Physical Layer Key Generation in Resource Constrained Wireless Communication Networks

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

Kemedi Moara-Nkwe

July 2020

Author's Declaration

I hereby declare that I am the sole author of this thesis.

Abstract

Secure wireless communication between resource constrained devices in dynamic deployment scenarios poses a significant challenge to cryptography. This is primarily due to the fact that the dynamic nature of the device deployment environment calls for sophisticated key management strategies which usually require a trusted third party along with either a highly complex symmetric key management scheme or a public-key scheme. This places a significant burden on the computational resources of a node. Physical layer security (or Information theoretic security) aims to reduce this efficiency burden on devices and add an additional layer of location-based security. Physical layer key generation and refreshment is concerned with techniques for establishing and refreshing cryptographic keys using wireless communication channel measurements between legitimate nodes.

Computational security-based public-key schemes usually derive their security on the basis of the difficulty of solving some mathematical problem such as prime number factorisation, discrete logarithm computation and the like. Practical physical layer-based schemes often derive their security on the difficulty of estimating particular wireless channel parameters with the exact same accuracy that a localised node can estimate them when you are not localised.

In this thesis, the issue of Physical Layer Secure Key Generation (PLSKG) is discussed and a novel pairwise PLSKG scheme and a novel Group Physical Layer Secure Key Generation (GPLSKG) scheme for resource constrained devices are proposed. The PLSKG scheme improves on the state of the art by proposing a key generation methodology that avoids the use of iterative quantisation for the purposes of key reconciliation, which reduces the loss of key entropy during the key reconciliation process. The proposed GPLSKG scheme improves on the state of the art by i) generating keys in a way that provides a means of evaluating and bounding the entropy of the generated key with respect to an adversary and ii) reducing the number of probes that need to be used for key reconciliation in certain deployment scenarios. The proposed schemes are then implemented on off-the-shelf devices and the performance of the schemes evaluated and compared to current state-of-the-art schemes. The schemes are shown to improve the performance of existing state-of-the-art PLSKG schemes and achieve near 100% success rates at short distances. The thesis also presents results on the error bounding in PLSKG schemes and presents results showing how these bounds can be used to make the key generation process more secure. Moreover, the thesis also discusses practical considerations in the design of PLSKG schemes, focusing on areas that have only received cursory treatment in current literature.

Acknowledgements

I would like to thank my supervisor Prof. Qi Shi for guiding me through this research and I would also like to thank Liverpool John Moores University for awarding me the studentship that allowed me to conduct this research. I would also like to thank my parents for all the help and support they have given me throughout my life and for giving the motivation to see this body of research through.

Contents

List of Tables

List of Figures

List Of Abbreviations

AES Advanced Encryption Standard

ECC Error Correcting Code

ECCs Error Correcting Codes

GKM Group Key Management

GPLSKG Group Physical Layer Secure Key Generation

ISI Inter-Symbol Interference

KDC Key Distribution Centre

LQI Link Quality Index

PFS Perfect Forward Secrecy

PHY Physical Layer

PLSKG Physical Layer Secure Key Generation

RF Radio Frequency

RSS Received Signal Strength

TTP Trusted Third Party

WSN Wireless Sensor Network

WSNs Wireless Sensor Networks

Chapter 1

Introduction

1.1 Physical Layer Security for Key Management in Resource Constrained Networks

PLSKG is concerned with the generation of a common cryptographic key between two or more wireless devices [4][5][6] [7]. A key application of physical layer based key generation is in resource constrained networks such as Wireless Sensor Networks (WSNs). This is largely because a particular pain point in securing WSNs is Group Key Management (GKM). In order for nodes clustered in groups to communicate securely, they need to establish cryptographic keys. The process to accomplish this task in a group scenario tends to be too cumbersome and inefficient to be practically applicable on resource constrained networks such as WSNs. Public key cryptography for instance, with schemes such as Diffie-Hellman based key establishment, provides an elegant solution for key establishment in many networks but is unsuitable for implementation on the small microprocessors that usually act as the main computing engine on resource constrained networks. This is due to the complexity of the exponentiation operations that usually form the basis of such schemes. Physical layer security based schemes aim to provide a more efficient and dynamic alternative to public key schemes. They also can be used in conjunction with upper layer cryptographic schemes in order to enhance security against remotely located attackers who would find it difficult to compromise a physical layer scheme as they are not localised.

A key desirable property of cryptographic key management schemes is Perfect Forward Secrecy (PFS). WSNs are often deployed in hostile environments and they can be susceptible to node capture where all the data in the node is compromised. In these instances, we want to minimise the impact of such security breaches. PFS ensures that even if an adversary obtains a session key and the long-term key in a node, the adversary cannot compute any other session key from that information. PFS essentially ensures that session keys stay independent of each other, so the knowledge of a session key or a continuous set of session keys does not help an adversary derive subsequent and/or preceding session keys. In order to enforce PFS in WSNs, we can use the wireless environment in which WSNs operate as a common source of randomness that can be leveraged to ensure secure pairwise communication between nodes by using random numbers generated over the physical layer as a base for key refreshment in the network. Utilising the physical layer in this way makes the key refreshment process very nondeterministic for remote attackers, hence making it difficult for the adversary to compute refreshed keys using past session keys or compute future session keys using current session keys.

1.2 Physical Layer Key Generation

Physical layer key generation is concerned with the generation of cryptographic keys via the observation of some stochastic random variable that is dependent on the physical layer being used between the communicating legitimate nodes. This communication can be done in the presence of an adversary who is not in the immediate locality of both the legitimate nodes during the initial physical layer key establishment process. The exact distance the adversary has to be away from the legitimate nodes is dependent on the particular physical layer scheme in use, the variability of the wireless channel and the minimum number of secret key bits that need to be generated per sample.

A typical pairwise PLSKG scenario can be seen in figure 1.1. Here two legitimate nodes, commonly referred to as Alice and Bob, wish to generate a physical layer cryptographic key in the presence of an adversary Eve. The legitimate nodes do this

by observing some parameter which characterises the physical layer channel that exists between them, h_{ab}. In an ideal scenario, h_{ab} is perfectly symmetric but in practical situations there are factors that make the channel not exactly the same but just highly correlated. These factors are numerous and include among other things: i) the fact that practical transceivers, due to variation in analogue circuitry, do not produce the exact same waveform output given the same input, ii) the time-varying nature of the channel that makes channel measurements taken at different points in time to vary, and iii) additive white noise (AWGN) that is due to time varying environmental interferences and electronic noise in the communication hardware.

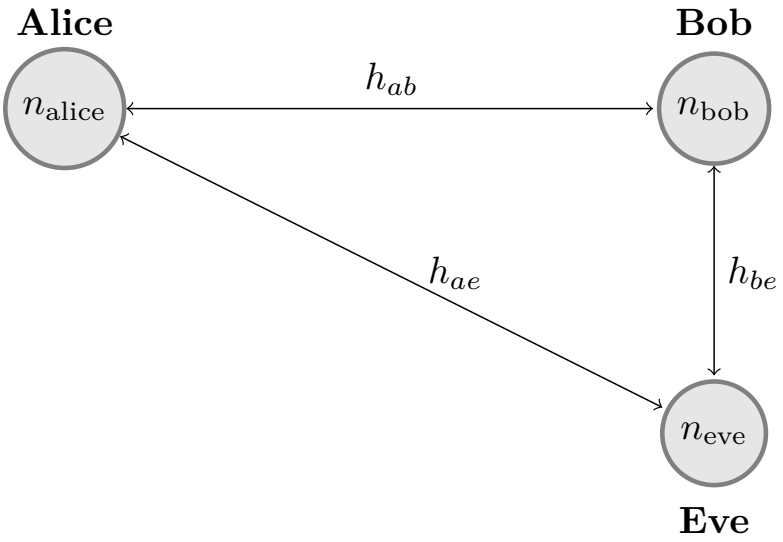


Figure 1.1: Channels between Alice, Bob and Eve

Factors such as the ones outlined above create disagreements in the measured physical layer keys that need to be carefully reconciled and characterised in practical key generation scenarios in order to minimise secret key leakage to localised observers. This thesis is primarily concerned with the design of practical key generation schemes for resource-constrained nodes in pairwise and group topologies that can fulfil these requirements. The thesis is focused on the generation of cryptographic keys in situations where the nodes are static (i.e. not moving) and can thus not depend on the channel variability that is brought above solely by mobility.

Research on PLSKG has been quite active over the past number of years and there has been a number of schemes proposed to achieve PLSKG. The vast majority of

the proposed schemes are meant for mobile environments and/or require special equipment to be used in order to increase the variability of the wireless communication channel. The variability is often introduced by using special antennas which randomly alter the antenna's Radio Frequency (RF) characteristics in order to induce variability.

A key shortcoming of these approaches is the narrow applicability and increased cost implications of implementing these schemes. The latter is especially true for schemes that require additional hardware to be installed and calibrated. The requirement of mobility is also restrictive as the nodes typically have to travel at speeds high enough for frequency dispersive fading to be significant enough for use in key generation.

The work in [1] is the first to consider the PLSKG on stationary nodes using what was defined as being the frequency selective nature of the channel. The scheme does not use a single, large bandwidth frequency selective channel but rather uses measurements taken over several relatively flat channels to generate keys. Due to the difference in noise levels and the difference in channel responses over the several channels, variability can be induced and that variability used as the basis for PLSKG. The scheme also considered PLSKG on off-the-shelf devices.

1.3 Aims and Objectives

The primary aim of this thesis is to explore mechanisms upon which resource constrained devices such as Wireless Sensor Network (WSN) devices can efficiently and securely transmit data. WSNs are networks that consist of a large number of resource-constrained, multi-functional and low-cost nodes deployed in a particular area of interest to observe particular physical phenomena. Because of the limited computational and power capabilities of the nodes in a WSN, it is important for all operations that run on a node to be of low complexity and high energy efficiency. Algorithms designed for WSNs should also be designed to be i) scalable because such networks can consist of thousands of nodes and ii) fault tolerant because the limited transmit power of nodes leaves them susceptible to transmission errors.

WSNs have been increasing in popularity in recent years and there is increasing interest from the network security community to devise schemes that will secure WSNs and thus ensure that such networks remain useful.

In particular, this thesis is concerned with techniques which exploit the environment where wireless devices are operating in to generate cryptographic keys which devices can then use to transmit information securely using symmetric encryption schemes. Generating keys in this manner is commonly referred to as generating keys at the Physical Layer (PHY) or PLSKG.

Figure 1.2 shows a typical five-layer network communication stack. The PHY refers to the lowest layer of the layered network communication stack and is the layer at which streams of bits received from upper layers are converted to analog waveforms which are then radiated from an antenna and made to propagate over the air to a remotely located receiver. The receiver then senses the waveform and attempts to estimate the waveform transmitted by the transmitter before proceeding to recover the original bits that the transmitter intended to communicate.

Current communication systems usually implement cryptography at higher layers such as the network, transport and/or application layer. Information at these layers is usually encrypted using a common cryptographic key and then passed on to the physical layer for transmission over the wireless medium. For this process to take place, legitimate nodes need a way to establish a common key to use for communication. This can be achieved via either some key distribution algorithm and/or by using some public key cryptographic scheme to establish the common key. In resource constrained environments, both these options can be very difficult to achieve. Physical layer based key generation schemes aim to ease this process by using the wireless channel to improve security.

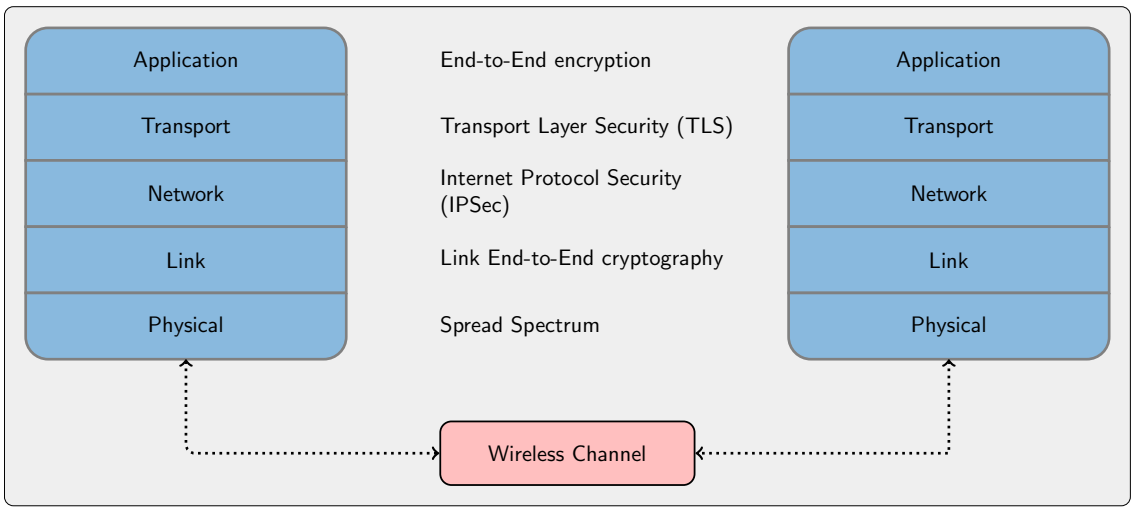


Figure 1.2: Figure Showing OSI 5 Layer Communication Network Stack

Whilst propagating from the transmitter to the receiver, the waveform radiated from the transmitter towards the receiver will get distorted. The transmitted waveform in this case is referred to as having being transmitted through a wireless communication channel. The wireless channel has been shown to be symmetric, a property which can be exploited for the generation of cryptographic keys. The symmetricity of the wireless channel means that a waveform radiated from some point A in space to another point B will suffer the same distortion as a waveform radiated from B to A. This is the channel reciprocity principle. At another point C, the received waveform will be different from the waveform received at B if C is not located at the same position as B. We can thus say that the channel from A to B, h_{AB}, is the same as the channel from B to A, h_{BA}, but dissimilar to the channel from A to C or B to C (i.e. h_{AC} or h_{BC}).

