enables richer

[1]Direct sequence spread spectrum with offset-quadrature phase shift keying modulation.

Figure 4.1: MICAz sensor mote with standard antenna used in the experimental analysis.

multipath fading was rejected. Thus, in the following experiments, only the highest output power level was used, which equals an output power of 1 mW. The sensitivity of the receiver, that is, the minimum power level of a signal that can be distinguished from the noise floor, is located at -90 dBm minimum, with a typical value of -94 dBm.

The hardware reports dimensionless RSSI values in the range from -60 to 40 that can be converted to RSS values (in dBm) using the linear relationship $RSS = RSSI - 45$. This relationship is not fully linear in practice, which must be considered when calculating received signal strength values. The procedure to correct these issues is described in Section 4.3.1. As the transceiver hardware uses phase-shift keying, the actual bit content of the signal has no influence on measured signal strengths.

The antennas used are included in the standard package, also shown in Figure 4.1, and are simple whip antennas consisting of a single insulated copper wire, with a length of 4 cm (a quarter of the wave length in the 2.4 GHz band). This antenna type is considered to be omnidirectional, because it radiates in a toroidal pattern in all directions perpendicular to the antenna. Only in the directions above and below the antenna no radiation is emitted. This design is very simply and offers only a small antenna gain. Yet, the experiments show that these antennas offer very good reciprocity characteristics, even if the antennas are not aligned to each other.

The processor is an Atmel ATmega128L 8-bit micro-controller with a processing frequency of 7.37 MHz, 4 kB of RAM and 128 kB of flash-based ROM. This hardware gives strict limits to the possible sampling rate of the presented key generation protocol, and requires small code sizes to ensure that the actual application has still enough space remaining to run. Communication with the environment is not only possible via wireless communication, the sensor motes are also equipped with a 51-pin expansion connector that provides (among others) a UART interface that can be used to communicate via serial connection. This connection can be used to received messages from the hardware, as well as for reprogramming purposes.

4.1.2 Test Environment

The experiments were conducted over several days on a university floor, that is, an indoor setting across several rooms. During the measurements, a wireless LAN access point was operating in the 2.4 GHz band, that is, the experiments are performed in a real-world environment with unpredictable factors. The environment contains concrete walls, as well as office furniture made of different materials. Especially metal objects such as shelves and cabinets with good reflection properties regarding electromagnetic waves are present. Thus, this test environment can be considered to generate rich multipath transmissions. Additional factors to this changing environment were movement from people in the corridors or office rooms.

Several different scenarios were considered to evaluate the impact of positioning to secrecy and robustness. A large meeting room was used for experiments where the sensor motes always maintained a line of sight connection, and several smaller office rooms were used to quantify the impact of shadowing objects and walls. In long-term and mobile scenarios both the mentioned rooms and the connecting corridors were used.

4.2 Sensor Mote Implementation

The implementations of Alice and Bob need an amount of 15.6 kB and 11.6 kB of ROM and 549 bytes respectively 361 bytes of RAM in the unoptimized MICAz implementation. The size in ROM is increased due to the need of a calibration table, as described in Section 4.3.1. This makes the protocol very space-efficient, the code can be deployed on sensor motes and the largest share of the memory is still available for the actual application code.

4.2.1 Implementation of the Sampling Process

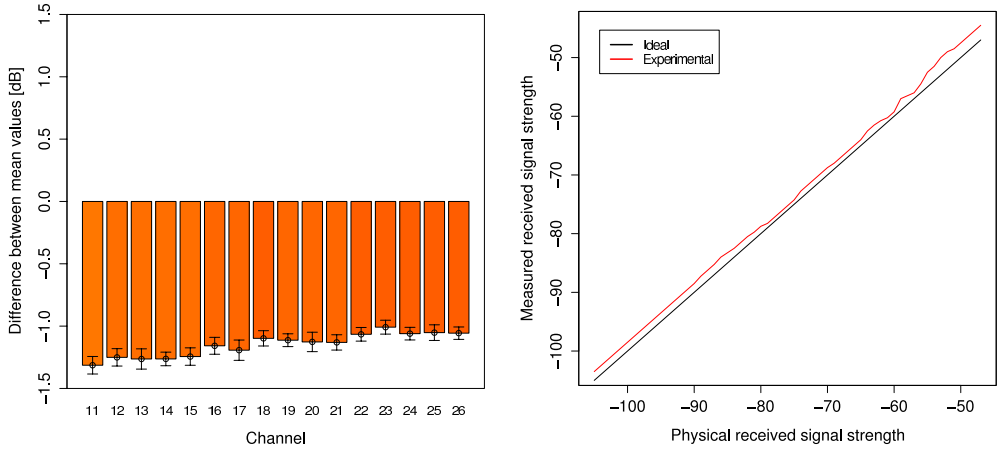
The sampling process is initiated and managed by Alice. She sends a sample message to Bob, who records the RSS value for this frame and send a reply, which in turn is measured by Alice. The node used as Bob was programmed to respond to queries for signal strength as fast as possible. This ensures that the answer is sent back during the channel coherence time, in which both parties are able to observe the same channel characteristics. When enough samples for a channel are collected, Alice initiates a channel switch and continues sampling until the measurement phase is complete. This process takes a large share in the overall time of a key agreement, with approximately 7.5 seconds.

To handle the event of a complete packet loss, a timeout mechanism is implemented that allows to skip channels with strong packet loss. If the sampling process for a single channel is not finished after 5 seconds, both nodes change to the next channel and try to recover from the connection loss. The channel on which this de-synchronization occurred is defined to have a signal strength of -95 dBm, that is, these channels can also be considered in the secret. Yet, this de-synchronization poses a large problem, as the two nodes can lose their possibility to communicate completely and have to restart the probing process on the default channel if they cannot recover from this state. Possible causes are the complete loss of all messages due to channel characteristics, but a loss of the channel switch messages is enough to have mismatches regarding the current channel. To mitigate the second source of errors, Alice can also switch back to the previous channel and rebroadcast the channel switch command in case Bob did not receive the previous messages.

The gathering and calculation of RSS mean values requires simple arithmetic operations, and a single data word of memory for each channel. To calculate the means, the RSS values can simply be summed, and if the number of samples is a power of two, a division by k is simply a shift to the right by $\log_2 k$. The error correction codes can also be implemented with binary right shifts if the tolerance values are restricted to powers of two as well. As the sensor hardware does not implement floating point arithmetic, it is advisable to follow these convention, but it is possible to use fixed-point arithmetic for arbitrary values.

4.3 Robustness Analysis

Previous research results show that the measurements of received signal strength are stable enough to support security services, even over long time periods (e.g., in [36, 55, 18, 10, 30]). Other researchers, however, consider the RSS metric to



(a) Uncalibrated measurements

(b) Calibration curve for the used mote pair

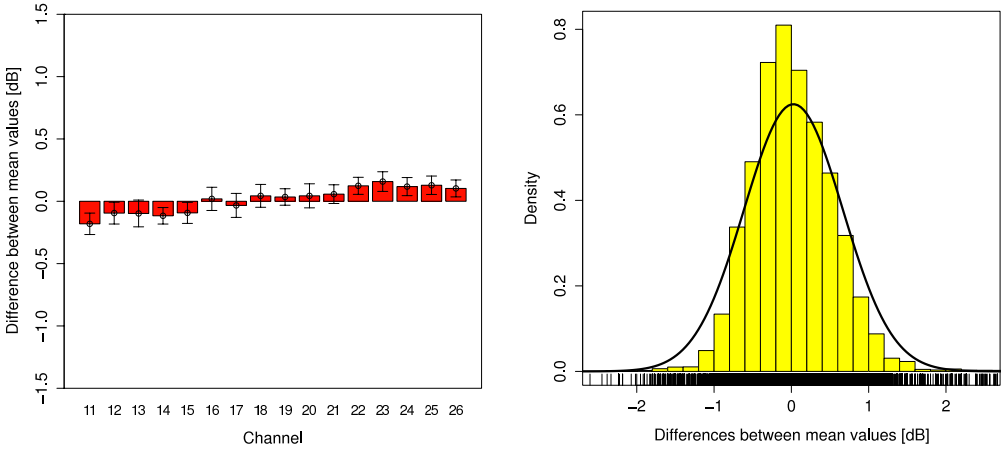
Figure 4.2: Calibration process of a pair of sensor motes.

be too unstable to be usable, especially in the context of localization [34, 13]. This section evaluates the robustness, that is, the success ratio of key agreements using the protocol in our wireless sensor testbed. We show empirically that the protocol performs very well under realistic conditions, mainly because the hardware provides accurate and stable measurements of the channel state. Similar experiments with IEEE 802.11 hardware show larger deviations, which can be explained by deviations introduced by the hardware and software drivers. Details on the WLAN implementation and the corresponding experiments are presented in Chapter 6.

In order to evaluate the robustness of the protocol, a total of 1600 positions of the two parties are tested, and the measurements and deviations between the two parties recorded. All stationary scenarios used for measurements are considered; a line of sight scenario in a single big room, a non-LOS scenario within several smaller rooms, and a continuous measurement with a large number of positions that considers many different distances and obstacles in the way. The deviations experienced show a non-zero mean, that is, a previous calibration is necessary, as the sensor motes perceived the signal strengths differently.

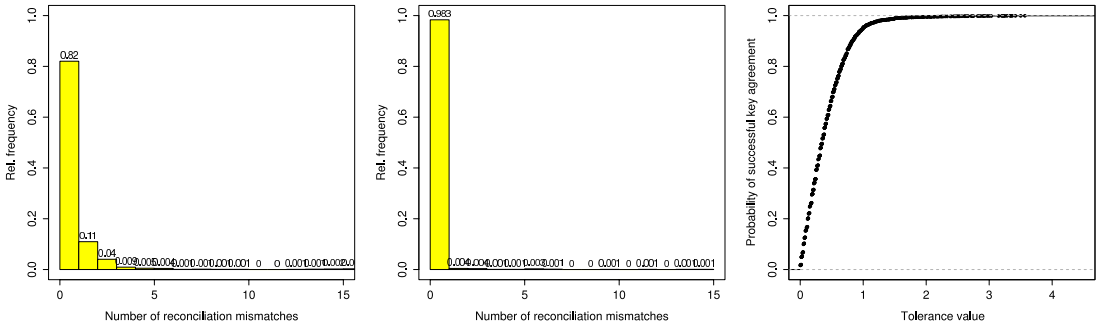
4.3.1 Calibration Process

The stationary deviations can be explained by the differences in transceiver hardware and antennas, and are different for every pair of nodes. The mean



(a) After calibration (b) Distribution of errors

Figure 4.3: Amount of deviations for all channels between Alice and Bob, based on measurements from 1600 positions.



(a) $t = 1$ (b) $t = 2$ (c) Empirical cumulative distribution function

Figure 4.4: Empirical robustness for different tolerances, based on 1600 positions.

values of the deviations, with a confidence level of 95% and based on all experiments, is shown in Figure 4.2a. The MICAz user manual points out that the relationship between received signal strength and the RSS value reported by the hardware is not linear, thus corrections must be made individually for each RSS value [14]. The calibration curve used in the experiments is given in Figure 4.2b. It shows the necessary shift on one sensor mote, based on the perceived deviations. The remaining deviations after calibration are presented in Figure 4.3a. There is still a skewness remaining, but the signal strength values experienced are now very similar. The different deviation means can be explained by slightly different power output levels of the transceiver hardware on different frequencies. From the implementation perspective, the needed calibration can be done before the sensor hardware is deployed. The need for calibration is therefore not a hindrance of the protocol's secrecy, as an on-line calibration that can be eavesdropped is not necessary.

4.3.2 Success Ratio

From the deviations observed, we can see that they are bounded. The histogram of deviations is given in Figure 4.3b, which also shows that these deviations are fitted well by a zero-mean Normal distribution after calibration, with a standard deviation of $\sigma = 0.638\,\text{dB}$ in the experiments. The distribution is even slightly narrower than the fitted Normal curve. Based on the experiments, we can conclude that the reciprocity of the wireless channel is very strong.