If legitimate nodes are at points A and B and an adversary is at point C then the condition that needs to be satisfied for some key generation capacity to be present and thus key generation to be possible is that $h_{\mathrm{BA}} \approx h_{\mathrm{AB}}$, $h_{\mathrm{BA}} \not\approx h_{\mathrm{BC}}$ and $h_{\mathrm{BA}} \not\approx h_{\mathrm{AC}}$. If the nodes keep refreshing their keys in every session then node capture would compromise the current session key but not the keys that were used in previous sessions. This is the key principle that underpins most physical layer key generation schemes. The difficulties of accomplishing this in practice will be discussed later in this thesis.

The work in this thesis shows that the way in which the key reconciliation stage is usually achieved in practical deployments guarantees that the entropy of the resultant key will be reduced in the key reconciliation stage with no way of bounding the entropy losses. The thesis then considers a pairwise PLSKG scheme that improves on the state-of-the-art scheme in a number of ways including the proposal of a novel key reconciliation process that uses the combination of careful quantisation and error coding control.

This thesis also extends the pairwise scheme to groups by proposing a group key generation scheme. Key difficulties in practical group key generation are the difficulties in i) showing that secret key generation capacity exists, ii) estimating it and iii) proving that the resulting scheme actually has been able to exploit that secret key capacity. It is often straightforward to check that the resultant keys have randomness properties but much less straightforward to prove that the resultant keys are theoretically secure.

This thesis paves the way for achieving the latter by showing that if the long-term correlation between all channels of concern are known then the group PLSKG scheme can be designed in a way to ensure that the keys have some entropy relative to an observer who is not collocated with respect to any of the legitimate nodes in the group. The key idea here is that if the correlation between legitimate nodes is known then the bit error between their quantised binary vectors can be both lower bounded and upper bounded. Similarly if the long-term correlation between legitimate nodes and adversarial nodes is known then the bit error can also be bounded. If this is achieved then the key reconciliation stage can be designed to be successful only if errors are in a particular range. This issue is discussed later on in detail in chapter 5.

This thesis also discusses several practical issues in implementing physical layer schemes on WSNs. A lot of factors that need to be overcome during the implementation stage of a PLSKG scheme have only received cursory treatment in current literature. This thesis aims to address this gap by providing some analysis on this front. In particular, the issue of the accuracy and linearity of physical layer mea-

surements can have significant implications on the final implemented scheme. This is covered in chapter 6.

The objectives of this thesis are therefore to:

- Discuss and illustrate the key ideas underpinning state-of-the-art PLSKG schemes and highlight the current shortcomings and challenges faced when trying to realise pairwise and group PLSKG schemes.

  The work relating to this objective is presented in chapters 3 and 2 where state-of-the-art PLSKG schemes and the key challenges currently facing the field are presented.

- Propose a novel pairwise PLSKG scheme that addresses some of the challenges faced by current PLSKG schemes. In particular, the proposed scheme addresses the issue of loss of entropy that occurs in the key reconciliation stage.

  The work relating to this objective is presented in chapter 4 where a novel and practical pairwise PLSKG scheme is proposed. The scheme improves on the state of the art by using error correcting codes (ECCs) to better address the issue of loss in entropy in the key reconciliation stage. Results from the implementation of the scheme on WSN nodes is also presented, evaluated and compared with the most relevant schemes in literature.

- Propose a group PLSKG scheme that allows collocated nodes to generate a common group key. The scheme improves on current schemes by proposing a key reconciliation method that allows the expected error between measured sequences at legitimate nodes to be upper bounded and the error at the adversary to be lower bounded. This allows the key reconciliation stage to be designed more securely.

  The work relating to this objective is presented in chapter 5 where a novel and practical GPLSKG scheme is presented. The scheme was implemented on off-the-shelf WSN devices, and the experiment results obtained from the implementation are presented, evaluated and compared to the state of the art.

- Discuss and highlight issues and difficulties in implementing PLSKG and

GPLSKG schemes on real off-the-shelf devices. In particular, the body of work in chapter 6 serves to illustrate how factors such the linearity and accuracy of the transceiver's Received Signal Strength (RSS) measurements affect the practical security of RSS based PLSKG schemes and also the practical design of key generation schemes.

1.4 Novelty

This thesis adds to the body of current knowledge in state-of-the-art physical layer key generation schemes by:

- Proposing a novel pairwise PLSKG scheme for resource constrained wireless devices. The proposed scheme is aimed at stationary nodes and uses off-the-shelf WSN devices. The scheme uses a combination of quantisation and error coding control to manage key reconciliation. This is in contrast to current state-of-the-art practical schemes which often use iterative quantisation in order to reconcile keys.

- Proving that schemes that achieve key reconciliation by iterative quantisation reduce the entropy of the resultant key with every iteration with no known method of bounding the entropy loss. This issue is discussed and the proof illustrated in chapter 4.

- Proposing a novel GPLSKG scheme for resource constrained wireless devices. The proposed scheme is novel as it provides a means of evaluating and bounding the entropy of the generated key with respect to the adversary. This is done by careful management of the quantisation and key reconciliation processes. Error bounding is possible in the case that the long-term correlations between all nodes are known.

  This information allows a concrete lower bound of the generated key's entropy to be established, something which is not possible with current GPLSKG schemes. The proposed scheme also reduces the number of probes required to generate a common GPLSKG key. This increases the energy efficiency of the scheme.

- Proposing novel measures that can be taken to reduce the impact of the non-

linearity and non-perfect accuracy of RSS measurements made at the physical layer. This includes proposing alterations to the proposed pairwise PLSKG scheme to provide extra resilience against these limitations.

Publications relating to the work in this thesis are listed below

- K. Moara-Nkwe, Q. Shi, G. M. Lee, and M. H. Eiza, "A novel physical layer secure key generation and refreshment scheme for wireless sensor networks", IEEE Access, vol. 6, pp. 11 374-11 387, 2018
- K. Moara-Nkwe and Q. Shi, "A practical physical layer group key generation scheme for resource constrained wireless devices", (In Preparation for Journal Publication), 2020.

1.5 Outline of Dissertation

This thesis is composed of 7 chapters. Chapter 1 introduces the thesis and outlines its main aims and objectives and the key properties of the physical layer cryptosystems that we wish to propose.

The thesis then goes over the fundamentals of physical layer key generation in chapter 2, this chapter provides some of the key fundamental results and introduces a lot of the terms and notations that will be used throughout the thesis.

A complete literature review of practical physical layer key generation is presented in chapter 3. This review goes over some of the most relevant state-of-the-art schemes, compares them with each other and highlights their advantages and disadvantages. The shortcomings of current PLSKG schemes, which have motivated much of the research work in this thesis, are also discussed.

The discussion on the pairwise PLSKG scheme that is being proposed can be found in chapter 4. This chapter first introduces pairwise key generation as a whole and then proceeds to propose a novel PLSKG scheme for wireless sensor networks in the chapter.

The proposed scheme is then followed by an extension to a novel GPLSKG scheme for straight line topologies which can be found in chapter 5. The chapter discusses the proposed GPLSKG scheme in the context of a straight line, 3 node network

and then explains how this can be extended to other arbitrary topologies. It then presents results of the implementation of the proposed scheme on a WSN testbed and evaluates the results.

Chapter 6 explores some of the key characteristics of off-the-shelf low power wireless devices and highlights some limitations and inaccuracies of RSS readings sourced from them. This chapter focuses on key properties of RSS readings in low power devices that have only received a very cursory treatment in the literature so far, such as the impact of linearity and accuracy on key generation. The chapter then presents processing techniques for overcoming this limitation.

Finally, chapter 7 summarises and concludes the thesis. This final chapter also outlines possible future work that could stem from this body of research.

Chapter 2

Fundamentals of Physical Layer Key Generation

Physical layer key generation is based on making observations of some random variable that is dependent on the wireless communication channel and using those observations to generate a secret key. In this chapter we will discuss the key ideas that underpin secure physical layer key generation and introduce key terms and notations that will be used throughout this thesis. We will also present sketches of key proofs that underly some of the core physical layer key generation theories. We start by considering the wireless communication problem and how it relates to physical layer key generation before moving on to the problem of PLSKG in a scenario where two legitimate nodes, n_{A} and n_{B}, wish to generate keys with no adversary present. After that, we then move on to the scenario where the legitimate nodes need to generate a key in the presence of an adversary Eve, n_{E}.

After these key concepts have been introduced, we will then move on to discuss the wireless communication channel in detail before moving onto highlighting which assumptions in theoretical analyses do no generally apply or are hard to realise in practice.

2.1 Physical Layer Key Generation Channel Models

In wireless communications, two nodes which do not share any physical connection radiate waveforms towards each other to communicate. The nodes can transmit information because they can encode information into the waveforms they radiate. When radiating a signal over the air, the waveform gets distorted. If a waveform x was transmitted, it can be shown that a waveform of form $y = h * x + n$ will be received where n is additive white gaussian noise (AWGN), the quantity h is a random variable called the channel response and $*$ is the convolution operation. It can be shown that h is a random variable. If the power of the transmitted waveform x is denoted as P_{T} and the power of the received waveform (i.e. the average receive signal strength) is denoted as P_{R}, then P_{R} can also be shown to be a random variable for constant P_{T}, a fact we will explore further in section 2.3.

If two nodes wish to generate cryptographic keys, they can thus transmit the same waveform at both ends and observe realisations of one or both of the random variables h and P_{R}. After this they can then use the observed sequence to generate a physical layer key, given that there is no adversary collocated to the legitimate nodes. If an adversary is not collocated to either node, then the adversary cannot estimate the realisations of h and P_{R} measured by the legitimate nodes with the exact same level of accuracy as they can, a pre-condition for there being some physical layer key generation capacity.

The situation outlined above can be mapped to a modelling framework and this can be done in a number of ways. A class of physical layer key generation models that will be used in this thesis is the single source models. A single source model models the process of the legitimate nodes observing realisations of h and/or P_{R} as a process of the nodes observing some common randomness source. A figure showing the single source model with no adversary can be seen in figure 2.1a. The legitimate nodes are named Alice and Bob.

In the single source model as shown in figure 2.1a, Alice and Bob are observing

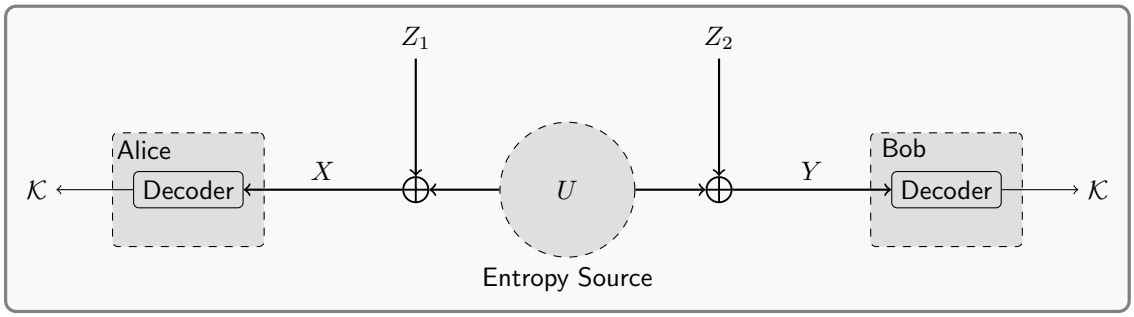
some random variables X and Y respectively. In the model, the realisations observed by Alice are drawn from a random variable X which is equal to the sum of the common entropy source U with another random variable which we denote as Z_1. Similarly, the observations sampled by Bob are realisations from a random variable Y which has been formed by summing the common randomness source U with another random variable Z_2. Z_1 and Z_2 are independent random variables and model various effects which may cause the wireless channel to not be perfectly symmetric. In the special case where the wireless channel is perfectly symmetric, then Z_1 and Z_2 are just constants.

The model as shown in figure 2.1a can be shown to be equivalent to the simpler model shown in 2.1b. In this model the observations at Alice are directly obtained from the common randomness source whilst the observations observed at Bob's end are drawn from some random variable which is equal to the sum of the common randomness source with another random variable Z, where $Z = Z_1 + Z_2$. Conditions on Z for key generation to be possible will be explored in the next section. It can be shown that for every observation observed at Alice and Bob, the two legitimate nodes can generate a maximum of $h(X) - h(Z)$ common bits, a result which follows directly from Shannon's channel coding theorem.

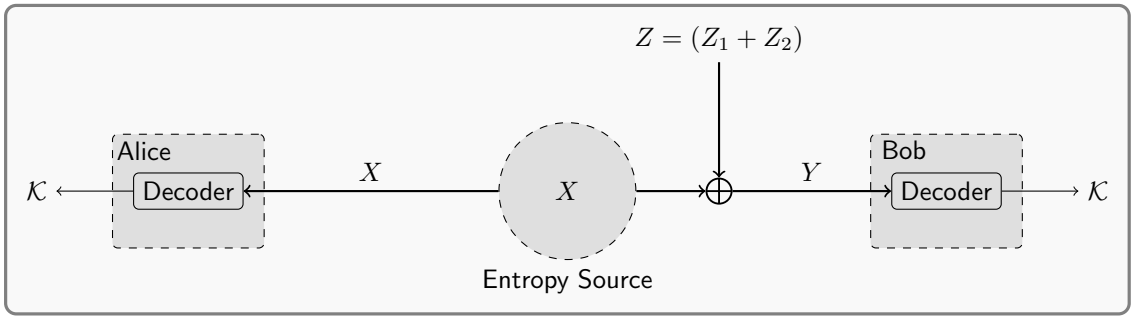
In scenarios where there is an adversary, the model can be adapted to the one shown in figure 2.2a. In this scenario, which is the most general scenario, there is an adversary who is observing realisations of a random variable which is related to the common randomness source X. The random variable observable at the adversary, E, is the sum of the common random variable X and another random variable Z_{E}.

A special case of the adversarial model is when the random variable observed by the adversary can be expressed as the sum of the random variable observable at Bob (Y) and another random variable (Z_{E}) which is independent of Y. E can be expressed as $E = Y + Z_{\mathrm{E}}$. This situation is shown in figure 2.2b and is referred to the degraded adversarial model. In this scenario, the adversarial channel is referred to as being degraded with respect to the main channel because the adversarial channel is equal to the main channel plus some noise. A degraded channel

is guaranteed to be "noisier" than the main channel and so is of primary concern in physical layer key generation as it guarantees that there is some secret key capacity.

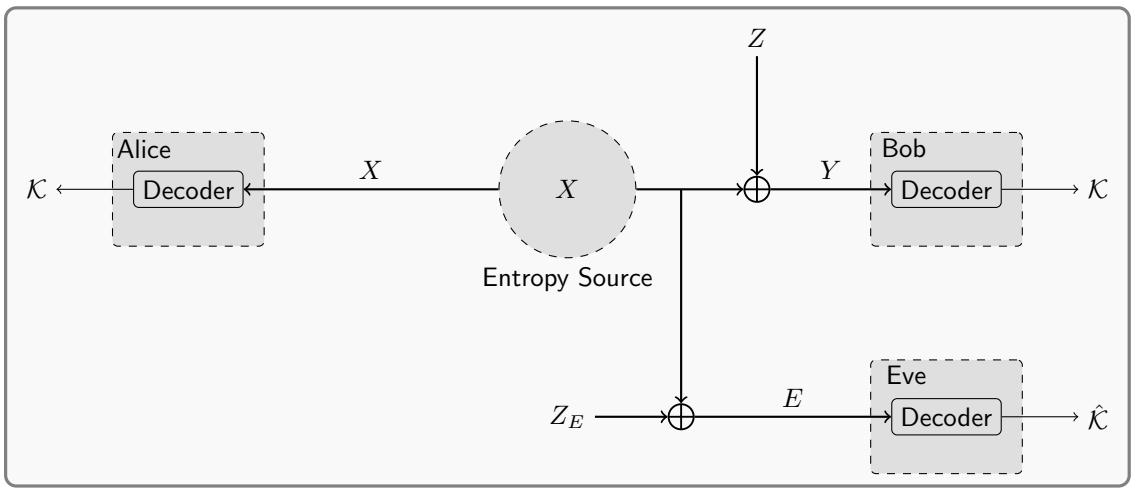


(a) Single Source, Dual Noise Model with no Adversary

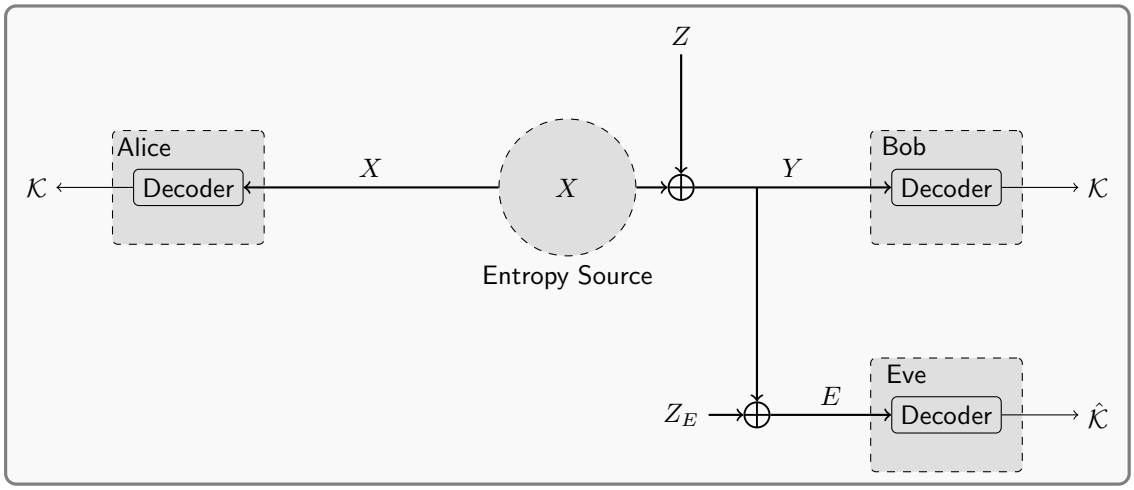


(b) Single Source, Single Noise Model with no Adversary

Figure 2.1: Single Source Models with no Adversary



(a) Single Source, Dual Noise Model with General Adversary



(b) Single Source, Single Noise Model with Degraded Adversary

Figure 2.2: Single Source Models with an Adversary

2.2 Principles of Physical Layer Key Generation

We first start by defining a random variable X which is accessible to n_{A} and another random variable Y which is accessible to n_{B}. Also, let $Y = X + Z$, where Z is another random variable. All random variables are taken as binary sources (i.e they have a support of set $\{0, 1\}$). Let X follow the Bernoulli distribution with the probability of getting a 1 being p (i.e $X \sim \mathrm{Bern}(p)$). If n_{A} samples the random variable n times where $n \to \infty$, then the resulting sequence will have np ones and $n(1-p)$ zeros with almost 100% certainty, this is the law of large numbers. The probability of the resulting sequence having the form described above is thus $(1-\epsilon)$, where ϵ is a very small positive number. The sequences having the form described are referred to as being strongly typical with respect to the distribution X and the

set that contains all of them is called the typical set and is denoted A_ϵ. We will denote all the sequences that are not in the typical set A_ϵ^c. All sequences in the typical set A_ϵ are approximately equally likely, a principle commonly referred to as the Asymmetric Equipartition Principle (AEP) [8] [9] [10] .

A consequence of AEP is that if n_{A} would like to observe an n bit sequence emitted by the source and represent it in the most succinct manner possible, the average number of bits they would need to do that, $\overline{n}_{\mathrm{x}}$, would be

$$\begin{aligned}\overline{n}_{\mathrm{x}} &= \mathbb{P}(x \in A_\epsilon) \times \overline{L}_{\mathrm{A}} + \mathbb{P}(x \notin A_\epsilon) \times \overline{L}_{\mathrm{A^c}} \\ &= (1-\epsilon) \times \overline{L}_{\mathrm{A}} + \epsilon \times \overline{L}_{\mathrm{A^c}} \\ &\approx (1-\epsilon) \times \overline{L}_{\mathrm{A}} + \epsilon_1\end{aligned}$$

Here, $\overline{L}_{\mathrm{i}}$, is the minimum average length of bits one needs to uniquely represent the elements in the set i. The equation above illustrates that the average number of bits we need to represent an n bit realisation of X, $\overline{n}_{\mathrm{x}}$, is determined primarily by the average bits per sequence we use for the sequences in the typical set A_ϵ. For any set with M equiprobable elements, the minimum average number of bits we can use to represent the elements in the set if we wish for all the encoded sequences to later be uniquely decodable, can be shown to be $\log_2 M$. All sequences in the typical set A_ϵ are approximately equiprobable due the AEP and so if the number of elements in A_ϵ is $|A_\epsilon|$, then the minimum average number of bits we need to represent them is $\log_2 M$. Since A_ϵ is the set of n bit numbers with exactly np ones, the number of elements in A_ϵ, $|A_\epsilon|$, is

$$\begin{aligned}|A_\epsilon| &= \binom{n}{np} \\ \implies \overline{n}_{\mathrm{x}} &= \lim_{\mathrm{n}\to\infty} \log_2 \binom{n}{np}\end{aligned}$$

This can be shown with the help of Stirling's approximation (i.e. $n! \approx n^n e^{-n}$) to evaluate to

$$\overline{n}_{\mathrm{x}} = np \log_2 \frac{1}{p} + n(1-p) \log_2 \frac{1}{(1-p)}$$

The ratio of the average number of bits needed to represent the sequence observed from sampling the random variable X n times, $\overline{n}_{\mathrm{x}}$ to the actual number of bits emitted from the source X is a very important quantity and is referred to as the entropy of the source X (i.e $\mathrm{H}(X) = \overline{n}_{\mathrm{x}}/n$). The number of sequences in the typical set is thus $\approx 2^{\overline{n}_{\mathrm{x}}} = 2^{n\mathrm{H}(X)}$.

We have shown that as n tends to be infinity, the number of sequences in the typical set of X, $A_{\epsilon,\mathrm{X}}$, is $2^{n\mathrm{H}(X)}$ and the number of elements in the typical set of Z, $A_{\epsilon,\mathrm{Z}}$, is $2^{n\mathrm{H}(Z)}$ [11]. If a particular sequence y is observed at n_{B}, then we know the $2^{n\mathrm{H}(Z)}$ sequences that could have been observed at X. This is because $x = y + z$ and due to the fact that there are $2^{n\mathrm{H}(Z)}$ different elements in Z.

For any given observed sequence at n_{A}, there are $2^{n\mathrm{H}(Z)}$ possible values that could be observed at n_{B}. For this reason, the observed value at n_{A} can not be used directly as a key as n_{B} has no way of reconciling it without receiving more information from n_{A}. Similarly, the observed value at n_{B} also cannot be used directly as it can't be computed at n_{A}. The legitimate nodes n_{A} and n_{B} thus cannot use observed sequences directly but have to use their observations together with some key extraction mechanism to construct common keys.

For key generation to thus be theoretically possible with a bounded arbitrarily low error rate, n_{A} and n_{B} have to agree a priori to a set of valid keys ($\mathcal{K}$) or a range in which valid keys will fall into. The number of keys in $\mathcal{K}$, $|\mathcal{K}|$, determines the entropy per key, given that every key is equiprobable. The entropy per key will be $C_{\mathrm{k}} = \log |\mathcal{K}|$. The highest possible value of C_{k} is the key capacity per sample and can also be shown to equal the transmission capacity per sample for the case where no eavesdropper is present.

The observation of y has thus given us some information about x if the entropy of Z is less than the entropy of X (i.e $\mathrm{H}(Z) < \mathrm{H}(X)$). This is because before the observation of y, we knew x could take any one of $2^{nH(X)}$ different sequences but after the observation of the sequence y which has been drawn from a random variable Y correlated to X, we know the $2^{nH(Z)}$ possible sequences that could have been sampled from X. After the observation of y, the number of possible sequences observed at x has thus reduced from $2^{n\mathrm{H}(X)}$ to $2^{n\mathrm{H}(Z)}$, which corresponds to a reduction in entropy from $H(X)$ to $H(Z)$. The difference between $\mathrm{H}(X)$ and $\mathrm{H}(Z)$ is called the mutual information and is denoted as $\mathrm{I}(X;Y)$ where $\mathrm{I}(X;Y) = \mathrm{H}(X) - \mathrm{H}(Z) = \mathrm{H}(X) - \mathrm{H}(X|Z)$.

The maximum value of $\mathrm{I}(X;Y)$ is called the capacity of the channel between X and Y. This is also the secrecy capacity in the special case where there is no adversary. It represents the maximum amount of information a realisation of y can give about x. In order for an observer of X and an observer of Y to agree to a common sequence, they have to constrain the common sequence to have an entropy less than $n\mathrm{I}(X;Y)$, where n here is the number of samples each party collects. The proof above is a converse proof that shows that the capacity cannot exceed $\mathrm{I}(X;Y)$. In order to prove that $\mathrm{I}(X;Y)$ is indeed the capacity, we have to prove that $\mathrm{I}(X;Y)$ is achievable if one uses some coding scheme. An achievability proof that shows that we can get arbitrarily close to achieving $\mathrm{I}(X;Y)$ is the Shannon channel coding scheme, a full elaboration of which can be seen in [11] [8].

If we introduce an adversary Eve, n_{E}, who observes E, where $E = X + Z_{\mathrm{E}}$, then the maximum entropy obtained per sample is $I(X;E)$ at n_{E}. The secrecy capacity can be shown to be at least $\mathrm{I}(X;Y) - \mathrm{I}(X;E)$ [8]. The actual capacity in general can be shown to be higher than this in certain cases but if the capacity is exactly $\mathrm{I}(X;Y) - \mathrm{I}(X;E)$, then the adversarial channel is said to be degraded with respect to the main channels [8]. This means that for an adversarial channel with capacity $\mathrm{I}(X;E)$, the case of a degraded adversarial channel is the worst case scenario for secrecy.

The legitimate nodes, n_{A} and n_{B}, need to be able to reconcile a key using their

observations. Let node n_{A} input its observation $\mathrm{x}_{\mathrm{A}}^n$ to use some function f to get the reconciled key (K_{AB}), and let n_{B} input its observed sequence $\mathrm{x}_{\mathrm{B}}^n$ into the same function f to get the same reconciled key.

$$\mathrm{K}_{\mathrm{AB}} = f(\mathrm{x}_{\mathrm{A}}^n) = f(\mathrm{x}_{\mathrm{B}}^n) \tag{2.1}$$

The function f is the key generating function. If there exists a function f that can achieve the objective above then it can be used for key generation. In practice, including the key generation schemes proposed in this thesis, the key generation process is usually into a number of steps and nodes trade information in order to achieve key generation.