The success ratio of the protocol can be directly controlled by the tolerance values of the used code, as codes with larger tolerance values are able to correct stronger deviations. The experienced success ratio in the experiments is given in Figure 4.4. With a tolerance of 1 dB, 82% of the key agreements are successful on the first run. This value is increased to 98.3% with a tolerance of 2 dB. The empirical cumulative distribution function (ECDF), which is shown in Figure 4.4c, gives a continuous statement on the success rate. The majority of deviations are below 2 dB, and only a small number of extreme outliers were measured. These are caused if the connection was asymmetric with a RSS value in the range of -90 dBm, as then one node still received the sampling messages and uses this value, while the other node considered the channel as a complete loss, and uses the default value of -95 dBm for this channel. This causes large deviations, which can be mitigated with an adapted implementation. But as these events are very rare, they can also be safely ignored. As the chosen tolerance value also has an impact on the secrecy of the resulting bit string, a careful trade-off between secrecy and robustness must be found. The amount of this impact is evaluated in the next section.

4.4 Secrecy Analysis

To evaluate the secrecy generated by this protocol against the adversary described in Section 2.2.1, we conducted several experiments to quantify the uncertainty in the outcome of signal strength measurements. In the best case, the codewords are distributed uniformly, but as long as the distribution is sufficiently flat, i.e., no single value is the outcome in the majority of observations, randomness can be generated from these measurements in any case.

We evaluated the effects in two different environmental settings: *(i)* connections with a strong line of sight component only; and *(ii)* connections with obstacles in the direct connection, that is, non-LOS connections. For each of these scenarios, 250 positions were considered, and the distance was kept constantly at 2.5 meters to avoid strong path loss effects. The first experiment was conducted in the meeting rooms where the two motes are placed on tables that provided reflective surfaces, along with the surrounding walls. The second experiment was conducted across several small office rooms, with only sporadic LOS connections through open doors.

The LOS experiment was intended as a worst case scenario because a strong line of sight component is said to be able to dominate the multipath fading behavior. But our experiments shows that this is not the case, and both experiments yield roughly the same entropy. As a reference, the robustness experiment with varying distances was also evaluated for its secrecy. In all experiments, several different tolerance values were considered to show the impact of this parameter on the secrecy.

As pointed out in Section 2.2.3.2 on joint entropy, the total secrecy provided is reduced by the amount of inter-dependencies between the channels. Thus, the secrecy evaluation is split into two parts: in the first part, the secrecy is considered under the assumption that the channels are independent, i.e., the measured codeword are not correlated with each other. The second part tries to quantify the amount of dependence and gives estimations for the joint entropy on all measured channels. As the coherence bandwidth is large, it is likely that the measurements between neighboring channels exhibit dependencies of some kind. If this is the case, then the observation of one codewords influences the outcome for codewords in the next channel, and thus the joint entropy is reduced. But this estimation is not simple, as a large amount of empirical data is necessary to quantify the joint entropy. In Chapter 5, a different approach is proposed, which is based on distributions of signal strength values.

4.4.1 Codeword Distributions

The empirical codeword distributions give a first impression on the uncertainty in the outcome of a measurement. We considered the impact of tolerances on the secrecy in our experiments as well. Figure 4.5 shows three different experiments, using two different tolerance values. Here, all codewords are considered independently from the channel the codewords was generated on, that is, the shown histograms represent the measurement results of all channels. All experiments show a preference for RSS values in the range between -65 and -80 dBm, and both deep fades and very strong signals are less probable, but these event are not too infrequent as well. Non-LOS connections show less skewness than the LOS experiment, the RSS values are shifted towards weaker signal strengths, as the signals are attenuated because of objects in the line of sight. The histograms also show the effect of different tolerance choices: the distribution envelope remains the same, but the possibilities, represented by the number of bars, are restricted to fewer choices, which has an impact on the secrecy.

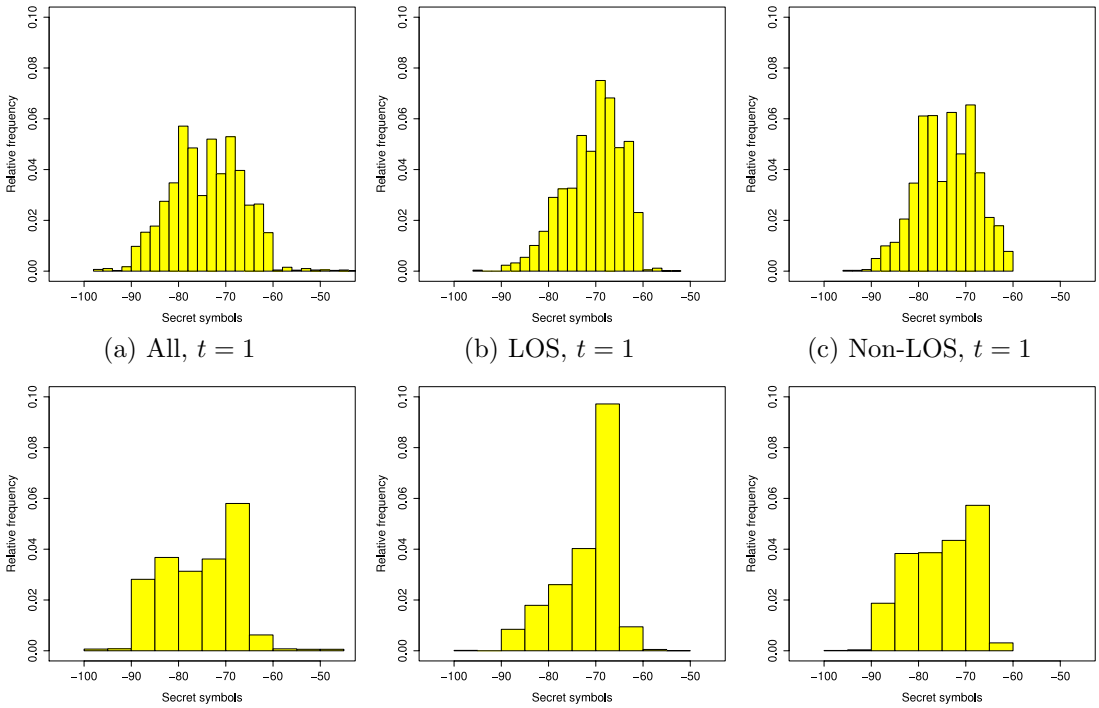
The empirical distributions show resemblance to the channel propagation models described in Section 2.2.2.1, but for the experimental analysis in this chapter, only the observations themselves are considered, the distributions of the RSS values are considered in the next chapter. These are the major sources of uncertainty, as they represent the effects of multipath propagation.

4.4.2 Entropy of Independent Channels

In this section, we assume independence of the measured codewords from different channels. In this case, the entropy values from individual channels can simply be summed to form the joint entropy. The entropy of single channels is determined experimentally by the histogram estimator

$$\hat{H} = -\sum_{c \in \mathcal{C}} p(c) \log_2 p(c),$$

with $p(c)$ being the relative codeword frequency. A total of three experiments is considered to quantify the available entropy; line of sight, non-line of sight, both with a fixed distance of 2.5 meters, and an experiment with changing distance and LOS characteristics. Three different tolerance values $t = \{1, 2, 4\}$ are considered to show the impact of this parameter. The empirical values based this method using these different experiments are collected in Table 4.1. The results show that the differences between the channels with respect to entropy values are very small, and that the chosen tolerance value has a strong



(a) All, $t = 1$ (b) LOS, $t = 1$ (c) Non-LOS, $t = 1$

(d) All, $t = 2$ (e) LOS, $t = 2$ (f) Non-LOS, $t = 2$

Figure 4.5: Distribution of codewords from different experiments, showing the impact of quantization for two different tolerance values.

Channel	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
LOS	3.42	3.49	3.46	3.41	3.54	3.49	3.49	3.54	3.50	3.55	3.48	3.55	3.49	3.53	3.40	3.57
Non-LOS	3.64	3.70	3.61	3.54	3.49	3.62	3.68	3.62	3.54	3.39	3.35	3.44	3.38	3.38	3.54	3.41
All	3.82	3.86	3.85	3.86	3.86	3.87	3.86	3.84	3.85	3.90	3.87	3.85	3.79	3.81	3.83	3.85

(a) Using tolerance $t = 1$

Channel	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
LOS	2.50	2.57	2.53	2.48	2.65	2.56	2.58	2.65	2.61	2.61	2.59	2.60	2.54	2.61	2.49	2.63
Non-LOS	2.72	2.77	2.70	2.59	2.61	2.69	2.80	2.70	2.60	2.49	2.46	2.50	2.54	2.54	2.63	2.52
All	2.90	2.93	2.93	2.93	2.95	2.92	2.94	2.90	2.89	2.95	2.94	2.91	2.84	2.89	2.89	2.90

(b) Using tolerance $t = 2$

Channel	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
LOS	1.63	1.70	1.70	1.64	1.70	1.70	1.75	1.72	1.70	1.72	1.75	1.75	1.65	1.66	1.57	1.73
Non-LOS	1.79	1.82	1.72	1.62	1.70	1.75	1.90	1.73	1.70	1.58	1.53	1.53	1.58	1.58	1.77	1.64
All	1.94	1.95	1.95	1.95	1.98	1.97	2.01	1.94	1.95	2.01	2.00	1.96	1.92	1.94	1.94	1.95

(c) Using tolerance $t = 4$

Table 4.1: Empirical Shannon entropy based on codewords for the different channels under assumption of independence, for different experiments and tolerance values (bit).

impact on the remaining secrecy. For the scenario with non-LOS connections, the entropy drops from approximately 3.6 bit to 2.6 bit and finally 1.6 bit every time the tolerance value is doubled. This is also in accordance with the intuition that if the number of possibilities is halved, then one bit is information is lost.

The cumulative entropy under the independence assumption is given by

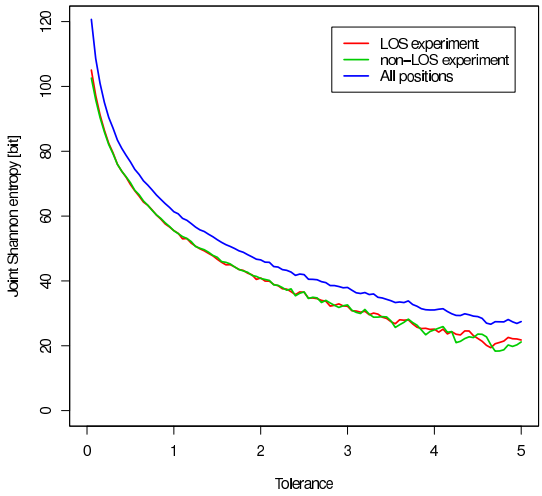
$$\hat{H}^n = \sum_{i=1}^{n} \hat{H}\left(\mathcal{C}_i\right),$$

which gives an upper bound for the joint entropy (total available secrecy) in the measurements. By increasing the number n of channels, we have a simple way to produce longer secrets.

The min-entropy evaluation for a single channel uses the estimator

$$\hat{H}_\infty = -\log_2 \max_{c \in \mathcal{C}} p(c).$$

The results of the cumulative analysis are given in Table 4.2. The LOS and non-LOS experiments show that the available entropies are very similar, and thus the type of the connection is not an important factor when considering the secrecy. The differences between the entropy values are in the order of one bit.



(a) Continuous analysis of the Shannon entropy

	$t = 1$	$t = 2$	$t = 4$
LOS	42.5569	28.8772	16.6041
Non-LOS	42.4487	32.4493	18.7394
All	48.8212	37.5010	23.0607

(b) Min-entropy

	$t = 1$	$t = 2$	$t = 4$
LOS	55.9041	41.2178	27.0705
Non-LOS	56.3189	41.8560	26.9246
All	61.6221	46.6558	31.4178

(c) Shannon entropy

Table 4.2: Cumulative Shannon entropy from 16 channels under the independence assumption in bit. The figure shows a continuous progression of tolerance values.

The min-entropy results show that this metric is a very conservative measure and underestimates the amount of uncertainty, but is included here, as it shows the lower bound of secrecy in the scenarios. The figure shows the impact of a continuous choice of tolerance values, ranging from 0.05 to 5 with a step size of 0.05. The entropy loss is the most severe with regard to tolerance values below $t = 1$, higher values do not have such a large effect. The instabilities for large tolerance values exist because the choice of codewords is very important, and the choices are varying for adjacent tolerance values.

Regarding the information content, we can observe that the resulting secret string has a length of 80 bit for a tolerance value of $t = 1$. As the entropy is lower than this value, this means that this string is not completely unpredictable, even under the independence assumption. For the experiment with all positions, the Shannon entropy of the string is 61.6 bit. Some of the bits in the secret seed are therefore redundant, as they can be predicted when the other bits are known. But the relationship is not obvious, every bit carries a fraction of the entropy. It is important to realize that a bit in the seed is not equal to a bit in a uniformly random string. Methods to extract this entropy and turn it into a string that cannot be distinguished from a uniform string are presented in Chapter 6.