In addition to using its own measurements to generate a key, a node uses some information sent from the other legitimate nodes. This information is called the syndrome. So the key generation process can be represented as:

$$K_{\mathrm{AB}} = f_2\left(\mathrm{x}_{\mathrm{A}}^n, f_1(\mathrm{x}_{\mathrm{B}}^n)\right)$$
$$\hat{K}_{\mathrm{AB}} = f_2\left(\mathrm{x}_{\mathrm{B}}^n, f_1(\mathrm{x}_{\mathrm{A}}^n)\right)$$

where f_1 and f_2 represent some functions or processes used in the key generation process. K_{AB} and $\hat{K}_{\mathrm{AB}}$ are computed at n_{A} and n_{B} respectively. Node n_i generates a syndrome ($s = f_1(\mathrm{x}_i^n)$) using it's measurements (x_i^n) and sends the syndrome to other legitimate nodes who use the syndrome to facilitate the key generation process. The scheme is designed in a manner that insures that $K_{\mathrm{AB}} = \hat{K}_{\mathrm{AB}}$ only if the correlation between x_A^n and x_B^n is high.

2.3 The Wireless Communication Channel

When communicating wirelessly between terminals, signals traversing from the transmitter to the receiver get distorted. This is referred to as the transmitted waveform having travelled through a wireless communication channel [12] [13] [14]. In this section, we aim to look at the nature of the wireless communication channel and then show why we are able to model the channel as a stochastic process upon

which cryptographic keys can be generated. This section is also important as it provides the background necessary to understand reasons why physical layer key generation schemes are usually designed to work in specific wireless channel types and why physical layer schemes designed to work under particular channel might be less secure if used in less variable channels.

Signal losses and distortions happen due to three main broad types of dispersions: i) spatial dispersion (caused by factors such as signal scattering), ii) time dispersion (caused by factors such as multi-path propagation) and iii) frequency dispersion (caused by mobility) [14][15][16].

The first main cause of losses is due to the dispersion in space of energy radiated by the transmitter, causing linear path loss. If the signal radiated by the transmitter is broadcast evenly in all directions, then not all that energy will reach the receiver. The farther away the receiver is from the transmitter, the lower the proportion of the transmitted energy will reach the receiver. The path loss is defined as the loss that a transmitter and a receiver separated by a distance d would face in free space, which is not the total loss or gain between the transmitter and the receiver. There are other losses which may cause the total loss at the receiver to be higher or (more commonly) lower at the receiver. The path loss is inversely proportional to the distance between the transmitter and receiver raised to some exponent and given in equation 2.2.

$$
\begin{aligned}
&\text{Power Received } (P_{\mathrm{RX}}) \\
&= (\text{Power Arriving At Receiver}) \times (\text{Aperture of Receiving Antenna}) \\
&= \left(\frac{\text{Power Transmitted} \times \text{Directivity}}{\text{Area of Sphere of radius } d}\right) \times (\text{Aperture of Receiving Antenna}) \\
&= \left(\frac{P_{\mathrm{TX}} G_{\mathrm{TX}}}{4\pi d^2}\right) \times A_{\mathrm{e}} \\
&= \left(\frac{P_{\mathrm{TX}} G_{\mathrm{TX}}}{4\pi d^2}\right) \times \left(\frac{\lambda}{4\pi} G_{\mathrm{RX}}\right) \\
&\implies \frac{P_{\mathrm{RX}}}{P_{\mathrm{TX}}} = G_{\mathrm{TX}} G_{\mathrm{RX}} \left(\frac{\lambda}{4\pi d^2}\right)^2 \\
&\implies P_{\mathrm{RX}}(\mathrm{dB}) = P_{\mathrm{TX}}(\mathrm{dB}) + 10 \log_{10} \left(G_{\mathrm{TX}} G_{\mathrm{RX}} \left(\frac{\lambda}{4\pi d^2}\right)^2\right) \\
&\implies P_{\mathrm{RX}}(\mathrm{dB}) = P_{\mathrm{TX}}(\mathrm{dB}) - L_{\mathrm{PATH}}(d) \qquad (2.2)
\end{aligned}
$$

P_{TX} and P_{RX} are the power transmitted and received respectively, λ is the wavelength of the transmitted wave, d is the distance between the transmitter and the receiver, G_{TX} is the gain of the transmitter, G_{RX} is the gain of the receiver and A_{e} is the effective aperture of the receiving antenna. Not all power that reaches the receiver can be received, the aperture is a measure of how effective an antenna is at receiving radio waves. The equation in 2.2 is called the Friis path loss formula and was first proposed in [17].

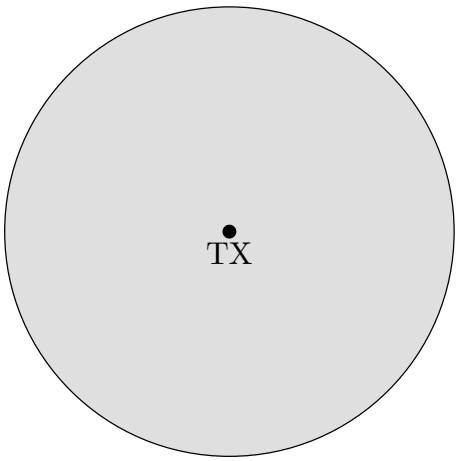
Equation 2.2 shows that the linear path loss is deterministic. It is the only main component of wireless channel loss which can be determined deterministically.

The second broad class of losses and distortions are attributable to time dispersion. When the transmitted waveform travels from the transmitter to the receiver, different copies of the signal propagate along different paths towards the receiver. This multi-path propagation causes these copies to arrive at the receiver at different times and with signals from each path having suffered different distortions, causing slow and frequency-selective fading. Slow fading is caused by distortions due to blockage from objects in the signal path. The arrival of different copies of the signal at different times is referred to as time dispersion as the time to receive a

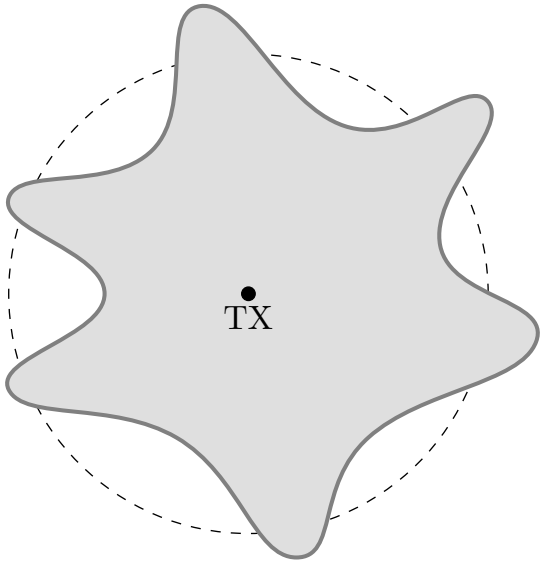
transmitted symbol differs from the time it takes to transmit it [18] .

A particular metric used to characterise the severity of slow fading is the delay spread (σ_{d}). The delay spread is a metric which characterises the time difference between the time of arrival of the first copy of the signal and the time of arrival of the last significant copy. A large delay spread means that a transmitted symbol will interfere with itself (intra-symbol interference) and also with subsequent symbols (inter-symbol interference). In particular, Inter-Symbol Interference (ISI) is particularly severe if the delay spread is greater than the symbol duration as it means that multi-path components from the preceding symbol will still be getting received whilst the current symbol is getting received. A channel with a delay spread greater than the duration of each symbol is referred to as being frequency-selective as the time-dispersion experienced causes different frequency components that make up the transmitted signal to experience different losses.

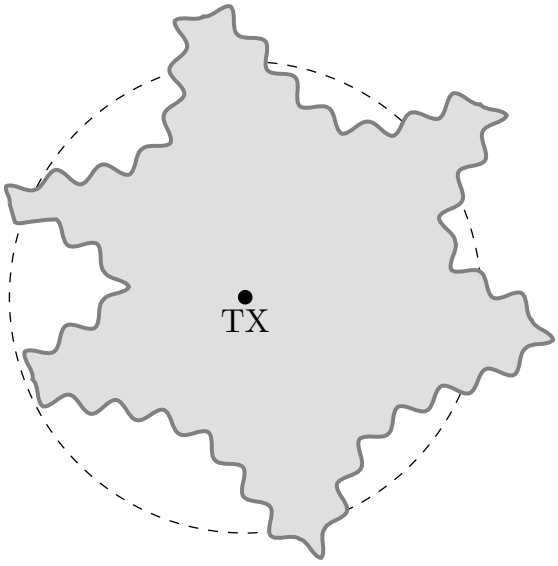
The receiver sensitivity level is the minimum received power level that a receiver will be able to identify and successfully demodulate. Figures 2.3a - 2.3c illustrate the main types of losses experienced over wireless channels. In the figures, a transmitter (TX) is transmitting a signal wirelessly. The shaded areas show the regions we would expect the transmitted signal to have an average power level exceeding the receiver's sensitivity level. From the figures, it is clear that with just path loss, the transmission range is the same in all directions whilst with slow and/or fading losses added, the transmission range has some spacial variation. We can also see that fast fading related losses tend to vary more rapidly than slow fading losses but the fast fading losses oscillate within a much smaller range.



(a) Free space path loss



(b) Free space path loss and slow fading



(c) Free space path loss, slow fading and fast fading

Figure 2.3: Different types of losses incurred over wireless channels. The grey area is the transmission range.

Frequency-selective fading and slow fading are present even in static wireless communication scenarios (i.e when the transmitter and receiver are static) and are thus the key mechanisms exploited for key generation in situations where nodes are not mobile.

Equation 2.3 shows the loss due to frequency-selective, slow fading [19]. As the equation shows, this loss is not deterministic like the path loss but a random variable. The distribution that the random variable takes is gaussian lognormal distribution. Due to the asymptotic equipartition property of random variables, we know that the observation of n samples (where n is large) of this variable will thus yield one of $2^{n\mathrm{H}(Z)}$ equiprobable sequences from the typical set of Z, $A_{\epsilon,\mathrm{Z}}$. In equation 2.3, σ denotes the standard deviation of the random variable.

$$p(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} \tag{2.3}$$

The third broad class of dispersion of concern is frequency dispersion induced losses that refer to losses induced by the mobility of nodes, causing the channel to vary with time. A node which is mobile cannot ascertain what frequency it is receiving accurately if its velocity relative to the transmitter is constantly changing. This is primarily due to the doppler effect. A receiver receiving a signal with a frequency f will perceive the frequency as actually being $f + \Delta f(t)$, where

$$\begin{aligned} \Delta f(t) &= \frac{\Delta v(t)}{c} f \\ &= \frac{||v(t)||}{c} \cos\theta(t) \\ &= f_{\mathrm{m}}(t) \cos\theta(t) \end{aligned} \tag{2.4}$$

Here, $\Delta v(t)$ is the relative velocity between the receiver and the transmitter, and c is the speed of light.

Frequency dispersion causes the perceived frequency received to depend on time which in turn causes the channel to vary with time (i.e. become time selective). The coherence time, T_{c}, is a key metric used to characterise the extent to which a

channel suffers from frequency dispersive losses. Let X be the correlation between the channel response at time t and channel response at time $t + T_{\mathrm{c,X}}$. The X% coherence time is defined as being equal to $T_{\mathrm{c,X}}$. This type of fading is commonly referred to as fast fading and can be shown to follow a Rician distribution in the case where there is a line-of-sight component or a Rayleigh distribution in the case of no line-of-sight component. Equation 2.5 shows the nature of the distribution of the power loss due to frequency dispersion in the case of there being line-of-sight and equation 2.6 shows the distribution when there is no line-of-sight component.

$$p_{\mathrm{Rician}}(y) = \frac{1}{2\sigma^2} e^{-(y^2+s^2)/2\sigma^2} I_0\left(\frac{s\sqrt{y}}{2\sigma^2}\right) \tag{2.5}$$

$$p_{\mathrm{Rayleigh}}(y) = \frac{\sqrt{y}}{\sigma^2} e^{-\frac{y}{2\sigma^2}} \tag{2.6}$$

Where $I_0(x)$ is the 0^{th} Bessel function, s and σ are distribution shaping parameters, a full characterisation of which is available in [20]. If the nodes are not mobile, the fluctuation due to fast fading is very small and the power profile of the received power can be approximated as just following a log-normal distribution.

2.4 Summary

This chapter has provided a brief overview of the theoretical foundations of physical Layer key generation over wireless communication channels. In particular, the chapter showed how different distortions affect the distribution of the signal that is wirelessly traversing from some transmitter to some receiver. The chapter also looked at different models for the adversarial channel in different physical layer key generation models. The model which is of specific importance is the degraded model which helps us estimate the lower bound of the secrecy capacity of a channel.

Chapter 3

Literature Review

3.1 Review of Group Key Management Schemes in Resource Constrained Networks

There has been a lot of research done into efficient techniques for pairwise and group key management in resource constrained networks over the years. These techniques have mainly focused on applications within the WSN space where key management is particularly difficult because of the dynamic nature of WSN topologies [21] [22] [23] .

Nodes in a WSN are typically deployed with some secret key information loaded onto them and they use that information to establish keys. They then use a key management protocol to establish, refresh and revoke keys when required. In many deployment scenarios, networks need to enforce forward security in WSNs and they do this by the refreshment of symmetric keys through either regular key updating from the base station or some form of rehashing over the current key to get a new key such as in [24].