4.4.3 Entropy of Dependent Channels

The entropy of dependent channels is hard to quantify, as the Shannon entropy operates on the knowledge of the underlying distributions, which are unknown in our case. A precise estimate of these distributions, that is, a large number of observations, is necessary to estimate the entropy precisely. This becomes increasingly difficult when considering joint distributions, as the number of needed observations increases rapidly. Yet, several tools are available to estimate the joint entropy from empirical data [50, 51, 47], although some problems were identified using these tools. The analysis presented here uses two different approaches: n-block Shannon, which tries to increase the amount of dependence with larger tuples of experimental data, and a construction complexity approach, which uses insights from the theory of information compression to find a shortest representation of the codewords, which also results in a maximum of entropy. For an analytical treatment of this problem, refer to the next chapter.

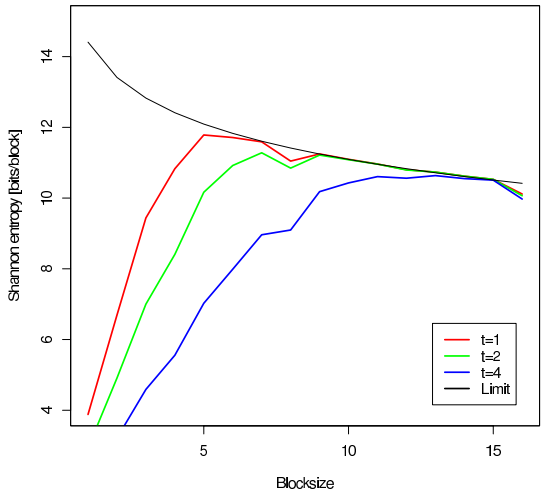


Figure 4.6: Shannon n-block entropy analysis.

4.4.3.1 Joint Entropy using n-Block Shannon

The Shannon entropy of several joint random variables can be estimated by calculating the relative frequency of longer and longer tuples of measurements [50]. The number n describes the size of the tuples, and $n = 1$ is equal to the independent channel analysis in Section 4.4.2. Here, the relative frequencies of codewords on single channels are considered. In the next step, the relative frequencies of tuples of neighboring codewords is considered, with increasing block sizes until the relative frequency of all 16-tuples is analyzed. Yet, this method has an important drawback: in order to show that the joint entropy of 16 channels is larger than for example 40 bit, more that 2^{40} tuples must be considered, and every tuple must appear exactly once to prove this level of secrecy using empirical measurements. Given the number of measurements collected in the experiments, a maximum joint entropy of 12 bit can be shown empirically using this method.

A graphical representation of the results are given in Figure 4.6. It shows the increase of entropy from different channels, using multiple tolerance values. The limit on secret bits that the experiments can prove is decreasing with increasing block sizes, as there are only a limited number of observations, and thus a smaller number of longer tuples. This bound is reached very fast, but judging from the steepness in the first part of the curves, the amount of joint entropy will be significantly larger that the value of 11 bit with a block size of 16. As expected, this method is not powerful enough for the evaluation of the joint entropy. An extrapolation of the curves, however, gives results that are

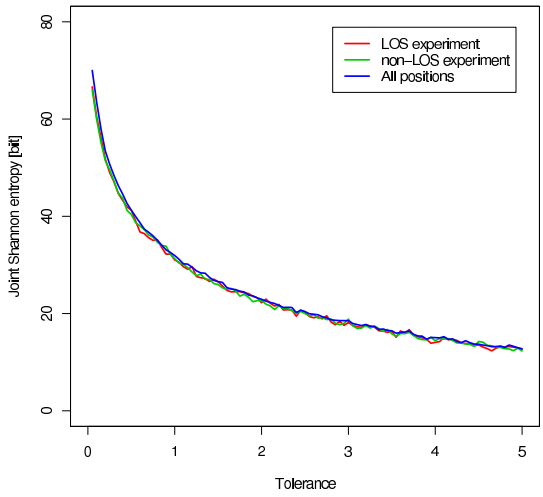
```
QMGECCCEEEEGIIIIOMGKOOQQSaWQOMKOOOOSSWSOOSUWUWOMOWSUKKSSOIEEEIGI
MMWKIMMGECCEIOUOAAEIMYQGGEOIEEEGIMSaOKGEKWEIQKIECECCEGKMMIGCAAAE
AGEEIEEEEIOOOWUQCEAAACCCEEEACCIOIIMMOOOMOQSaOIQOSSQUOMQQOOOIQWGE
ECEIOOQECCCSYSOQaSSQOOOSQOQIC99AOKKKIKMMMMIIIMOOGEKMMMOUOOQQSOSO
GCCEIKGECEECACOQSGEGIIOUOKMUQMMIEECGMSMMMOIGGGOOOSSWSOOSUWUWOMCC
CIEECEEEECCEEAOOQOMMOOQaQKKMMMEECEEEEECEEKIEGMOOUSKGCEKGIQSKECEG
EEUKCCCEIWOOUQOOOOQMQQOSQWWUWUACEGIOQMCACOIMOOOOMMOWMIMMIOQOOOQS
SQMIIKMMOACCAACIMCAAOSUMMMOOQaOKKMMMIMOMGGOKCECCGKAAAACAAACCECCE
SQOMMSUMKIKQQSOMQQMKKOQQOUSOOQQSCGECAAAGEEIOKKQaSMIIGGIOaOKMUQKQ
OMOMMOMKSQMMMIMQOMOMMOMKSQMMMIMWQOMQQWOUQOQOOEEIaMOIGKIMSICEGMME
CEKSOIMMGIIGIOOIIIKOUOQQQWQSaUa9ACAKIIKMOMMOGGEAACKEEAAACAEOIECK
QMOQUSOOOUUUSMIACCAACIMCAAMIGIIECEIMOOSMGGMEEGIMMIIOMGEGOMSaKEEE
ECCCEMOMOOEEAAACAEOIECMMIIGEGKIKQSKIKKEAAOIEACECAKGAACESMECCEECC
CCAEGEACEICACIIIIQSOMMIIOQOaQGCAAAEIIEEIOMMaSMOOMIKECEKGCEMKEAOO
OUaQSOMMQUSQMOCGKSECAACGQOIKEEIEGECGGGGGOMIEMKCCECAMEEIEGECGGGII
MKIMIEECGMSMMMOIGGGOOOGGIKKOOMOOOSOKEGIIOQMIGGGGGMOSQOSOOaMOGCCI
IOMAACGaKEEGMEAOOOUaQSOMMQUSQMOIIEAAEMMOMMQWQOACAAKEAAEGGGIIOOYS
```

Figure 4.7: Example of a T-string, the string representation of the codewords, with a unique character for every codeword. This is an example input with codewords encoded to ASCII characters. The string exhibits a pattern, that is, it is not fully random.

similar to the next method, with a joint entropy of roughly 26 bit for $t = 1$, 20 bit for $t = 2$ and 15 bit for $t = 4$.

4.4.3.2 Joint Entropy using Construction Complexity

An alternative approach to measure information content and entropy is given in algorithmic information theory [20]. Information theory and complexity theory are closely related, but have different notions for comparable statements [47]. The Kolmogorov complexity is a measure of the minimal space requirements to specify a string object, which is the reason why this notion is an important part of the theory of compression. The terms entropy and construction complexity are closely related, as the entropy also refers to the smallest number of bits that must be transmitted to send an arbitrary message, coming from a given distribution. But the major difference is that in the case of Shannon entropy, the distribution must be known in order make statements for individual observations. In the case of construction complexity, a single string object can



(a) Continuous T-analysis

	$t = 1$	$t = 2$	$t = 4$
LOS	30.9872	22.2208	14.6576
Non-LOS	31.2272	23.1968	14.3312
All	31.8832	23.2212	14.9168

(b) T-entropy for chosen tolerance values

Table 4.3: T-entropy for 16 channels from different experiments (bit). The figure shows a continuous progression of tolerance values.

be considered, even if the dependency structure and underlying distribution is unknown. As this is the case in our empirical data, this approach offers better insights into the joint entropy of our protocol.

Several different approaches to estimating the Shannon entropy can be considered: Lempel-Ziv 76-78 [29, 59] and the notion of T-complexity [51]. Speidel *et al.* [50] show in their work that T-complexity is the fastest to converge to the true value of Shannon entropy, and provide an algorithm that enables a fast computation of entropy values.

The tool `tcalc` [57], developed by the same group, was used to evaluate our results. As this tool operates on byte strings, we had to convert the lists of codewords to arrays consisting of different symbols as input to the tool. It then automatically adjusts to the number of possible symbols, that is, it alters the alphabet size automatically. Then, all of these characters were concatenated to form a large string that can be used as input to `tcalc`. An example of the resulting strings is shown in Figure 4.7. The ordering inside one measurement was retained, i.e., the strings have the form

$$c_1\, c_2\, \cdots\, c_{16}\, c_1'\, c_2'\, \cdots\, c_{16}'\, c_1''\, \cdots$$

This preserves the correlation structure of the strings, and results in an accurate estimation of the joint entropy. The results are collected in Table 4.3. For example, the available entropy in the measurements drops from 61.6 bit in experiment with independence assumption to a value of 31.9 bit for a tolerance value of $t = 1$. Therefore, the measurements in the different channels show some dependence, but the use of more channels adds to the joint entropy, that is, by adding a new channel available for measurement, roughly 2 bit of secrecy can be added. In the next chapter, we consider the possibility for longer secrets, given more channels, analytically.

4.5 Mobile Scenarios

We also evaluated the possibility to execute the protocol during device movement. In this experiment, the nodes are moved via different rooms and corridors in a random fashion, with changing objects between the two parties and different relative antenna positions. 32 samples were collected for each channel. These experiments exhibit a large robustness as well because the deviations are low, due to strong reciprocity. Figure 4.8 shows the similarities in form of a histogram of deviations. The histogram is also bell-shaped, but a larger number of outliers is present.

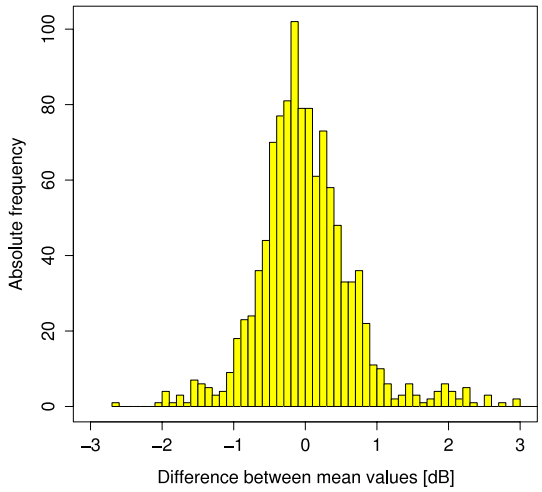
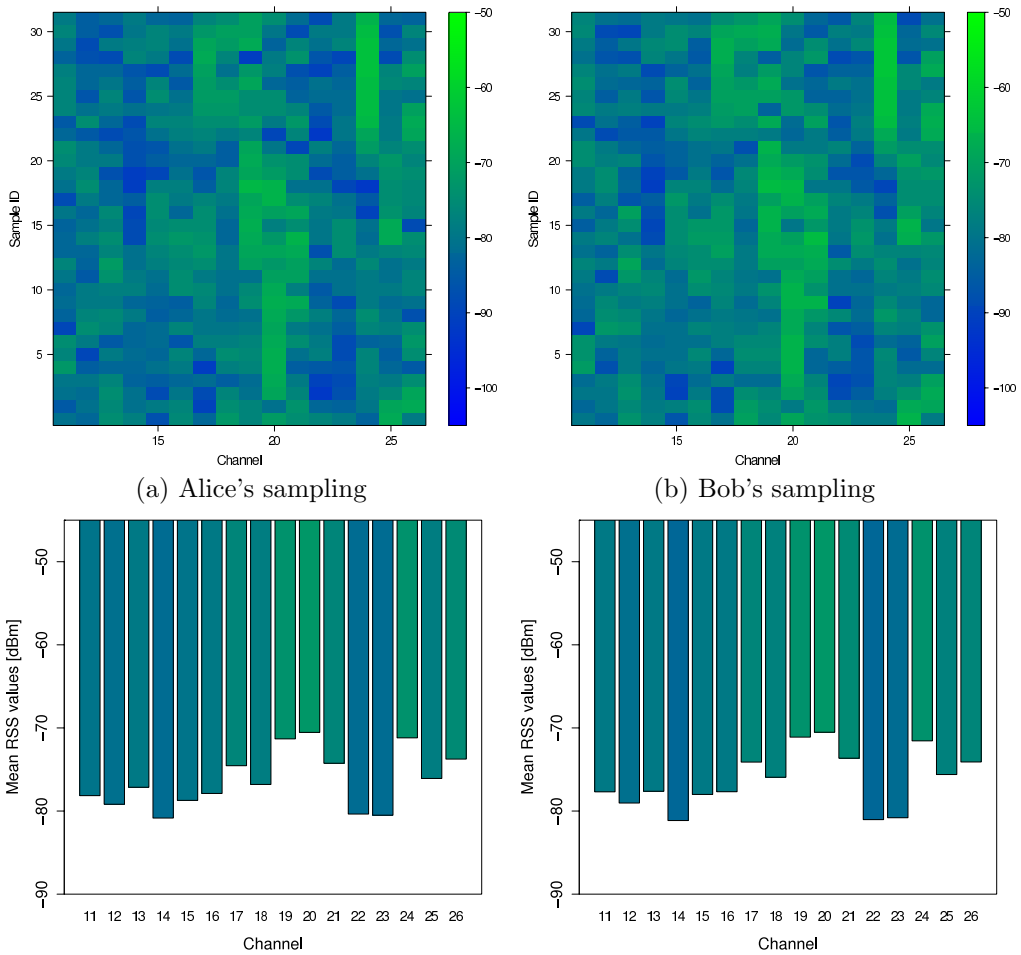


Figure 4.8: Measured deviations in the mobile scenarios.

In mobile scenarios, every sample messages experiences different wireless channel characteristics. Figures 4.9 a,b show the sampling process as a levelplot, in which every received signal strength of all sampling messages is shown. Every sample is affected by a change in transmission paths taken, and the measurements in the channels are not stationary as in the experiments before, but also highly variant. The calculation of mean values reduces the impact of temporal effects, and the resulting values are again robust (c.f. Figures 4.9 c,d).

4.5.1 Secrecy Considerations

No in-depth analysis of the secrecy of mobile scenarios is provided in this thesis, yet some observations can be noted. The sample size should be as small as possible to capture the entropy content of rare events such as deep fades. The use of mean values in the proposed protocol can hide these carriers of entropy, which results in a diminished amount of secrecy, because of the small number of possibilities that the mean can take. Figures 4.9 c,d shows that the use of different channels can be beneficial in mobile scenarios, but further analysis is necessary. As presented in the related work, a high sampling rate on a single channel increases the amount of secret bits that can be generated with mobile devices in the same way. A hybrid protocol of the one specified here, using a single sampling frequency, might be better suited for the case of mobile sensor networks. These adaptions are straightforward, but as other protocols are known that can be used in such scenarios, even though they also must be adapted to WSNs, this was not researched further in this thesis.



(a) Alice's sampling

(b) Bob's sampling

(c) Mean values of Alice

(d) Mean values of Bob

Figure 4.9: Mobile scenario - Sampling and resulting signal means.

4.6 Conclusion

In conclusion, the protocol performs well in terms of robustness, successful key agreements can be guaranteed both in static and mobile scenarios, as the tolerance values can be adjusted to the needs of the network. A large amount of secrecy can be extracted using this simple sampling method, even on low-cost hardware with standard antennas. This resulting secrecy is information theoretic secure, an adversary has no other possibilities but brute-force attacks, even if he has perfect knowledge of the distribution of RSS values. The amount of entropy in the measurements is limited only by the number of available channels in the IEEE 802.15.4 standard. Longer secrets can be generated if more channels are available. Judging from the joint entropy from the T-string analysis, roughly 64 channels suffice to form bit strings with 128 bit of entropy.

Figure 4.10 shows the collected empirical results of this chapter. The joint entropy under independence assumption is roughly twice the amount of the joint entropy of the T-analysis, as dependencies between the channels are present. The use of tolerance values below $t = 1$ have a potential to increase the joint entropy significantly, but the safer option is to choose a conservative tolerance and use a larger number of channels instead. A tolerance value of $t = 2$ is sufficient to almost guarantee successful key agreements, a further increase of tolerance is not beneficial.

4.6.1 Parameter Choices Revisited

In static scenarios, the number of samples determines the time the key agreement process takes, as well as the quality of the estimate of the mean. A larger number affects the secrecy in a positive way, as the means become more precise. Therefore, more information on the channel state is available, and it is also more likely that Alice and Bob have equal measurements. In mobile scenarios, however, this is not the case, as taking the mean of several samples hides rare events, which are the most unpredictable and thus most important events. This must be kept in mind when the parameters are chosen, the specific scenario must also be taken into account.

The tolerance values influence both robustness and secrecy. A trade-off must be made, it is possible to choose if a higher success ratio or longer secrets are required. Yet, as the deviations follow a Normal distribution, it is easy to find tolerance values that provides a suitable robustness-secrecy trade-off.

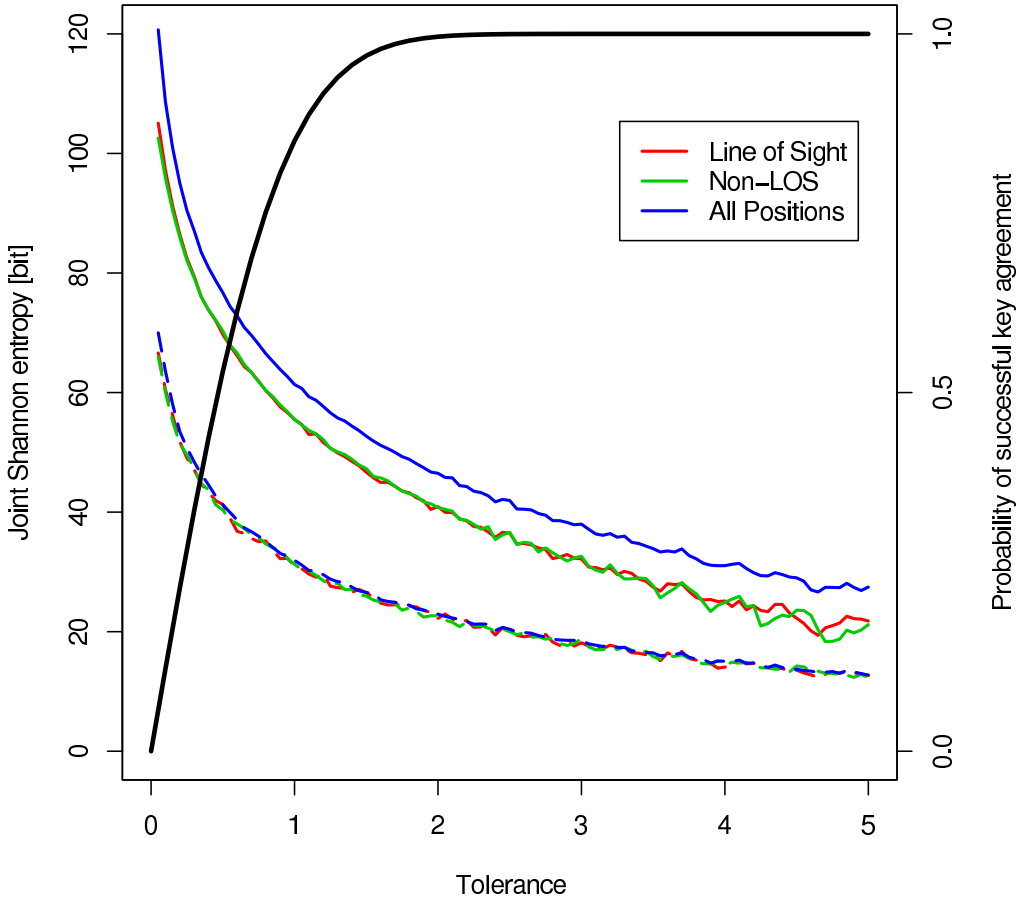


Figure 4.10: Collection of empirical results. The solid lines represent joint Shannon entropy values for three different experiments under independence assumption, the slashed lines represent the joint analysis based on t-entropy. The axis on the right side, along with the thick black line, show the success ratio of the protocol with respect to the tolerance values.

5 Analysis

In this chapter, we derive an analytical model based on the empirical data from the WSN experiments. The model must capture the amount of entropy correctly that the underlying signal strength distributions provide. The main obstacle is the correlation structure between the measurements on the individual channels. This structure must be captured by the model to generate reliable results. The models aims to answer the following questions: What can happen if the technology provides us with additional resources such as more channels and higher measurement precision? What happens if other RSS distributions are considered, for example, when examining outdoor scenarios?

5.1 Modeling Concept

The goal is to find a joint probability distribution that correctly captures the amount of entropy exhibited in the empirical data, but at the same time is applicable for any underlying signal strength distribution and parameter setting, such as the number of available channels. The concept used to derive the analytical model is the following: *(i)* find continuous distributions that describe the empirical signal strength values well, and estimate the required parameters of these distributions, *(ii)* hypothesize that the distributions of the RSS values are all following a single, joint distribution and estimate the necessary parameters, and *(iii)* show that the joint entropy which can be calculated for this distribution is accurate, that is, gives similar results compared to the experimental analysis. The following section shows how this concept is applied to derive the analytical model of the described key generation protocol.

5.2 Distribution of the Experimental Data

The secrecy and robustness characteristics of the protocol directly follow the distributions of the random variables concerned. As the deviations between Alice and Bob can be approximated well by a Normal distribution in all experiments, we can make statements on the robustness of the protocol in other

Channel	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
MLE	9.47	9.86	12.17	10.90	9.86	9.89	10.31	8.81	8.70	9.66	8.57	8.80	8.62	8.78	8.59	10.62
KS.test	10	10	13.75	13	10	10	10.5	8	8.5	10	8.25	8.25	9	9.5	9	10.5

(a) LOS experiment

Channel	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
MLE	10.14	10.20	9.75	9.21	9.82	10.18	10.78	9.84	9.73	9.94	8.59	10.27	10.33	10.39	9.40	10.31
KS.test	10	10	13.75	13	10	10	10.5	8	8.5	10	8.25	8.25	9	9.5	9	10.5

(b) Non-LOS experiment

Table 5.1: Parameter estimations of σ of the Rayleigh distribution for different experiments, using two different estimation methods.

Channels	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Rayleigh-MLE	4.59	4.65	4.96	4.80	4.65	4.66	4.72	4.49	4.47	4.62	4.45	4.49	4.46	4.48	4.44	4.76
Rayleigh-KS.test	4.67	4.67	5.13	5.05	4.67	4.67	4.74	4.35	4.44	4.67	4.40	4.40	4.52	4.60	4.52	4.74
Normal	4.54	4.62	4.63	4.49	4.68	4.62	4.69	4.76	4.65	4.63	4.67	4.69	4.56	4.62	4.51	4.63

(a) LOS

Channels	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Rayleigh-MLE	4.69	4.70	4.64	4.55	4.65	4.70	4.78	4.65	4.63	4.51	4.45	4.71	4.72	4.73	4.58	4.72
Rayleigh-KS.test	4.81	4.81	4.67	4.67	4.67	4.67	4.81	4.67	4.67	4.52	4.52	4.81	4.81	4.81	4.67	4.74
Normal	4.73	4.76	4.71	4.61	4.65	4.73	4.81	4.72	4.59	4.53	4.46	4.49	4.53	4.60	4.64	4.59

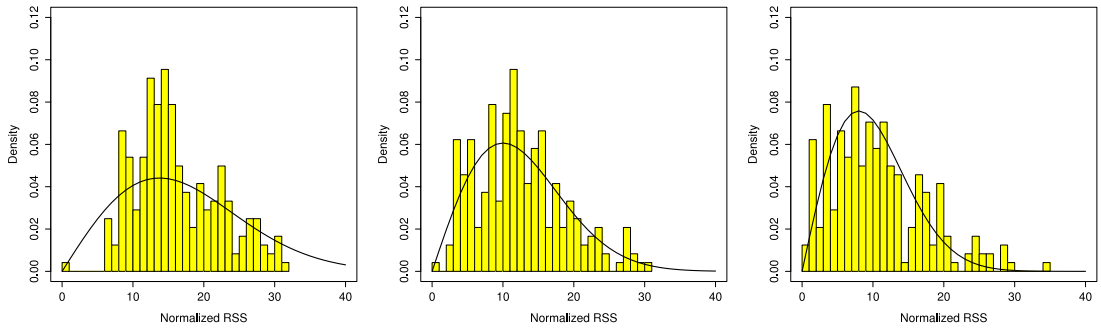
(b) Non-LOS

Table 5.2: Comparison of different distributions with respect to the differential entropy (bit)