There are two broad classes of key establishment schemes - i) public-key schemes and ii) symmetric key schemes [25][26]. In public-key schemes nodes establish cryptographic keys with the help of key pairs - with one key being private and the other key being public. The private key of a node is a secret key which is unique and

known only to that node and a public key is a key which is unique to a node but is known by all parties, including adversaries.

The general key establishment process in a public-key scheme involves two nodes trading public keys and then computing a common cryptographic key between themselves. The schemes rely on using a mathematical function that outputs a common key between two nodes when provided with a private key from one node and a public key from another. After the establishment of the pairwise key, the nodes can either use some symmetric key based protocol to establish keys or they can use a public-key group key establishment scheme, such as the Diffie-Hellman, algorithm to generate any required keys [25] [27] [28] [29] .

In symmetric key establishment schemes, keys are deployed beforehand or are established after deployment with the help of some Trusted Third Party (TTP). Keys that are going to be used for secure communication need to be preloaded into individual nodes in some way, which can be done in a centralised or distributed way. In a centralised scheme, nodes wishing to communicate with each other can do so with pairwise encryption keys obtained from some trusted, centralised node using a key distribution protocol like the popular Kerberos protocol [30].

In [30], all nodes share a key with a trusted node called a Key Distribution Centre (KDC) but not with other nodes. When a node A (n_{A}) wants to communicate with another node B (n_{B}) it first sends a request to the KDC and the KDC generates a session key and sends two copies of the key to node A, one encrypted with n_{A}'s key and one encrypted with n_{B}'s key. Node A then retrieves the session key and sends node B the session key encrypted with the key belonging to n_{B}, so both nodes now share a key and can start communicating. In centralised approaches, key refreshment is handled centrally. Every time the base station needs to refresh keys, it has to send them directly to the nodes, which incurs communication overhead. In the case of key distribution via a KDC, refreshment is initiated by the nodes themselves through session key requests.

If a group key is broadcasted over a secure channel, sensor nodes will need to authenticate that group key before accepting it in order to thwart replay attacks. In a typical scheme such as the μTesla scheme [31] the base station assures authentication by using one-way hash chains. The i^{th} group key is the hash of the $(i+1)^{th}$ group key, i.e. $K_{\mathrm{G,i}} = H(K_{\mathrm{G,i+1}})$. This is achieved by choosing a random value of $K_{\mathrm{G,N}}$ and using it to precompute all group keys $\{K_{\mathrm{G,N-1}}, K_{\mathrm{G,N-2}}, ..., K_{\mathrm{G,0}}\}$ at the base station i.e. $K_{\mathrm{G,N-1}} = H(K_{\mathrm{G,N}}), K_{\mathrm{G,N-2}} = H(K_{\mathrm{G,N-1}})$, etc. After the initial group key, $K_{\mathrm{G,0}}$, is computed and broadcast to the nodes, subsequent group keys can be authenticated by hashing them and checking if the hash value equals the previous key. This provides easy authentication but it does not provide the forward secrecy of group keys because if an adversary obtains a particular session group key then the adversary can get all previous keys by just hashing the current session key.

In the distributed approach, key material is pre-loaded onto individual nodes prior to deployment and the nodes use that information to generate pairwise keys. Schemes that utilise this approach include a basic scheme where all nodes come preloaded with pairwise keys of every other node. This solution is not scalable as memory requirements rise quickly with the increasing network size, and the lack of a centralised node or protocol handling refreshment also means that forward secrecy is not assured. Because of the scalability, memory and security issues of preloading all pairwise keys into nodes, other random distributed schemes such as those detailed in [32, 33, 34, 35] are usually used instead.

In a typical scheme such as [32] there is a set of network keys and each node is preloaded with a random subset of these network keys prior to deployment. Two nodes wishing to communicate with each other then go through a key agreement phase and pick a common key from their key sets and use that as their session key. In the scheme detailed in [34], the key material is a subsection of a special symmetric secret matrix. Node n_{i} gets loaded with row i (r_{i}) and column i (c_{i}) of the secret matrix and similarly node n_{j} gets r_{j} and c_{j}. When n_{i} and n_{j} want to communicate with each other, they exchange columns and compute the session key as $K_{\mathrm{ses}} = r_{\mathrm{i}}.c_{\mathrm{j}} = r_{\mathrm{j}}.c_{\mathrm{i}}$. There are many other similar schemes such as [35] where the key material is a fraction of a polynomial. Forward secrecy via key refreshment in

these distributed schemes is achieved by either i) regularly updating the key material in nodes or ii) performing hashes over the current keys to form new session keys.

The actual implementation of the hashing can be done in several ways depending on the needs of the WSN. This can be, for example, done by forming hash chains by encrypting the communication between two nodes with a key K_{v} where $K_{\mathrm{v}} = H^{v}(K_0) = H(K_{\mathrm{v}-1})$, H^{v} is a hash chain of order v (i.e performing v hashes), K_0 is the initial key and v is the version number [24]. Alternatively refreshment can also be done by encrypting the communication with key $K_{\mathrm{v}} = H(K_{\mathrm{v}-1}||v)$ [24] or encrypting the communication with $K_{\mathrm{v}} = H(K_{\mathrm{v}-1}||r)$, where $'||'$ is the concatenation operator and r is a random number chosen from a small set of numbers [36]. In the first two schemes there will be additional communication overhead because nodes will have to send the key version v used in encryption along with the payload and in the later scheme there is additional computational overhead because the receiving node will only have $K_{\mathrm{v}-1}$ and will thus have to compute K_{v} by iterating and trying out all numbers in the set from which number r was chosen. Key refreshment using physical layer keys could allow us to refresh keys in a similar manner to the latter scheme but with a random number r that is not restricted to an element in a small set of numbers but instead has been agreed upon by both nodes prior to refreshment. The generation of practical physical layer keys is explored in [37, 38, 39].

3.2 Requirements and Review of Physical Layer Key Generation

A PLSKG scheme needs to fulfil certain requirements for it to be applicable to WSNs. These requirements include both computational and energy efficiency in addition to the requirements that all PLSKG schemes need to fulfil such as secureness with respect to adversarial nodes. The requirements that needs to be fulfilled by a PLSKG scheme for WSNs include:

- Energy efficiency: The scheme needs to be as energy efficient as possible so as to allow it to run on resource constrained nodes such as WSN nodes, which

tend to run on battery power.

- Computational efficiency: The algorithms that make up the scheme need to require low computational resources so as to allow them to run on microprocessors which often power resource constrained wireless devices.

- Low cost: The scheme should ideally require little or no additional specialist hardware to be added on standard off-the-shelf wireless devices for it to function. This helps keep the cost low. A lot of additional hardware can add additional costs which can negate the benefits of using a resource constrained device.

- Secureness: The scheme should be secure relative to an adversary who is not collocated with respect to any of the collocated nodes. This requirement requires the scheme to generate keys with an entropy that is as large as possible and also computable. In most PLSKG schemes, the entropy of the final generated key can be hard to compute even if the eavesdropper's statistics are known, which hamper's their applicability.

- Flexible: The PLSKG scheme needs to ideally be suitable for stationary nodes and not just mobile nodes. Mobile wireless channels tend to be more variable than wireless channels between stationary nodes and thus easier to generate keys over but nodes in real deployments are often not always in continuous motion.

In a group setting, nodes can be deployed in a wide variety of topologies and so in addition to the requirements outlined above, a group PLSKG scheme also needs to ideally be applicable to nodes in a wide variety of deployments. The nature of PLSKG makes this requirement very difficult to fulfil.

The pairwise and group PLSKG schemes proposed in this thesis are suitable for use in scenarios where the nodes will be in the same locality when key generation takes place and the adversary will not be collocated with respect to the legitimate nodes. Experiments were done for a maximum separation of 3m for each of the nodes in a group setting and a separation of 10m in a pairwise setting. The proposed schemes are therefore suitable for a variety of applications such as indoor IoT home applications. In these applications, the legitimate nodes are usually close to each other and in an indoor setting whilst an adversary is usually not near the legitimate

nodes at the time at which the cryptographic key is first established. The proposed scheme is not suitable for situations where the adversary can be collocated relative to the legitimate nodes or scenarios where the adversary lies directly between the legitimate nodes. In scenarios where nodes are dispersed over a very wide indoor area, such as in a large factory, nodes would need to be divided into clusters (where each cluster of nodes is in a set area) and then physical layer keys generated for each cluster. The cluster keys can then be used to formulate a common group key using conventional tree based techniques.

Keys are generated over the physical layer mainly using two different approaches, namely i) via channel estimation or ii) via quantisation of the receive signal strength indication (RSSI) [38, 39]. The first approach tries to estimate the channel impulse response (h) of the channel at both nodes and then uses h as a basis of the generation of a random bit sequence. This can be done by extracting a symbol s from the estimate h and then mapping s to a random bit sequence such as in [40]. The physical channel changes after every coherence time interval and so a symbol s can be estimated after every coherence time interval. The second approach quantises the RSS value and then either i) uses particular portions of that quantised value as a common source of randomness between two nodes such as in [41] or ii) obtains a series of RSSI measurements over several transmissions and combines them in some way to extract random bits such as in [37].

In physical layer-based schemes, secret key information does not necessarily need to be furnished beforehand. Nodes can either be deployed with no secret key information or very little secret key information. Nodes can then use measurements of some physical layer parameters to generate physical layer keys. If an adversarial node is not collocated with respect to the legitimate nodes at the point at which key generation happens, then the legitimate pair or group of nodes can generate a common key [8].

Practical key generation involves the design, implementation and analysis of techniques that generate cryptographic keys using practical transceivers. They aim to realise theoretical key constructions and also identify aspects of physical layer security which may be accurate in theory but may be prohibitively difficult to implement in practice and to identify particular properties of the physical layer hardware

upon which allow the implementation of physical layer schemes, can contribute to or compromise security.

The first work that quantitively studied the secrecy capacity achievable over some public channel was Shannon in his work in [42]. The work considered the case when two legitimate parties communicated over a noiseless public channel. The work showed that in this scenario the entropy of the secret key known only to legitimate parties needed to be at least equal to the entropy of the message in order to achieve perfect secrecy. A cryptographic setup which achieves this is Vernam's cipher, commonly referred to as a one-time pad.

If on the other hand the entropy of the key is less than the entropy of the message (i.e. $H(\mathsf{K}) < H(\mathsf{M})$), then only $H(\mathsf{K})$ bits of security are guaranteed. In practical cryptographic schemes, this is satisfactory as long as $H(\mathsf{K})$ is large enough to make it impractical for an adversary to make a brute force search for it.

One of the first works to look at the impact of secrecy when legitimate parties communicate over stochastic channels is Wyner in [43]. Wyner showed that in the event that the adversarial channel was noisier than the legitimate channel, it was theoretically possible for the transmission of a secret message over the legitimate channel with an arbitrarily small probability of error, to take place if a sufficiently large codeword length, n, is used. Channels in which the above condition holds are channels which are said to have secrecy capacity. The model Wyner considered is called the wiretap channel model and is illustrated in 3.1. A wiretap channel model (shown in figure 3.1) models the adversarial model as a channel which is degraded with respect to the legitimate channel. The model models a situation where legitimate nodes transmit information over a noise free channel and an adversary observes information over a noisy channel. Codes which aim to transmit messages securely over a wireless wiretap channel are called wiretap codes.

Instead of attempting to transmit messages directly through wiretap codes, we can aim to utilise the presence of secrecy capacity in the channel to generate a crypto-

graphic key. The cryptographic key can then be used to facilitate secure communication using a symmetric encryption scheme such as the Advanced Encryption Standard (AES). A popular practical physical layer key generation framework is the one outlined in [8]. The framework does not provide a specific key generation scheme but it provides a general set of stages (or layers) that a physical layer key generation scheme can decompose the generation process into. The stages are i) the randomness sharing stage, ii) the key reconciliation stage and iii) the privacy amplification stage.

The randomness sharing stage is where nodes measure the value of some common physical layer quantity. The key reconciliation stage is where nodes attempt to use their respective measurements to formulate a common secret key. The privacy amplification stage is where nodes convert all forms of secret information they hold into one common (or shared) key.

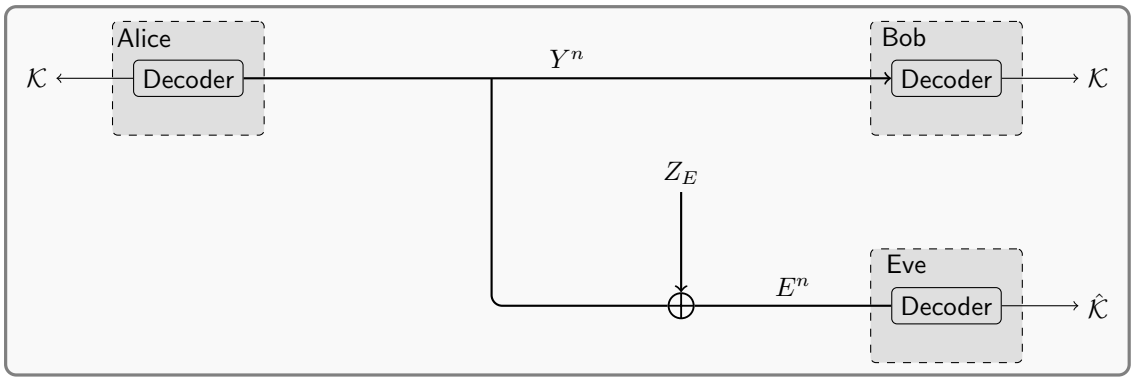


Figure 3.1: Wyner's Wiretap Channel

3.3 Review of Pairwise Physical Layer Key Generation Schemes

Pairwise PLSKG schemes have been proposed over the years, the majority of which target general communication networks although some have been proposed for resource constrained networks as well, particularly for wireless sensor networks where key management can pose numerous challenges.