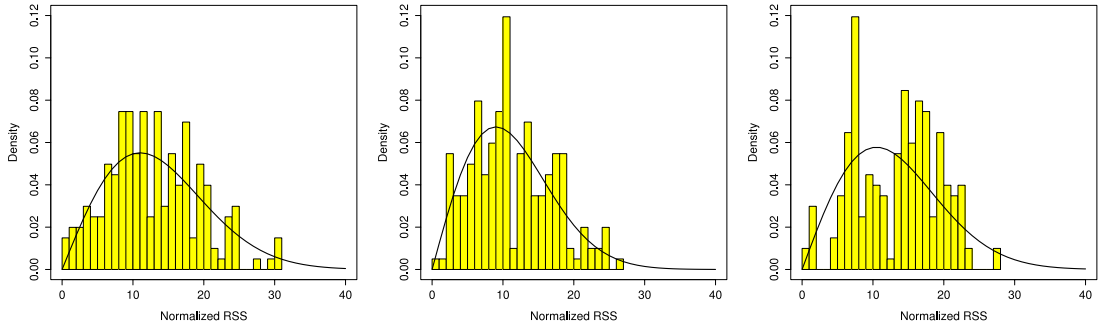
scenarios. But as the distribution of received signal strengths is strongly depending on the environment, a simple solution which fully describes the secrecy characteristics of the protocol is unlikely to find. In this section, we strive to find a model for the joint secrecy from several channels. It must support different RSS distribution, as well as tolerance values, to be as general as possible. In the first step, the experimental signal strength measurements of single channels are fitted to continuous distributions.

5.2.1 Distributions of RSS Values

Based on the probability distribution of the received signal strength values, we set out to fit a general model that exhibits a similar behavior with respect to entropy. Based on the visual appearance, the propagation models described in Section 2.2.2 seem to fit the empirical results. In order to keep the number of parameters low, we chose the Rayleigh distribution, as it seems general



(a) LOS - Fit on channel 13 (b) LOS - Fit on channel 15 (c) LOS - Fit on channel 18



(d) Non-LOS - Channel 11 (e) Non-LOS - Channel 20 (f) Non-LOS - Channel 26

Figure 5.1: Fitting the Rayleigh distribution to the RSS values from two experimental settings.

enough to fit all empirical findings. In order to justify this, we estimated the parameters for the Rayleigh distribution using two methods. In the first method, the parameter σ is estimated using the maximum likelihood estimator

$$\hat{\sigma} = \sqrt{\frac{1}{2N}\sum_{i=1}^{N} x_i^2}.$$

The estimated parameters using this method are labeled MLE in Table 5.1.

For a second approach to estimate the parameter σ, we used the Kolmogorov-Smirnov test (`ks.test` in R) on the empirical data to compare it to a sample coming from a Rayleigh distribution. This test is able to determine if two different samples are drawn from the same distribution. We compared our experimental sample with the output of `rrayleigh` from the R package VGAM with different parameters σ, which provides random values from chosen Rayleigh distributions. The parameters found, which lead to the acceptance of the Rayleigh hypothesis, are labeled KS.test in Table 5.1. In general, these estimates give a better visual fit than the parameters estimated using the MLE method. Most of the values lie in the range of 9-11, with the exception of two outliers that require a higher σ of 13 and 13.75, respectively. Figure 5.1a shows that there are outliers in the measurements which increase the parameter, and without these, a σ in the region of 10 would be the most fitting on these channels, as well. Figure 5.1 shows more experimental distributions of RSS values for selected channels, along with the fitted probability density function.

The Rayleigh distribution fits the data from the experiments very well, but how do the entropy of this distribution and the empirical distribution relate to each other? The entropy of the Rayleigh distribution, depending on the parameter σ, is given by

$$H_{\text{Ray}} = 1 + \ln\left(\frac{\sigma}{\sqrt{2}}\right) + \frac{\gamma}{2},$$

where $\gamma \approx 0.577$ is the Euler–Mascheroni constant. The entropy values are given in Table 5.2 for the line of sight and non-LOS experiments. The values represent the differential entropy, that is, this represents the entropy independent of tolerance intervals. This leads to values that are higher than, for example, the entropies of the empirical codewords in Table 4.1, as not quantization is used. Still, the entropy after quantization is comparable to the empirical data, the Rayleigh entropy is therefore a good candidate for an analytical model, and we can now consider a joint distribution that captures the correlation structure as well.

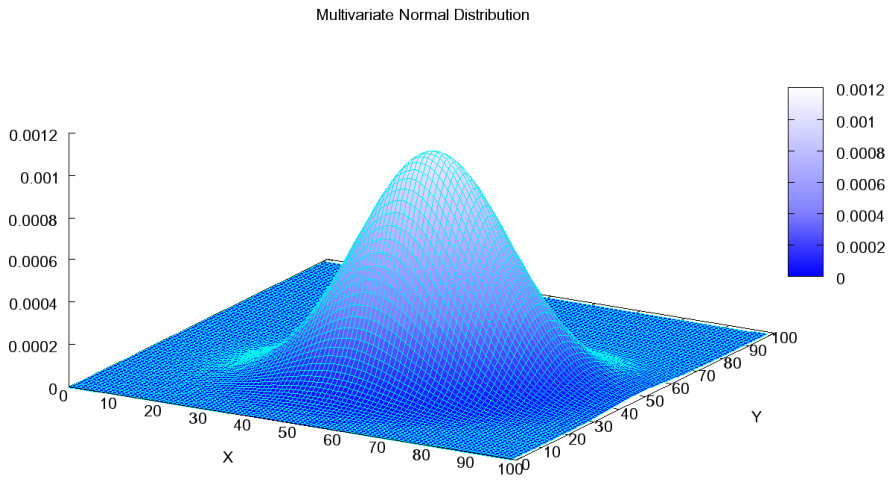


Figure 5.2: Example plot of two-dimensional multivariate Normal distribution.

5.2.2 Multivariate Distributions

The generalization of one-dimensional distributions to several dimensions are the multivariate distributions. They represent a joint probability distribution of several, possibly dependent, random variables. If a multivariate representation can be found, then the full inter-dependencies are captured, and we can make an accurate statement on the joint entropy.

Yet, most multivariate distributions become intractable even with a small number of dimensions, and unfortunately, the multivariate Rayleigh distribution is problematic in this respect. Reference [42] presents the analysis of the 3 and 4-dimensional Rayleigh distribution regarding their properties. It is clear that the n-dimensional case that is required for this analysis cannot be derived in the context of this thesis, especially with respect to the property of interest, the entropy of the joint distribution. A further drawback of most multivariate distributions is the fact that the correlation structure must be defined *a priori*, which is what we want to avoid, as this structure is unknown. This is not an issue for the Normal distribution, as the correlation coefficient completely describes the dependence structure. Other multivariate distributions considered were the Log-Normal, Exponential and t-distributions [25, 26], which show similar drawbacks.

Channel	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
LOS	5.76	5.99	6.04	5.48	6.42	5.97	6.27	6.61	6.09	6.03	6.19	6.28	5.76	6.00	5.55	6.03
Non-LOS	6.46	6.58	6.38	5.49	6.11	6.47	6.83	6.42	5.85	5.61	5.36	5.46	5.63	5.92	6.07	5.86

Table 5.3: Parameter σ of the Normal distribution for different channels

Thus, the only reasonable choice in the context of this thesis is the multivariate Normal distribution (confer Figure 5.2 for a two-dimensional example). The complete distribution has only two parameters, which capture the complete behavior: the vector of mean values $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^T$, and the covariance matrix Σ that represents the correlation between the different dimensions. If Σ is not singular, then the Normal distribution with the random vector $\mathbf{X} = (X_1, \ldots, X_N)$ can be described by the following probability density function:

$$f_X(x_1, \ldots, x_N) = \frac{1}{(2\pi)^{N/2}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right).$$

We can estimate these parameters with our experimental data to find an explicit representation of the correlation structure. A further advantage is that the function describing the joint entropy is available in closed form.

But first, we have to take a step back again to check the differences in entropy that this modeling choice produces.

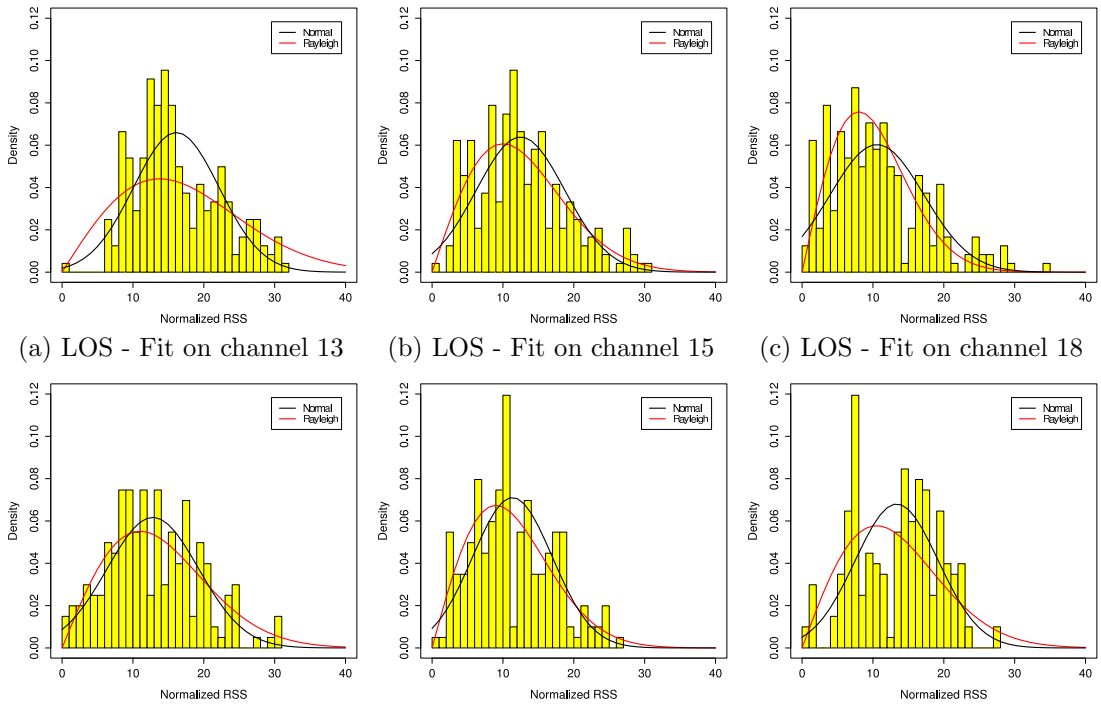
5.2.2.1 RSS Values and the Normal Distribution

We have to estimate the parameters μ, σ of the Normal distributions for every channel before we can proceed with the multivariate case. The parameter estimates of σ using the sample standard deviation are given in Table 5.3, and the graphical representation of the fits to the Normal distribution are shown in Figure 5.3, along with the Rayleigh fittings as derived in the beginning of this section.

With these parameters, we can now calculate the entropy values. The entropy of the Normal distribution (in natural units) is given by

$$H_{\mathcal{N}} = \ln \sqrt{2\pi\sigma^2 \mathrm{e}}.$$

Analytically, we can now derive a relation between the parameter of the Rayleigh distributions, for this discussion relabeled from σ to b, and the parameter that influences the entropy of the Normal distribution, σ. The goal is to determine



(a) LOS - Fit on channel 13 (b) LOS - Fit on channel 15 (c) LOS - Fit on channel 18

(d) Non-LOS - Channel 11 (e) Non-LOS - Channel 20 (f) Non-LOS - Channel 26

Figure 5.3: Fitting the Normal distribution to the RSS values from two experimental settings using the `ks.test` method. The Rayleigh distribution is also provided as reference.

the choice of parameters such that the entropy of the Normal distribution is smaller than the Rayleigh distribution. The derivation is as follows:

$$\begin{aligned} H_{\mathcal{N}} &< H_{\text{Ray}} \\ \ln\sqrt{2\pi\sigma^2 e} &< 1 + \ln\left(\frac{b}{\sqrt{2}}\right) + \frac{\gamma}{2} \\ \ln\sigma + \ln\sqrt{2\pi e} &< \ln b + 1.288 - \ln\sqrt{2} \\ \sigma &< 0.62b \end{aligned}$$

When comparing the estimated parameters for both distributions, we can see that the condition does not always hold. For example, a b-value of 10 requires σ to be lower than 6.2 to yield a smaller entropy for the Normal distribution, and some channels violate this relation. A comparison of the entropy values for the two fittings can be found in Table 5.2. We can observe that the resulting entropies from the Normal and Rayleigh fittings are very similar, although sometimes the entropy of the Normal distribution is slightly higher. Still, we believe that the Normal distribution still gives an accurate estimation of the entropy for the single channel case, and we can accept the Normal distribution for further modeling purposes.

5.2.2.2 Multivariate Normal Distribution

After we have validated the Normal distribution to be a good choice for the one-dimensional case, we can now extend the model to the 16-dimensional case from the WSN experiments. For this purpose, we again have to estimate parameters, this time, two multi-dimensional parameters have to be estimated: the mean vector $\boldsymbol{\mu}$ and the covariance matrix Σ of the multivariate Normal distribution. The mean vector can be estimated for each channel independently, and the non-biased maximum-likelihood estimator of the covariance matrix from a sample of k observations is

$$\hat{\Sigma} = \frac{1}{k-1}\sum_{i=1}^{k}\left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right)^T.$$

The results for the parameter estimation are given in Table 5.4 for both LOS and non-LOS experiments. The variance σ^2 of the independent channels is on the main diagonal of the covariance matrix. With increasing distance to the main diagonal, the values decrease, which shows a decreasing correlation with a larger distance in the frequency domain, i.e., if the channels spaced further apart, then the channels are only weakly correlated to each other. With a

11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
-69.41	-69.11	-68.82	-68.65	-69.34	-69.16	-69.39	-69.82	-69.74	-69.53	-69.67	-70.07	-70.20	-70.26	-69.91	-70.56

(a) Mean vector $\boldsymbol{\mu}$ - LOS experiment

Ch.	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
11	32.20	21.73	15.04	12.03	8.63	6.78	7.43	5.76	8.40	9.98	7.82	6.29	6.36	5.75	5.88	6.72
12	21.73	35.84	28.32	18.63	15.63	12.83	12.53	12.38	16.18	15.52	14.58	13.72	13.74	11.70	11.37	11.45
13	15.04	28.32	36.53	24.61	20.21	16.94	16.62	15.89	17.04	16.86	15.93	14.20	13.48	11.97	12.27	12.51
14	12.03	18.63	24.61	29.98	26.35	17.59	13.23	11.66	12.80	12.29	11.14	9.43	10.26	9.90	10.36	10.01
15	8.63	15.63	20.21	26.35	38.96	27.98	18.84	14.53	13.83	11.37	10.80	8.94	12.25	10.98	11.74	9.93
16	6.78	12.83	16.94	17.59	27.98	35.63	29.41	20.92	16.27	11.18	9.23	8.06	10.57	9.55	11.29	8.70
17	7.43	12.53	16.62	13.23	18.84	29.41	39.34	32.28	22.69	16.26	12.05	8.93	10.01	10.87	13.11	12.20
18	5.76	12.38	15.89	11.66	14.53	20.92	32.28	43.75	30.41	20.74	15.03	11.40	10.96	11.47	13.66	14.21
19	8.40	16.18	17.04	12.80	13.83	16.27	22.69	30.41	37.13	28.52	22.03	17.10	14.35	14.21	14.09	12.65
20	9.98	15.52	16.86	12.29	11.37	11.18	16.26	20.74	28.52	36.39	31.63	24.51	18.01	13.67	13.30	13.88
21	7.82	14.58	15.93	11.14	10.80	9.23	12.05	15.03	22.03	31.63	38.34	33.29	22.31	16.85	16.05	15.77
22	6.29	13.72	14.20	9.43	8.94	8.06	8.93	11.40	17.10	24.51	33.29	39.52	28.62	20.71	18.16	16.36
23	6.36	13.74	13.48	10.26	12.25	10.57	10.01	10.96	14.35	18.01	22.31	28.62	33.13	26.15	19.98	16.49
24	5.75	11.70	11.97	9.90	10.98	9.55	10.87	11.47	14.21	13.67	16.85	20.71	26.15	35.95	26.75	20.37
25	5.88	11.37	12.27	10.36	11.74	11.29	13.11	13.66	14.09	13.30	16.05	18.16	19.98	26.75	30.83	27.32
26	6.72	11.45	12.51	10.01	9.93	8.70	12.20	14.21	12.65	13.88	15.77	16.36	16.49	20.37	27.32	36.41

(b) Covariance matrix Σ - LOS experiment

Table 5.4: Estimated parameters for the multivariate Normal distribution

11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
-71.93	-72.76	-72.58	-72.29	-71.86	-72.12	-72.88	-73.11	-73.20	-73.37	-73.40	-74.01	-74.24	-74.27	-74.27	-74.30

(a) Mean vector $\boldsymbol{\mu}$ - non-LOS experiment

Ch.	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
11	41.78	33.47	26.05	23.83	20.44	20.01	18.30	14.71	11.68	8.57	10.18	14.51	13.57	14.84	16.17	13.10
12	33.47	43.28	34.73	27.27	24.08	20.62	17.80	13.84	10.45	6.66	9.85	14.50	15.24	16.57	16.56	13.35
13	26.05	34.73	40.70	32.17	25.73	22.63	18.06	13.35	9.01	6.19	9.15	15.80	16.45	17.77	16.90	14.37
14	23.83	27.27	32.17	35.30	29.75	25.74	21.10	14.94	10.13	8.34	11.04	15.43	16.31	16.40	14.67	11.07
15	20.44	24.08	25.73	29.75	37.39	33.38	27.42	17.64	12.10	9.99	10.32	11.18	14.23	15.73	14.48	10.27
16	20.01	20.62	22.63	25.74	33.38	41.89	38.35	25.58	16.73	12.09	11.02	10.95	13.26	14.97	13.87	10.11
17	18.30	17.80	18.06	21.10	27.42	38.35	46.61	35.36	23.42	14.65	12.60	11.67	12.66	13.03	14.04	10.27
18	14.71	13.84	13.35	14.94	17.64	25.58	35.36	41.17	29.30	18.39	13.91	13.31	13.12	12.03	14.03	10.42
19	11.68	10.45	9.01	10.13	12.10	16.73	23.42	29.30	34.25	26.09	18.48	13.77	11.22	10.52	13.56	10.89
20	8.57	6.66	6.19	8.34	9.99	12.09	14.65	18.39	26.09	31.48	24.30	14.85	10.82	7.73	8.86	6.72
21	10.18	9.85	9.15	11.04	10.32	11.02	12.60	13.91	18.48	24.30	28.76	21.31	14.66	11.24	11.04	7.94
22	14.51	14.50	15.80	15.43	11.18	10.95	11.67	13.31	13.77	14.85	21.31	29.78	25.00	19.92	18.89	14.56
23	13.57	15.24	16.45	16.31	14.23	13.26	12.66	13.12	11.22	10.82	14.66	25.00	31.68	26.24	21.97	15.89
24	14.84	16.57	17.77	16.40	15.73	14.97	13.03	12.03	10.52	7.73	11.24	19.92	26.24	35.03	29.27	21.27
25	16.17	16.56	16.90	14.67	14.48	13.87	14.04	14.03	13.56	8.86	11.04	18.89	21.97	29.27	36.81	29.95
26	13.10	13.35	14.37	11.07	10.27	10.11	10.27	10.42	10.89	6.72	7.94	14.56	15.89	21.27	29.95	34.39

(b) Covariance matrix Σ - non-LOS experiment

Table 5.5: Estimated parameters for the multivariate Normal distribution

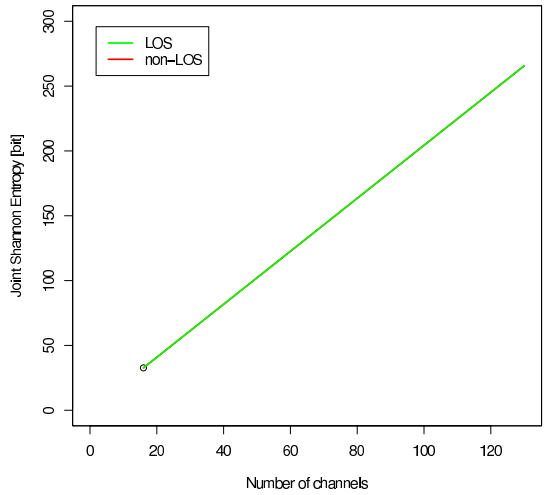


Figure 5.4: Secrecy depending on the number of available channels. The values are co-located in the figure.

broad frequency spectrum available, this can be used to increase the entropy, as dependencies are diminished.

5.3 Secrecy Analysis of the Multivariate Model

Now that we have derived an analytical model with a joint distribution which reflects the dependencies between the per-channel variables, we can make a statement about the joint entropy of the complete multivariate distribution.

5.3.1 Joint Entropy

The differential entropy of the multivariate Normal distribution with pdf f is given by

$$\begin{aligned} h_{\mathrm{mv}\mathcal{N}} &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(\mathbf{x})\ln f(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= \frac{1}{2}\left(N + N\ln(2\pi) + \ln\det\Sigma\right) \\ &= \frac{1}{2}\ln((2\pi\mathrm{e})^N \det\Sigma) \end{aligned}$$

With this closed form expression and the estimations for $\boldsymbol{\mu}$ and Σ, we can directly calculate the entropy values for the WSN measurements. The joint entropies are for the line of sight case 32.713 bit of entropy, and 32.695 bit for

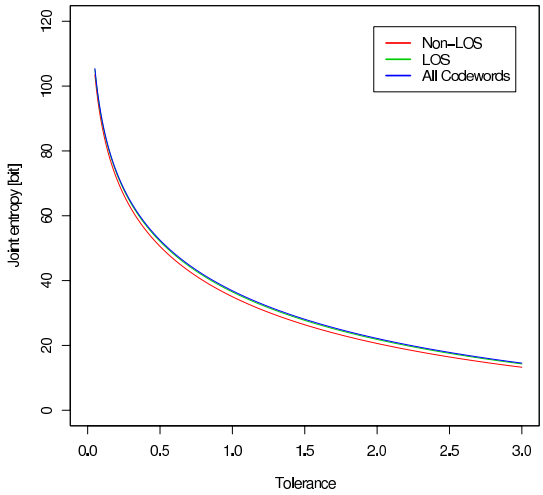


Figure 5.5: Comparison of different tolerance values, analysis based on the multivariate channel model.

the non-LOS case. The model therefore gives good results for the experimental data, especially in accordance to the t-entropy analysis, which resulted in 31.23 bit and 31.88 bit, respectively. Therefore we can extrapolate the correlation structure and analyze an arbitrary number of channel to provide strong secrets. This clearly shows that additional frequencies give a boost in available entropy.

Figure 5.4 shows the relation between the number of channels and the available entropy if we use a constant relationship between the channels, i.e., we leave $\det(\Sigma)$ fixed and change only the number of channels N. The relationship is linear, and an additional channel that can be used for probing provides roughly 2 bit of additional entropy. The gain in entropy is likely to be higher, because given the decreasing dependencies observed, the determinant of larger covariance matrices will also be larger.

5.3.2 Robustness and Secrecy Considerations

The impact of different tolerance values can now also be analyzed. The use of quantization changes the continuous signal strength distribution into a discrete distribution of codewords, which has a strong influence on the entropy values. Using the formula

$$u(f) = -\log\left(\int_{\mu_1-t}^{\mu_1+t}\cdots\int_{\mu_n-t}^{\mu_n+t} f(\mathbf{x})\mathrm{d}\mathbf{x}\right)$$

for different tolerance values t, the uncertainty an attacker experiences can be evaluated. Figure 5.5 shows the joint entropy of 16 channels with respect to

different tolerance values. Using small tolerance values, the joint secret in this analysis can be increased in length, but this comes at the cost of robustness. The best way for more secrecy is the use of more wireless channels.