The paper [37] proposes an antenna reactance based practical PLSKG scheme. It achieves key generation by using a set of special antennas fitted to the ends of the

legitimate nodes called Espar antennas. The idea is that by varying the directional amplitude and phase characteristics of the radio waves radiating from the Espar antennas at both ends of the legitimate communication channel, the received power at both ends can be varied in such a way that the correlation between the legitimate channels can be preserved whilst the adversarial channel is being made to be noisier with respect to the legitimate channels.

The scheme not only exploits the variation in RSS in the main channel but also introduces more variation in the channel by randomly varying the effective receive and transmit power. This method requires special antennas to be used on the legitimate nodes, a condition which is hard to meet in resource constrained networks such as WSNs where device cost is a major factor of concern.

The main constraint with the use of channel impulse response as the basis for physical layer key generation is that the channel response will not be easily available to higher layers of resource constrained devices. This forces practical key generation schemes for resource constrained devices to resort to using coarse measures of channel randomness such as RSS.

The work outlined in [41] considers secure PLSKG using receive power between mobile legitimate nodes. It focuses on exploiting the mobile wireless channel (i.e. when the relative velocity between the legitimate nodes is high) to generate keys. If nodes are mobile, there is additional variability in the channel owing to fast fading losses due to frequency dispersion caused by doppler shifts. This variability provides an additional source of entropy in the main channel relative to a stationary adversary which is exploited by the scheme to generate keys.

The schemes outlined so far all require some special channel conditions (such as mobility) or some special equipment (such as hardware) to operate. In the cases where devices are not mobile and/or we are concerned with applicability in resource constrained networks, we need to look at generating the cryptographic key from standard indoor and/or outdoor wireless channels using a general transceiver.

Research to this end for resource constrained networks has been conducted over the years such as the works outlined in [1].

In [1] a physical layer key generation scheme is proposed for resource constrained networks using RSS measurements. The scheme is suitable in static networks (i.e. networks where the relative velocity between nodes is zero or near zero) and uses different frequencies to generate cryptographic keys. In the case of static nodes, it is common for nodes to generate keys by exploiting the variability in the different frequency channels.

The physical layer key generation randomness sharing procedure involves the nodes 'hopping' between different frequency transmission bands and using measurements from all the channels to generate keys. The reason for different channels having different channel conditions are numerous, including the fact that variations in levels of channel use cause noise floors in different channels to differ. The work in [1] uses this difference in channel conditions to generate keys.

The key shortcoming of current key generation schemes lies in the fact that they often require special hardware and/or special channel conditions (most commonly mobility) in order to operate. The key reconciliation processes of these schemes also do not allow an implementer to properly ascertain the resulting entropy of the resulting key even if the correlation between the legitimate nodes' measurements is known. This makes it difficult to establish what level of error correction is required in the key reconciliation stage.

It is clear that if an ECC which corrects too many errors is used, then the resultant key will no longer have any entropy. Current state-of-the-art schemes do not allow one to establish exactly what this error correction threshold is even if all relevant correlations are known. An implementer can thus not compute the threshold at which the entropy falls to zero. The proposed scheme in this thesis aims to address this issue, and its key reconciliation process is applicable to both pairwise and group key generation.

The review of existing pairwise PLSKG schemes above has highlighted a number of weaknesses that exist in current state-of-the-art PLSKG schemes, which all have one or more weaknesses such as:

- their use of special hardware in addition to the hardware that would typically be found in a wireless device such as in [37]. This adds additional costs and greatly reduces the applicability of the scheme.
- their limited support of non-mobile wireless channels. A lot of schemes rely on the variability of mobile channels and therefore require relative motion between legitimate nodes for key generation to take place.
- their use of purely quantisation based methods in the key reconciliation stage. This reduces the final entropy of the key in an unpredictable way, making it impossible to ascertain what the final entropy of the generated key will be.

3.4 Review of Existing Group Physical Layer Key Generation Schemes

In the group key generation scenario, the goal is to generate keys for use for the purpose of secure group communications between three or more nodes. This, as in pairwise physical layer key generation, is done by making measurements over the legitimate channels. In this thesis we will be concerned with group key generation in scenarios where the adversarial channel is degraded with respect to all legitimate channels.

Group key generation schemes have also been investigated in a number of research works in recent times. The majority of the work has been on the theoretical constructions of group key schemes or practical constructions implemented on a computer simulation platform.

The work in [44] presents a theoretical framework for the generation of group keys using the channel impulse responses of the legitimate wireless channels. The work covered various topologies and was able to show that the group key generation capacity can be achieved given some prerequisites. Another theoretical framework

for the generation of physical layer keys is presented in [45][46]. The work proposes a tree based key formulation framework to generate group keys from pairwise keys.

The work covering the practical implementation of a proposed group key scheme was first presented in [47][48]. The scheme proposes a secure group key scheme for mobile wireless devices that uses the trend (i.e gradient) of measured RSS values to generate physical layer keys. A relay-based technique for enabling nodes to use the keys they have generated to establish a common group key is then presented. The scheme considered star and chain (line) topologies.

The schemes in [47][48] are purely quantisation based and therefore suffer from the limitation that some errors cannot be corrected at the end of the quantisation stage. The schemes are also relay based and is aimed at mobile nodes, with the key generation taking place by nodes alternately sending probes to each other and using those probes to try and estimate the losses across one pre-chosen channel.

The key generation of the schemes is done by using two different methods: i) an RSS fading trend and median threshold (RTM), and ii) RSS fading trend and quantisation (RTQ). The fading trend is a sampled, moving average value of the RSS. It is used in lieu of direct RSS measurements in the schemes because the schemes were targeted at mobile networks. In such networks, the relative velocity between the nodes causes frequency dispersion in the channel, which then causes time-selectivity (i.e. causes the channel to change quickly in time).

The RTM method involves the nodes using n measured RSS values as a n binary sequence, where each binary digit at position i indicates whether the i^{th} RSS value received was below or above the median of all the RSS values received. The RTQ is a thresholding procedure which involves thresholding using multiple thresholds. The formulation of these thresholds is done by using the probability distribution of the channel between nodes, so the distribution has to be known a-priori or estimated. The thresholds are then set in such a manner that the probability of a RSS value falling between any two adjacent thresholds is constant.

State-of-the-art GPLSKG schemes have provided a good basis for the investigation of group key generation but current shortcomings include the fact that most either i) do not provide a practical manner in which the theoretical frameworks proposed can be translated to a practical GPLSKG scheme or ii) are only suitable for mobile

nodes.

The review of existing group PLSKG schemes above has highlighted a number of weaknesses that exist in current state-of-the-art group PLSKG schemes, which all have one or more weaknesses such as:

- their use of relay-based techniques to generate a group key leads to an inefficient use of probes in some deployment topologies.
- their limited support of non-mobile wireless channels. A lot of schemes rely on the variability of mobile channels and therefore require relative motion between legitimate nodes for key generation to take place.
- their use of purely quantisation based methods in the key reconciliation stage. This reduces the final entropy of the key in an unpredictable way, making it impossible to ascertain what the final entropy of the generated key will be.

3.5 Summary

In summary, there has been a lot of advancements in both pairwise and group PLSKG schemes over the past decade but there still remains a lot to be done. Current state-of-the-art schemes provide a basis for the generation of physical layer keys over wireless channels but often have one or more weaknesses that hamper their wide spread use.

The main findings of this literature review show that the manner in which PLSKG schemes are designed often lead to PLSKG schemes being very difficult to implement in practice due to requirements such as i) requiring nodes to be in continuous motion during the key generation process (mobility requirement), ii) requiring nodes to have additional hardware installed to facilitate the key generation process (which often leads to increased costs which can negate the use of resource constrained devices in the first place) or iii) requiring devices to use the channel impulse response in the key generation process, a parameter which is hard to estimate and use in low-power devices. In addition to these, it is often not possible to ascertain what entropy the final generated key will be even if characteristics of the legitimate and adversarial channels are known. This is due to the manner in which the key recon-

ciliation stage of the key generation process is often carried out.

In light of these weaknesses, this thesis aims to address most of these shortcomings in the environments where resource constrained nodes that are non-mobile. The thesis will propose novel pairwise and group PLSKG schemes that are suitable for use in such environments and allow the entropy of the resultant keys to be estimated in the case where the long-term correlation between all channels is known.

Chapter 4

Pairwise Physical Layer Key Generation

This chapter first details a general framework within which physical layer keys are generated and then proceeds to detail a proposed physical layer secure key generation scheme for resource constrained networks. The proposed scheme was implemented on WSN nodes and the resulting keys tested for cryptographic hardness.

4.1 Physical Layer Key Generation Framework

As detailed previously, physical layer keys are generated by observing a random variable X_{A} which is jointly distributed (or correlated) with another random variable X_{B}. The resulting key bits that would then be extractable would be $I(X_{\mathrm{A}}; X_{\mathrm{B}})$ per sample in the event of there being no eavesdropper and $I(X_{\mathrm{A}}; X_{\mathrm{B}}) - I(X_{\mathrm{A}}; X_{\mathrm{E}})$ bits per sample in the case that an eavesdropper's observations follow a distribution, X_{E}, which is physically degraded with respect to the legitimate channel. The secrecy capacity per sample with no adversary (C_{s}) and the secrecy capacity with the adversary with a degraded channel present ($C^{d}_{\mathrm{s,E}}$) are given in equations 4.1 and 4.2 respectively.

$$C_{\mathrm{s}} = \max I(X_{\mathrm{A}}; X_{\mathrm{B}}) \tag{4.1}$$

$$C^{d}_{\mathrm{s,E}} = \max I(X_{\mathrm{A}}; X_{\mathrm{B}}) - \max I(X_{\mathrm{A}}; X_{\mathrm{E}}) \tag{4.2}$$

We now firstly look to how we quantify the value $I(X_{\mathrm{A}}; X_{\mathrm{B}})$ from observations of X_{A} and X_{B}, x_{A}^n and x_{B}^n.

Theorem 1. *If $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$ are random Gaussian variables and both independent of Z with $Y = X + Z$, then the mutual information between X and Y is $-\frac{1}{2}\log_2(1-\rho^2)$, where ρ is the person correlation coefficient between X and Y [11].*

Proof. The proof utilises the fact that if the joint distribution of X and Y is bivariate normal, then the relation between their mutual information and their correlation can be easily derived.

Let U and V be any two independent normally distributed Gaussian random variables. By definition, any two random variables,X and Y, that can be expressed in the form

$$X = c_1 U + c_2 V$$
$$Y = c_3 U + c_4 V$$

have a joint distribution that is bivariate normal. Where c_1, c_2, c_3 and c_4 are constants. By taking $U = X$ and $V = Z$, it can be seen that X and Y can be expressed in the form below:

$$X = (1)U + (0)V$$
$$Y = (1)U + (1)V$$

X and Y are two mutually dependent gaussian random variables, and the joint distribution (X, Y) will be a bivariate normal distribution U where U is

$$U \sim (\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where

$$\boldsymbol{\Sigma} = \begin{bmatrix} \rho & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \rho \end{bmatrix} \qquad \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \tag{4.3}$$

From the definition of mutual information and using the fact that for any Gaussian distribution with mean μ and variance σ^2 the entropy is $\frac{1}{2}\log_2 2\pi e\sigma^2$, we can evaluate the mutual information between X and Y as

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - H(X,Y) \\ &= \frac{1}{2}\log_2 2\pi e\sigma_1^2 + \frac{1}{2}\log_2 2\pi e\sigma_1^2 - \frac{1}{2}\log_2 2\pi e|\mathbf{\Sigma}| \\ &= \frac{1}{2}\log_2 2\pi e\sigma_1^2 + \frac{1}{2}\log_2 2\pi e\sigma_1^2 - \frac{1}{2}\log_2(2\pi e(\sigma_1^2\sigma_2^2 - \rho^2\sigma_1^2\sigma_2^2)) \\ &= -\frac{1}{2}\log_2(1-\rho^2) \end{aligned} \tag{4.4}$$

□

Theorem 1 enables one to use observations over a channel to estimate the mutual information, a property which will be used later, and then proceed to estimate the secrecy capacity per key as outlined in equation 4.2. Theoretically, if one wants to generate a cryptographic key of length N_k, then the legitimate parties will need to sample the channel for a minimum of $N_k/C_{s,E}$ times to generate the key.

It is important to note that the vast majority of key results are statistical in nature and rely on the assumption $n \to \infty$ to characterise the secrecy and key generation capacity. Practical physical layer schemes will only rely on a limited number of channel observations to generate a key and so the actual achieved probability of successful key agreement (the key agreement rate) is often much lower than 1. The key agreement rate can be maximised by designing the key generation procedure carefully and adding error correction stages which leak as little information as possible to a localised adversary.

Practical physical layer key generation schemes, including the one proposed in this thesis, usually break the key generation procedure into three generic stages: i) the randomness sharing stage, ii) the key reconciliation stage and iii) the privacy amplification stage.

The randomness sharing stage involves all legitimate nodes that wish to acquire a group key sampling the channel parameter of interest multiple times and storing the observations. The channel parameter of interest is most usually either the channel state information (CSI) or RSS for resource constrained nodes. Variability in the channel is often induced by mobility, some form of frequency hopping or a variable

response antenna. The measurements are then quantised and moved on to be used by the next stage, which is the key reconciliation stage.