5.4 Conclusion

This chapter described the derivation of an analytical model that captures the correlation structure between the individual random variables in our measurements. Using this model, it is possible to calculate the increase of secrecy for arbitrary distributions, as long as a multivariate Normal distribution with comparable entropy can be found. We have found that the described protocol scales linearly with the number of channels, and an arbitrary number of secret bits can be generated, given a sufficient number of usable frequencies.

6 Discussion

THIS chapter discusses some of the applications for the resulting secret string, as well as an outlook on technologies with characteristics different from wireless sensor networks.

6.1 Applications

There are several possible applications for the bit string that is produced by a protocol run. The uncertainty in the string is not uniformly distributed, and a bit in the string is not equal to a bit in a uniformly distributed string from a security perspective. This section presents some of the applications for our secret strings.

A first possibility is to use the string directly. By mixing it with a pre-distributed key, a new key can be derived that is unknown to an attacker even if the original keying material is compromised. Such a mixing is possible by, e.g., applying an XOR operation on the seed and a second bit string that is to be protected. This approach is related to security mechanisms that employ low-entropy sources of randomness such as password-based or biometric security [24]. The generated secret can therefore be used to amplify or protect a second secret. The rationale is to make secrets more "localized" and provide an additional obstacle to an adversary. This method is similar to *salting* in the context of password security, as the password is altered to avoid an easy breaking of the security of the password. This is the best application for the secret strings generated by the key generation protocol at the moment, as a secrecy of 30 to 40 bits is not sufficient in security sensitive scenarios. The direct application can also be used to support cryptographic primitives, i.e., it can be used as a seed for a (pseudo-) random number generator (RNG) or used as a nonce that can be kept secret and therefore contributes to the security.

A further idea is to use the topologies and the broadcast nature of wireless sensor networks to combine several short secrets to a longer one. The nodes communicate with each other by combining several available secrets from different links in a way that makes it hard for the attacker to infer the individual secrets. This can be done by sending hashes over concatenated secrets or by

XORing several secrets to one. This network-coding-like approach must ensure that the transmitted messages are not revealing too much information on the individual secrets. A related concept is presented in [45].

The most attractive application is to turn the non-uniform string into a perfectly secret, uniform bit string. There are several information theoretic tools that can be used to extract the randomness and form a uniform string. These tools are referred to as randomness extractors.

6.1.1 Randomness Extractors

Some constructions are known (e.g., [44, 23]) to be able to extract secure bit strings from random variables with a length in the order of the min-entropy. After extraction, the distribution of the result cannot be distinguished from a string drawn from a uniform distribution. The degree of distinguishably is given by the statistical difference. The statistical difference for the random variables A, B with the same support is defined as

$$SD(A, B) = \frac{1}{2} \sum_{\nu} |\Pr(A = \nu) - \Pr(B = \nu)| .$$

Privacy amplification can be used to make arbitrary distribution more uniform [7], and especially the notion of randomness extractors [16] enables the derivation of strong secrets:

Definition (Strong Randomness Extractor): Let $ext : \{0,1\}^{n_0} \rightarrow \{0,1\}^{l_0}$ be a polynomial time probabilistic function which uses r bits of randomness. We say that ext is an efficient $(n_0, h_{\min}, l_0, \epsilon)$-*strong extractor* if for all distributions W over $\{0,1\}^{n_0}$ with min-entropy $h_{\min}$ holds

$$SD((ext(W; X), X), (U_{l_0}, X)) \leq \epsilon,$$

where X is uniform on $\{0,1\}^r$, U_{l_0} is the uniform distribution on l_0 bit binary strings and SD is the statistical distance between two probability distributions.

This means that the input string of length n_0 can be turned into a shorter string of length l_0, but at the same time, the min-entropy $h_{\min}$ is preserved. The output string can only be distinguished from a uniform string with a probability of ϵ. Of course, the string is not more random as before, but a computationally bounded adversary cannot differentiate between real and "fake" randomness. This principle is also used for pseudo RNGs, where the pattern behind the output sequence is hard to guess by an attacker.

This is very attractive, as the resulting string can be used directly as keying material for cipher keys. But there are several drawbacks to this notion. As the entropy must be known a-priori, this must be estimated for every position. When the estimate is too conservative, then secret bits are lost in this scheme. If the estimate is too high, this can lead to a loss of secret bits, as the randomness extractor is operating on different assumptions on the nature of the input. Additionally, the cost in terms of secret bits is steep. Strong extractors can extract at most $l = m - 2\log\left(\frac{1}{\epsilon}\right) + O(1)$ nearly random bits [16], when m is the number of input bits and ϵ is the bound on the statistical difference. ϵ must be chosen very small to result in a key that truly can be called random from an adversarial point of view. Researchers suggest an ϵ in the order of 2^{-80} to generate keys with 128 bit of secrecy [43]. For a key length of $l = 128$ bit, this means the input string $\{0,1\}^m$ must have a length of at least 288 bit. Another publication [12] suggests that even this figure is not conservative enough and that 292 bit of entropy are needed to achieve an overall security level of 80 bit. This amount of secret material is clearly not achievable with WSN technology as it is. In our experiments, we have showed that 30 to 40 bit of entropy are available on the limited number of 16 channels, and therefore these tools are not usable in our context.

6.2 Other Technologies

Wireless sensor network motes benefit the most from such key generation schemes, but other hardware platform can be considered as well to evaluate the general applicability of the protocol.

6.2.1 Wireless LAN (IEEE 802.11)

The IEEE 802.11 standards *a, b,* and *g* offer an increased number of channels in the 2.4 and 5 GHz range. A total of 54 channels is available, which can already provide enough entropy for strong secrets. Channels 1 to 11 are in the 2.4 GHz range, and are equivalent to channels 11–22 in the IEEE 802.15.4 standard. But a number of additional channel in the 5 GHz band, separated by 10 MHz or even 20 MHz are also usable, ranging from channel numbers 34 to 165.

Several research contributions show that RSS measurements can be used to reach security goals. The signal strength can aid in localizing nodes to distinguish legitimate and adversarial nodes by their positions or hardware characteristics.

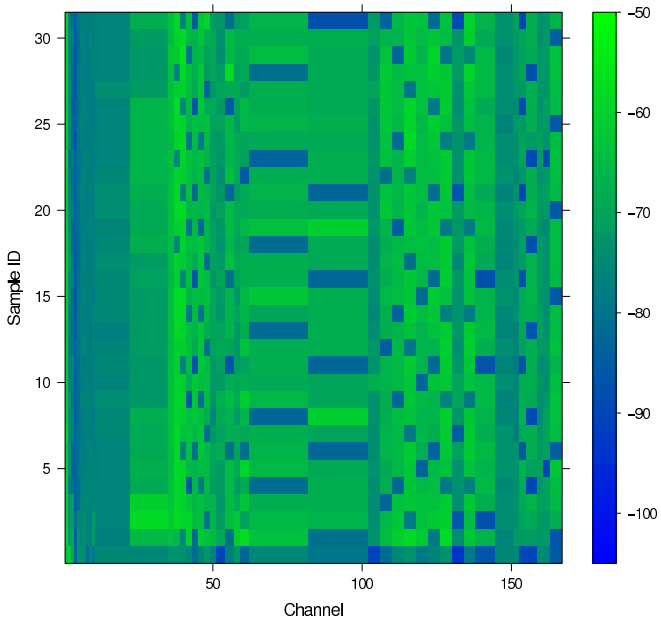
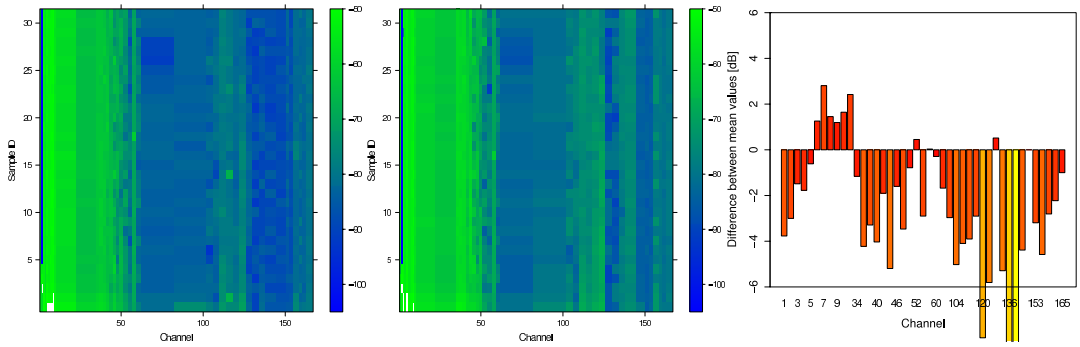


Figure 6.1: WLAN calibration effects in the sampling phase - the hardware driver start a calibration which causes a loss in perceived signal strength



(a) Alice's sampling (b) Bob's sampling (c) Differences

Figure 6.2: Measurements on the WLAN hardware platform.

On the other hand, some researchers found that the use of RSS values is not stable enough to rely on such measurements. Our measurement with wireless LAN hardware revealed the reason RSS measurements have such a bad reputation in terms of stability and reliability. Several aspects make current platforms unusable for stable measurements [27]. First, the hardware driver constantly re-calibrates the power levels of the sent frames. An example from our measurements is given in Figure 6.1. Every second, a drop in the measured RSS values can be noted, which is characteristic for Atheros-based cards. Second, multiple antennas are used in current laptop PCMCIA cards and build-in cards, which are both not omnidirectional and selected dynamically (the so-called antenna diversity). This seriously damages the channel reciprocity.

To address these issues, several modifications to the Atheros Linux device driver were mode to make the driver more usable for experimentation. Additionally, antenna diversity was switched off and a specific transmission power level was hard-coded for use. But the results in the end were not as satisfactory as the WSN based measurements. Figure 6.2 shows the deviations in the sampling process, and the resulting deviations in the mean. No calibration efforts were undertaken, but from the experiments a variance value of 5 dB can be assumed.

But once the benefits of such a key generation scheme are discovered, more hardware vendors and device driver manufacturers can integrate hardware that is designed for this task. If good performance can be achieved with cheap radio hardware such as the chips on the MICAz motes, then higher priced hardware should also be able to give better estimates for the channel state.

6.2.2 Cognitive/Software-defined Radio

Software-defined radios (SDRs) provide the possibility to have better access to the channel state. High-speed analog-to-digital converters (ADCs) used in this technology are able to generate a sequence of precise channel transfer property measurements. SDRs are not bound to defined frequencies, and the carrier frequency, signal bandwidth and modulation method can be chosen freely. This freedom can be used in our protocol to expand the set of frequencies which can be sampled to a sufficient number, even for the requirement of large secrets. Additionally, the channels can be chosen such that the spacing is large enough to assume independence, this make the analysis of the secrecy straightforward. The large sampling rate and the possibility to probe several channels simultaneously can boost the performance to make it on par with cryptographic methods such as elliptic curve Diffie-Hellman (ECDH), even on computationally strong hardware such as personal laptops.

Further network technologies will benefit from this evolution as well. Future wireless sensor networks will have a better access to the channel state, which will boost the usefulness of the proposed protocol further.

7 Summary

THIS chapter concludes the thesis and provides an outlook on possible future directions.

7.1 Conclusion

This thesis presented a novel key generation protocol that can be used to derive secret bit strings from the wireless channel even on resource-constrained hardware. This derivation of secret material is usable in both static and dynamic environments, which extends related key generation protocols to common scenarios found in WSNs. By implementation and experiments in a real-world wireless sensor network, the applicability is shown, and a thorough secrecy analysis showed that the concept can indeed be used to support security services. By providing a trade-off between the generated secrecy and the robustness of the protocol, it is possible to design a protocol that can generate as much entropy as possible with a very high success rate. This results in a large robustness that is able to overcome the natural deviations immanent in the physical channel. Experiments show that the protocol is able to successfully generate keys in over 95% of the cases, even in scenarios with frequent environmental changes or device mobility. The protocol can be implemented with a minimum of computational and memory requirements, and has been shown to work on typical wireless sensor network devices. The proposed key generation protocol is therefore a performance-aware alternative to key distribution methods in the literature.

7.2 Outlook

There are several interesting future directions that can make the protocol better. The sampling process in the protocol is uni-cast, i.e., the probing takes place for every single link in the network. But by exploiting the broadcast nature of the wireless channel, one sample message arrives at several sensor motes, which can be used to reduce the sampling complexity. This means that

the sampling process is only needed once per node instead of once per link. This way, the complete network can join in a very efficient key establishment phase in the beginning of its lifetime, while nodes joining the network at a later time can also use this property to generate keys with several other nodes in the network at once.