The key reconciliation stage is the stage in which errors in measurements are corrected. The aim of the stage is to enable two legitimate nodes that have access to a series of observations that have been sampled from correlated random variables to reconcile a key. The stage involves correcting errors between measurements or reconciling keys in such a way that the process is resistant to a certain threshold of errors in the measurement vectors.

The privacy amplification, serves the purpose of taking a sequence and transforming it to another sequence that has an entropy that corresponds to its length. The stage usually involves some form of cryptographic hashing together with a challenge-response exchange of messages between legitimate nodes in order to enable them to ascertain if they managed to generate a common key which they can then use for secure communication. Table 4.1 shows the key stages involved in PLSKG.

Stage	**Function**
Randomness Sharing	Enables legitimate nodes to accumulate observations of the current state of the channel.
Key Reconciliation	Enables legitimate nodes to reconcile key by correcting or reconciling differences in their channel observations.
Privacy Amplification	Enables nodes to extract a cryptographic key from the longer reconciled key.

Table 4.1: Key Stages of PLSKG

4.2 System Model

4.2.1 Wireless Channel Model

PLSKG schemes aim to generate cryptographic keys by observing the physical channel between two nodes and using the channel as a common source of randomness upon which to generate keys. The wireless channel is modelled as being comprised of two main components: i) a multiplicative fading loss h (which we will in this text

just refer to as the 'channel') and ii) an additive noise component n. When a node n_{alice} sends a symbol x to another node n_{bob}, n_{bob} receives a noisy signal y_{ab} where $y_{\text{ab}} = h_{\text{ab}} * x + n_{\text{ab}}$ with $*$ being the convolution operation. Conversely, when n_{bob} sends x to n_{alice}, n_{alice} receives $y_{\text{ba}} = h_{\text{ba}} * x + n_{\text{ba}}$ [49].

The additive noise components n_{ab} and n_{ba} are independent random variables, so they cannot be used as a common source of randomness [50] [51]. For a static channel, the fading components h_{ab} and h_{ba} stay highly correlated over a coherence bandwidth (B_{c}) because of the channel reciprocity principle. If the frequency used by n_{alice} and n_{bob} is within that band, then $h_{\text{ab}} \approx h_{\text{ba}}$.

In the case of a non-static mobile channel, channel fading will stay highly correlated for time intervals shorter than the coherence time t_{c} provided that the frequency remains within the band B_{c} (where t_{c} is directly proportional to the relative speed at which nodes move). A third node n_{eve}, which is a distance d away from n_{bob} and receives a noisy signal from n_{alice}, will receive $y_{\text{ae}} = h_{\text{ae}} * x + n_{\text{ae}}$, where h_{ae} decorrelates with respect to h_{ab} as d increases. At a distance $d \geq \lambda/2$ (where λ is the wavelength of the transmitted signal), it can be shown that the fading components seen by n_{alice} and n_{bob} will be largely independent of h_{ae}. Figure 1.1 shows the relationship between channels between n_{alice}, n_{bob} and n_{eve}. The fact that mobile channels experience an additional type of fading that non-mobile channels don't experience - fast fading - make the key generation capacity in mobile channels exceed that found in non-mobile channels.

4.2.2 Adversarial Model

It is important to outline the assumptions made about the adversary's computational and physical advantages just as in conventional cryptographic protocols. The adversary in this case is modelled as being passive and in the reception range of all packets exchanged between n_{alice} and n_{bob}. The adversary, n_{eve}, is also assumed to be at a distance greater than 2m from both n_{alice} and n_{bob}. If the adversary is too close to n_{alice} and n_{bob} then the adversary's channel might correlate with legitimate party channels (i.e. if the distance between to n_{eve} and any of the legitimate nodes

is less than 2m).

Given the assumptions above, the adversary must not be able to recover the common key generated by n_{alice} and n_{bob} or use a compromised session key to calculate session keys that were used prior to or after the compromised session.

4.3 Overview of Proposed PLSKG Scheme

The proposed PLSKG scheme consists of three main stages as shown in figure 4.1. These are i) randomness sharing, ii) key reconciliation and iii) privacy amplification. In the randomness sharing phase, n_{alice} and n_{bob} trade N_{i} messages over N_{j} different frequencies. They then filter and process those samples as shown in figure 4.1 to formulate an initial key.

In the key reconciliation stage, that key is reconciled using ECCs. This process involves n_{alice} generating a random number and encoding it with an ECC. A one-time pad is then performed with the encoded random number and the key that has been generated by n_{alice}. The result is sent over the wireless channel to n_{bob}. n_{bob} then processes the information received as shown in figure 4.1 to reconcile its key with the key generated by n_{alice}. After both n_{alice} and n_{bob} have generated the keys, the process moves on to the privacy amplification stage.

The privacy amplification stage serves two main purposes: i) to ensure perfect forward and backward security and ii) to ensure that the final key has bits that are well distributed. To achieve this goal, the privacy amplification stage formulates the key in such a way that a new session key forms a hash chain that uses both the previous key and the hashed value of the current physical layer generated key as arguments. Formulating the current session key in this manner makes it impossible for an intruder with knowledge of compromised session key K_{i} to compute the key that was used prior to or after K_{i}. The details of these three stages will be presented in the separate sections below.

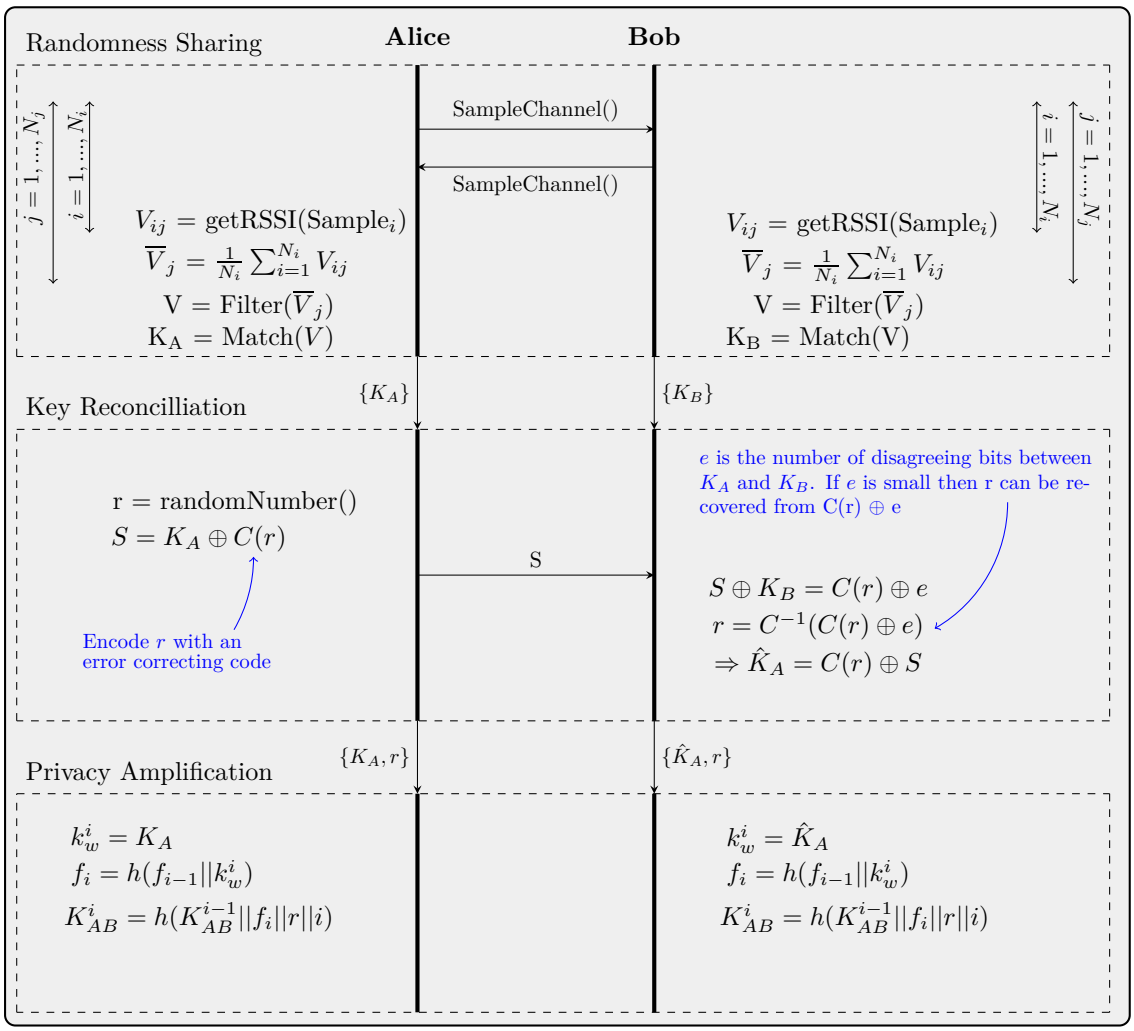


Figure 4.1: Key Generation and Refreshment Process

4.4 Randomness Sharing & Quantisation

Randomness sharing is arguably the most important stage in the PLSKG process because it is the stage when a physical layer parameter (in our case the RSS) is observed. In this section we will first look at the limitations and implementation issues that arise when using OTS 802.15.4 compliant WSNs. We will then briefly investigate the impact that channel fading has on the variability of static WSN channels, before detailing the full procedure for randomness sharing used in our proposed PLSKG scheme.

4.4.1 Implementation Issues with Randomness Sharing on Real Nodes

A WSN node consists of sensors, a transceiver and a microcontroller. A user wishing to deploy a PLSKG scheme has the option of i) performing the entire key generation procedure in hardware within the transceiver where all the other physical layer tasks are performed, or ii) just sampling a physical layer parameter from the transceiver and using that parameter as a source of randomness for a key generation procedure taking place in the microcontroller. Opting for the former solution would mean using the transceiver with physical layer functions for key generation built in, which would hinder quick adoption and deployment of the scheme. The latter option involves employing a current 802.15.4 compliant transceiver (such as the popular CC2420 transceiver [2] used in TelosB WSN nodes) which is used in nodes to sample a physical layer variable such as the RSS and have the other stages of SKG implemented as software on the microcontroller. This approach has been the most popular one used in practical WSNs, and such as in [1, 41].

The most useful and hence most intuitive channel parameter to use as a shared source of randomness between two parties is the channel response (or multiplicative fading loss) h. The channel response encapsulates the amplitude and phase changes that a transmitted signal goes through whilst travelling through the wireless medium. The channel response also has two degrees of freedom upon which to generate keys - the channel amplitude and the channel phase - and will thus yield a better key generation rate than parameters with just one degree of freedom. When trying to generate physical layer keys using OTS transceivers, the channel response will be most likely unavailable, so another parameter such as the RSS might be used instead. OTS transceivers do not compute and pass on a channel response estimate because it is not directly needed for communication to occur so it does not warrant additional computational resources to estimate and provide it. Even if the channel response was estimated and provided on a symbol-by-symbol basis, a device running at a frequency much higher than a microcontroller's frequency would be needed to sample the values.

4.4.2 Sources of Channel Fading in WSNs

PLSKG schemes rely on sampling the channel response h. Thus the rate at which we can generate keys at the physical layer is limited by how much the channel varies in time, frequency and space. If a node is moving, the frequency received will differ from the frequency transmitted according to the formula 4.5 below because of the Doppler shift [19]. The relationship between the receive and transmit frequencies f_{RX} and f_{TX} is defined below:

$$\begin{aligned} f_{\mathrm{RX}} &= f_{\mathrm{TX}} + \Delta f \\ &= f_{\mathrm{TX}} + \frac{||\vec{v}(t)||}{\lambda} \cos\theta(t) \\ &= f_{\mathrm{TX}} + f_{\mathrm{MAX}} \cos\theta(t) \end{aligned} \tag{4.5}$$

Here, $\overrightarrow{v(t)}$ is the relative velocity between the two nodes, λ is the wavelength of the transmitted wave, $\theta(t)$ is the angle of signal arrival at time t, and f_{MAX} is the maximum Doppler shift.

This shift causes a time-varying channel and fast fading. The coherence time t_{c} is the time interval where we have $f_{\mathrm{RX}} \approx f_{\mathrm{TX}}$. In the case of mobile nodes, this variability can be exploited to generate keys by sampling the channel every t seconds where $t > t_{\mathrm{c}}$. In the case of static nodes, where $\overrightarrow{v(t)} = 0$, there is no time-varying channel fading, so the channel remains largely unchanged for a long period, which reduces the key generation capacity.

There is also an opportunity to exploit the frequency selective characteristics of a channel to generate keys. Frequency selectivity is caused by the propagating environment in which the nodes are communicating and the rate at which nodes are transmitting data. Frequency selectivity when transmitting symbols over a channel is caused by multipath propagation which causes time dispersion, leading to Inter-Symbol Interference (ISI). Multipath propagation results in different copies of a signal to take different paths from a transmitter to a receiver. Since these paths are of different lengths, they arrive at the receiver at different times and cause ISI.

The coherence bandwidth is a statistical measurement of the range of frequencies over which the channel can be considered flat. The X% coherence bandwidth is

defined in [52] as being equal to the value of Δf such that:

$$\frac{X}{100} = \frac{E\{h(f) * h(f + \Delta f)\}}{E\{|h(f)|^2\}} \tag{4.6}$$

Here, X is the correlation factor, $h(f)$ is the channel response at frequency f, $|h(f)|$ is the channel gain and $E\{x\}$ is the expectation of random variable x. If two frequencies are within the X% coherence bandwidth then the channel responses at the two frequencies will have a correlation coefficient of at least X/100 with the best possible correlation between channels being 1. The 50% and 90% coherence bandwidths, for example, can be shown to be equal to $1/5\sigma_\tau$ and $1/50\sigma_\tau$ respectively. Here, σ_τ is the delay spread.