A second possible future direction is the use of adaptive error correcting codes, i.e., not all codewords have uniform distances. This can make the protocol more efficient, as currently only a small subset of the available codewords are used, and some occur very rarely. By using more codewords in the region of interest, it should be possible to increase the amount of secrecy without affecting the success rate.

The information of the physical layer can be used to support further security services. It can be used to detect impersonation attacks, as the signal strength can be checked for large deviations from the expected values. A security scheme that can leverage this data is presented in [35]. So, similar to keying material that can be used to derive multiple keys, the resulting "sampling material" contains location information that is hard to spoof, and can have the dual use of a source for shared secret information and a source for location-conditioned information to detect changes in the fingerprints of communication partners.

Nomenclature

RNG	Random Number Generator
BER	Bit Error Rate
DDH	Decisional Diffie-Hellman Assumption
DH	Diffie-Hellman Key Exchange
DSSS	Direct Sequence Spread Spectrum
ECC	Error Correction Codes
ECDF	Empirical cumulative distribution function
ECDH	Elliptic Cure Diffie-Hellman
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial, Scientific and Medical Radio Bands
LOS	Line of Sight Connection
MICAz	MICAz sensor nodes from Crossbow
Non-LOS	Non-Line of Sight Connection
O-QPSK	Offset-Quadrature Phase Shift Keying
PDF	Probability density function
RSSI	Reveived Signal Strength Indicator
RSS	Received Signal Strength
SDR	Software-defined Radio
UWB	Ultra-wideband
WLAN	IEEE 802.11 Wireless Local Area Network
WSN	Wireless Sensor Network

Bibliography

A

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A Survey on Sensor Networks. *IEEE communications magazine*, 40(8):102–114, 2002.

[2] Ross Anderson, Haowen Chan, and Adrian Perrig. Key Infection: Smart Trust for Smart Dust. *IEEE International Conference on Network Protocols*, pages 206–215, 2004.

[3] Tomoyuki Aono, Keisuke Higuchi, Makoto Taromaru, Takashi Ohira, and Hideichi Sasaoka. Experiments of IEEE 802.15.4 ESPARSKey (Encryption Scheme Parasite Array Radiator Secret Key) - RSSI Interleaving Scheme. In *IEICE Tech. Rep.*, volume 105, pages 31–36, Kyoto, April 2005.

[4] Babak Azimi-Sadjadi, Aggelos Kiayias, Alejandra Mercado, and Bülent Yener. Robust Key Generation from Signal Envelopes in Wireless Networks. In *CCS '07: Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 401–410, New York, NY, USA, 2007. ACM.

B

[5] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid. NIST SP800-57: Recommendation for Key Management – Part 1: General(Revised). Technical report, March 2007.

[6] Charles H. Bennett, François Bessette, Gilles Brassard, Louis Salvail, and John Smolin. Experimental Quantum Cryptography. *Journal of Cryptology*, 5(1):3–28, 1992.

[7] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Ueli M. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6):1915–1923, 1995.

[8] Dan Boneh. The Decision Diffie-Hellman Problem. In *Proceedings of the Third Algorithmic Number Theory Symposium*, volume 1423, pages 48–63. Springer, 1998.

C

[9] Seyit A. Çamtepe and Bülent Yener. Key Distribution Mechanisms for Wireless Sensor Networks: a Survey, March 2005. Technical Report TR-05-07 Renesselaer Polytechnic Institute, Computer Science Department.

[10] Yingying Chen, Wade Trappe, and Richard P. Martin. Detecting and Localizing Wireless Spoofing Attacks. In *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks*, pages 193–202, May 2007.

[11] Matthias Christandl, Artur Ekert, Michal Horodecki, Pawel Horodecki, Jonathan Oppenheim, and Renato Renner. Unifying classical and quantum key distillation. *Proceedings of the 4th Theory of Cryptography Conference, Lecture Notes in Computer Science vol. 4392, pp. 456-478, 2007*, February 2007.

[12] Yvonne Cliff, Colin Boyd, and Juan Manuel González Nieto. How to Extract and Expand Randomness: A Summary and Explanation of Existing Results. In *ACNS*, pages 53–70, 2009.

[13] Tran-Xuan Cong, Eunchan Kim, and Insoo Koo. An Efficient RSS-Based Localization Scheme with Calibration in Wireless Sensor Networks. *IEICE Transactions on Communications*, 91-B(12):4013–4016, 2008.

[14] Crossbow Technology Incorporated. Mote Processor Radio (MPR) Platforms and Mote Interface Boards (MIB), June 2006.

[15] Imre Csiszár and János Korner. Broadcast channels with confidential messages. *IEEE Transactions on Information Theory*, 24(3):339–348, 1978.

D

[16] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM Journal on Computing*, 38:97, 2008.

E

[17] Artur K. Ekert. Quantum cryptography based on Bell's theorem. *Physical Review Letters*, 67(6):661+, 1991.

F

[18] Daniel B. Faria and David R. Cheriton. Detecting Identity-based Attacks in Wireless Networks using Signalprints. In *WiSe '06: Proceedings of the 5th ACM workshop on Wireless Security*, pages 43–52, September 2006.

[19] Steven J. Fortune, David M. Gay, Brian W. Kernighan, Orlando Landron, Reinaldo A. Valenzuela, and Margaret H. Wright. WISE Design of Indoor Wireless Systems: Practical Computation and Optimization. *IEEE Computational Science & Engineering*, 2(1):58–68, 1995.

G

[20] Peter Grünwald and Paul Vitányi. Shannon Information and Kolmogorov Complexity, October 2004.

H

[21] Carl Hartung, James Balasalle, and Richard Han. Node Compromise in Sensor Networks: The Need for Secure Systems. Technical report, University of Colorado at Boulder, January 2005.

[22] Amer A. Hassan, Wayne E. Stark, John E. Hershey, and Sandeep Chennakeshu. Cryptographic Key Agreement for Mobile Radio. In *Digital Signal Processing*, number 6, pages 207–212, 1996.

[23] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A Pseudorandom Generator from any One-way Function. *SIAM Journal on Computing*, 28:12–24, 1999.

K

[24] John Kelsey, Bruce Schneier, Chris Hall, and David Wagner. Secure Applications of Low-Entropy Keys. In *ISW*, pages 121–134, 1997.

[25] Samuel Kotz, Narayanaswamy Balakrishnan, and Norman L. Johnson. *Continuous Multivariate Distributions*, volume 1: Models and Applications. John Wiley & Sons, New York, 1972.

[26] Samuel Kotz and Saralees Nadarajah. *Multivariate t-Distributions and Their Applications*. Cambridge University Press, February 2004.

[27] Karol Kowalik, Marek Bykowski, Brian Keegan, and Mark Davis. Practical issues of power control in IEEE 802.11 wireless devices. In *Proceedings of 15th International Conference on Telecommunications ICT'08*, 2008.

[28] Cynthia Kuo, Mark Luk, Rohit Negi, and Adrian Perrig. Message-in-a-Bottle: user-friendly and secure key deployment for sensor nodes. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 233–246. ACM New York, NY, USA, 2007.

L

[29] Abraham Lempel and Jacob Ziv. On the complexity of finite sequences. *IEEE Transactions on Information Theory*, 22(1):75–81, 1976.

[30] Zang Li, Wenyuan Xu, Rob Miller, and Wade Trappe. Securing Wireless Systems Via Lower Layer Enforcements. In *WiSe '06: Proceedings of the 5th ACM Workshop on Wireless Security*, pages 33–42, September 2006.

[31] An Liu and Peng Ning. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In *IPSN'08: Proceedings of the International Conference on Information Processing in Sensor Networks*, pages 245–256, 2008.

[32] Donggang Liu, Peng Ning, and Wenliang Du. Group-based key predistribution in wireless sensor networks. In *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*, pages 11–20, New York, NY, USA, 2005. ACM.

M

[33] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[34] Guoqiang Mao, Barış Fidan, and Brian D. O. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, 2007.

[35] Ivan Martinovic, Luc Cappellaro, Nicos Gollan, and Jens B. Schmitt. Chaotic Communication Improves Authentication: Protecting WSNs Against Injection Attacks. *Security and Communication Networks, Special Issue on Security in Wireless Sensor Networks, Wiley*, 2(2):117–132, March 2009.

[36] Ivan Martinovic, Frank A. Zdarsky, Matthias Wilhelm, Christian Wegmann, and Jens B. Schmitt. Wireless Client Puzzles in IEEE 802.11 Networks: Security by Wireless. In *Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec 2008)*, Alexandria, VA, USA, March 2008.

[37] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. Radio-telepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel. In *MobiCom '08: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pages 128–139, New York, NY, USA, 2008. ACM.

[38] Ueli Maurer. Protocols for Secret Key Agreement by Public Discussion Based on Common Information. In *Advances in Cryptology — CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 461–470. Springer-Verlag, August 1993.

[39] Ueli Maurer, Renato Renner, and Stefan Wolf. Unbreakable keys from random noise. In P. Tuyls, B. Skoric, and T. Kevenaar, editors, *Security with Noisy Data*, pages 21–44. Springer-Verlag, 2007.

[40] Ueli Maurer and Stefan Wolf. Secret-Key Agreement Over Unauthenticated Public Channels - Parts I-III. *IEEE Transactions on Information Theory*, 49(4):822–851, April 2003.

[41] Dominic Mayers. Unconditional security in quantum cryptography. *Journal of the ACM*, 48(3):351–406, 2001.

N

[42] Saralees Nadarajah and Samuel Kotz. On the Infinite Series Representations for Multivariate Rayleigh Distributions. *Communications, IEEE Transactions on*, 55(3):392–393, March 2007.

[43] Noam Nisan and Amnon Ta-Shma. Extracting Randomness: A Survey and New Constructions. *Journal of Computer and System Sciences*, 58(1):148–173, February 1999.

[44] Noam Nisan and David Zuckerman. Randomness is Linear in Space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

O

[45] Paulo F. Oliveira and João Barros. A Network Coding Approach to Secret Key Distribution. *IEEE Transactions on Information Forensics and Security*, 3(3):414–423, 2008.

R

[46] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

S

[47] Thomas Schürmann and Peter Grassberger. Entropy estimation of symbol sequences. *CHAOS*, 6:414, 1996.

[48] Claude E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423, 1948.

[49] Claude E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28:656–715, 1949.

[50] Ulrich Speidel, Mark Titchener, and Jia Yang. How well do practical information measures estimate the Shannon entropy? In *CSNDSP '06: Proceedings of the 5th International Symposium on Communication Systems, Networks and Digital Signal Processing*, pages 861–865. IEEE, 2006.

T

[51] Mark R. Titchener, Radu Nicolescu, Ludwig Staiger, T. Aaron Gulliver, and Ulrich Speidel. Deterministic Complexity and Entropy. *Fundamenta Informaticae*, 64(1-4):443–461, 2005.

W

[52] Arvinderpal Wander, Nils Gura, Hans Eberle, Vipul Gupta, and Sheueling Chang Shantz. Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks. In *PerCom '05: Proceedings of the third annual IEEE International Conference on Pervasive Computing and Communications*, pages 324–328, March 2005.

[53] Robert Wilson, David Tse, and Robert A. Scholtz. Channel Identification: Secret Sharing using Reciprocity in Ultrawideband Channels. In *ICUWB '07: IEEE International Conference on Ultra-Wideband*, pages 270–275, sep 2007.

[54] Aaron D. Wyner. The wire-tap channel. *Bell System Technical Journal*, 54(8):1355–1387, 1975.

X

[55] Liang Xiao, Larry J. Greenstein, Narayan B. Mandayam, and Wade Trappe. Fingerprints in the Ether: Using the Physical Layer for Wireless Authentication. In *ICC*, pages 4646–4651. IEEE, 2007.

[56] Yang Xiao, Venkata Krishna Rayi, Bo Sun, Xiaojiang Du, Fei Hu, and Michael Galloway. A Survey of Key Management Schemes in Wireless Sensor Networks. *Computer communications*, 30(11-12):2314–2341, 2007.

Y

[57] Jia Yang and Ulrich Speidel. A Fast T-decomposition Algorithm. *Journal of the UCS*, 11(6):1083–1101, 2005.

[58] Zhen Yu and Yong Guan. A Key Management Scheme Using Deployment Knowledge for Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(10):1411–1425, 2008.

Z

[59] Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, 1978.