For a channel to be very frequency selective, we usually need to have a signal bandwidth greater than the 50% channel coherence bandwidth, namely we need $B_\mathrm{s} > B_{\mathrm{C},50}$. In non-line-of-sight indoor WSNs we will typically have the value of σ_τ in the range from $\approx$ 8ns to $\approx$18ns [53] and an allocated bandwidth of 2MHz per channel in 802.15.4 networks. Hence we can see that even if we take the most severe case of delay spread given as 18ns in [53] we have $B_{\mathrm{C},50} \approx 11$ MHz. This means that just one channel does not provide enough frequency dependent variation on which to base key generation, but if we use the entire spectrum of available 802.15.4 channels, we can exploit the fact that each individual channel has a different frequency response to generate keys.

The above problem with static channels not having a lot of variability is a key challenge when trying to formulate PLSKG schemes for WSNs as nodes are usually non-mobile and utilise low bandwidths. A common approach to circumventing this problem is to induce variability at the nodes by limiting the schemes to mobile nodes [48], switching transmission channels [1] or altering hardware characteristics every time we sample the channel [37]. The presence of some frequency selective fading means that we can exploit the variability of slow fading loss (L_p) at different centre frequencies by switching between different frequency channels.

4.4.3 Randomness Sharing in Proposed PLSKG Scheme

The scheme we propose alternatively hops between L channels according to a predetermined frequency hopping schedule S known to both nodes n_{alice} and n_{bob} and computes the mean RSS on each channel at each end. In order for the nodes to generate keys they need to change their frequency channels synchronously, and the frequency hopping schedule S helps them do that. The schedule also plays an important part in ensuring that the generated keys vary with time even in static channels. Going through the key generation process multiple times in quick succession will still produce different keys because the set of channels used and the order in which they are used will change with each iteration according to the synchronised schedule S. The generation of S can be arbitrary, so legitimate parties can easily generate S by seeding a pseudo number generator and iterating it after every sample period.

After this, the samples are first processed by removing the direct current (DC) component in the samples (this involves calculating the mean of the samples and then deducting that mean from each sample value). The average value (i.e. mean) of the resulting samples after this processing will then be zero. This is done so that the differences in transmit powers between n_{alice} and n_{bob} do not affect the key generation process. After this, each value in the resulting sequence is matched and swapped with its grey code equivalent. Converting to grey code helps minimise bit disagreement between n_{alice} and n_{bob} by ensuring that the difference between any two adjacent values is zero.

After this we take our bit sequence and use it to formulate a weak key. In this case we will define the weak key as the intermediate key that comes out as a result of the randomness sharing phase and is used in the key reconciliation process. If we need a longer weak key, we can change the schedule S and repeat the process above to get another longer bit sequence. We can repeat this process until we have the required number of bits in our weak key, although the longer the key required the higher the energy cost. The rationale behind formulating weak keys in this way and not

just using directly quantised values is to further add resilience to transmit power differences between nodes that would otherwise cause mismatches and to increase the key variability between sessions. The randomness sharing procedure is shown in algorithm 1.

Input: Channel Frequency Hopping Schedule S
Result: K_{RS} - Key from Randomness Sharing Stage
for *each channel index* $j \in [1, N_{\mathrm{j}}]$ **do**
  Channel Frequency $\leftarrow S(j)$;
  for *each sample number* $i \in [1, N_{\mathrm{i}}]$ **do**
    $V_{\mathrm{ij}} \leftarrow \mathrm{getRSSI}(\mathrm{Sample_i})$;
  end
  $\overline{V}_{\mathrm{j}} = (1/N_{\mathrm{i}}) \times \sum_{\mathrm{i}=1}^{N_{\mathrm{i}}} V_{\mathrm{ij}}$;
end
$\overline{V} = (1/N_{\mathrm{j}}) \times \sum_{\mathrm{j}=1}^{N_{\mathrm{j}}} \overline{V}_{\mathrm{j}}$;
for *each channel index* $k \in [1, N_{\mathrm{k}}]$ **do**
  $K_{\mathrm{RS}}(k) \leftarrow \mathrm{convertToMatchingGrayCode}(\overline{V}_{\mathrm{k}} - \overline{V})$;
end

Algorithm 1: Pseudocode for Randomness Sharing Stage

4.5 Information Reconciliation

Information reconciliation is the process of using quantised values of a physical parameter to generate a common key between two communicating parties. This stage aims to reduce the bit errors between two communicating nodes by having the nodes share some information that would help them reconcile their keys. The main existing approaches proposed for doing this include the use of error correcting codes or the exchange of some information regarding the quantisation of keys to help reconcile keys. The former approach is popular in 802.11 networks (particularly using LDPC (Low Density Parity Check) codes (ECCs) or BCH (Bose-Chaudhuri-Hocquenghem) codes) whilst the later approach is popular in a WSN setting.

The use of the Error Correcting Code (ECC) process involves n_{alice} choosing a random number, r, encoded with an error correcting code C, performing a one-time pad with its key K_{a} to create a syndrome S, and sending it to n_{bob}. n_{bob} then performs a one-time pad with its key K_{b} to get an estimate of the encoded data and then uses it to compute K_{a}. More formally, the above process can be defined

as follows:

$$
\begin{aligned}
&\text{Alice :} \\
&\qquad s = C(r) \oplus K_{\mathrm{a}} \\
&\text{Bob :} \\
&\qquad \hat{r} = \mathrm{Decode}(s \oplus K_{\mathrm{b}}) \\
&\qquad\quad = \mathrm{Decode}(C(r) \oplus K_{\mathrm{a}} \oplus K_{\mathrm{b}}) \\
&\qquad\quad = \mathrm{Decode}(C(r) \oplus e) \\
&\qquad\quad = r \quad (\text{if } H_{\mathrm{D}}(e, \mathbf{0}) < \text{some threshold } t) \\
\implies &\hat{K}_{\mathrm{a}} = C(\hat{r}) \oplus s
\end{aligned}
\tag{4.7}
$$

Here, the bitwise $\oplus$ operation is the exclusive OR operation, e is the error vector and $H_{\mathrm{D}}(a, b)$ is the hamming distance (i.e the number of disagreeing bits) between numbers a and b.

The choice of ECC in a WSN setting will depend on the required key length as different codes have different decoding capabilities and resource requirements. The hamming code, for example, is easier than other resource intensive codes such as LDPC codes to implement but it will only have a decoding threshold of $t = (d_{\min} - 1)/2$, where $d_{\min}$ is the minimum distance between codes, whilst LDPC codes can correct many more errors as the key size increases but consumes more resources because of its iterative decoding process.

The chosen ECC needs to be able to correct errors but it should not be able to correct a large number of errors. If the ECC can correct a large number of errors then it may be possible for n_{eve} to reconcile its key with n_{alice} even though the difference between their keys is fairly large. Hamming and Polynomial codes have the advantage of having a clearly defined error correcting capability, so we know that they will only be able correct errors up to a fixed threshold. Polynomial codes are similar to hamming codes in terms of both error correction capability and resource intensiveness, so we propose the use of either hamming codes or polynomial codes

for error correction.

Polynomial codes create a codeword $c(x)$ by using the original message vector $m(x)$ and a primitive polynomial $g(x)$, where $c(x)$, $m(x)$ and $g(x)$ are all in polynomials with coefficients belonging to the Galois field of two elements (i.e. GF(2)). The code can correct one bit error in the received message $r(x)$ by computing the remainder after dividing $r(x)$ with $g(x)$. This can be seen below:

$$c(x) = m(x)g(x)$$
$$r(x) = c(x) + e(x)$$

$$\text{rem}\left(\frac{r(x)}{g(x)}\right) = \text{rem}\left(\frac{c(x)+e(x)}{g(x)}\right) = \text{rem}\left(\frac{e(x)}{g(x)}\right)$$

Here, $e(x)$ is the error monomial. The set of all values of $rem\left(\frac{x^i}{g(x)}\right)$ for all i (i.e. for all error monomials) is precomputed and stored in a look-up table and thus the error location i can be computed at the receiver, provided only a one bit error has occurred.

A hamming code is specified by a generator matrix G and a parity check matrix H such that $HG^T = 0$. The code is computed as $c = mG$ and the received vector $\vec{r} = \vec{c} + \vec{e}$ is decoded by first computing a value called the syndrome, s , by $s = Hr^T$ and comparing which column of parity check matrix H matches with s. Here, the message vector is m and the error vector is e. The matched column is the error location of the single bit error. If $s = 0$, then there is no error. A polynomial code with a message length k and a code length n can be represented as a cyclic hamming code by setting:

$$G = \begin{pmatrix} g(x) & xg(x) & ... & x^k g(x) \end{pmatrix}^T \tag{4.8}$$

$$H = \begin{pmatrix} \left(\text{rem}\left(\frac{x^1}{g(x)}\right)\right)^T & ... & \left(\text{rem}\left(\frac{x^n}{g(x)}\right)\right)^T \end{pmatrix} \tag{4.9}$$

The design of our proposed scheme uses a series of Hamming codes with an addi-

tional parity bit with a code length of $n = 8$ and message length $k = 4$ to achieve error correction. Using a small code length helps keep the complexity of the key reconciliation stage low. The key is first interleaved by using a (4×8) block interleaver before encoding. Interleaving is used in order to make the ECC more robust to burst errors, so the scheme works even if a particular segment of the key has a high density of errors.

The interleaved key is first divided into n chunks, and each chunk is then padded with a Hamming code encoded random bit sequence $(C(r))$ to form a set of syndromes. Afterwards, these syndromes are sent from node n_{alice} to node n_{bob} for n_{bob} to carry out key reconciliation. When n_{bob} first receives the set of syndromes, n_{bob} performs the following for each syndrome. n_{bob} first tries to recover the original encoded key segment by performing a one-time pad of the received syndrome with the key that n_{bob} has measured. After performing this operation, n_{bob} will obtain the original encoded random number with $H_{\text{D}}(e, 0)$ errors, where e is the error vector $(K_{\text{a}} \oplus K_{\text{b}})$.

Using the error vector e, n_{bob} can then recover the original error free encoded random sequence $C(r)$ by decoding $C(r) \oplus e$ to get r and then encoding the result to recover $C(r)$. After this, the key measured by n_{alice} can be recovered by computing $C(r) \oplus s$. The random number, r (which is encoded to create $C\,(r)$), is generated using the Park-Miller Minimal Standard Generator. This generator is a multiplicative linear congruential generator ($r = (a \times s) \bmod (2^{31} - 1)$) with a = 16007 and the initial seed value s sourced by our scheme from the lower bits of the transceiver's automatic gain control (AGC) magnitude register (the transceiver datasheet specifies that these lower bits can be used for random number generation). After error correction both n_{alice} and n_{bob} can then proceed to the privacy amplification stage.

4.6 Privacy Amplification

The general idea behind privacy amplification is to ensure that if the number of bits in the key generated after reconciliation by a key generation scheme is greater

than the entropy of the key, we need to adjust the size of the key so that it aligns well with the entropy of the key. Take for example the following reconciled 32 bit key, K_{AB}, that was quantised by looking at the number of deep fades that a chip sequence experienced whilst travelling over a channel:

$$\begin{aligned} \mathsf{K}_{\mathrm{AB}} &= 10111101111111111110111011111101 \\ H(\mathsf{K}_{\mathrm{AB}}) &= \mathrm{length}(\mathsf{K}_{\mathrm{AB}}) \times \\ &\quad \left(\Pr(0) \log_2 \frac{1}{\Pr(0)} + \Pr(1) \log_2 \frac{1}{\Pr(1)} \right) \\ &\quad << \ \mathrm{length}(\mathsf{K}_{\mathrm{AB}}) \end{aligned} \tag{4.10}$$

Here, $H(\mathsf{K}_{\mathrm{AB}})$ is the entropy of K_{AB}. So we need to have the key to be of length $l \approx H(\mathsf{K}_{\mathrm{AB}}) < \mathrm{length}(\mathsf{K}_{\mathrm{AB}})$ for the key to have a level of cryptographic security that corresponds to its length. This can be done by using privacy extractors or by hashing the long key and choosing the first n bits of $H(\mathsf{K}_{\mathrm{AB}})$ as the final key K_{AMP}.

$$\begin{aligned} \mathsf{K}_{\mathrm{AB}} &= 10111101111111111110111011111101 \\ \mathsf{K}_{\mathrm{AMP}} &= h(\mathsf{K}_{\mathrm{AB}})|_{\mathrm{BIT}\ 0}^{\mathrm{BIT}\ (n-1)} \end{aligned} \tag{4.11}$$

The calculations above assume n_{eve} observes a completely decorrelated channel from n_{alice} and n_{bob}. In order to evaluate n, we would need to know n_{eve}'s channel statistics and thus in our scheme we take n as being equal to $\mathrm{length}(\mathsf{K}_{\mathrm{AB}})$ meaning that the resultant generated key will have a length that is longer than its true entropy. This keeps the complexity and hence the computational cost low.

In our proposed key generation scheme, we propose that the privacy amplification should be done in a way that the generated keys refresh the current session key to formulate the next session key and in so doing form a hashed key chain. In other words, a physical layer generated key in this instance is a function of all the previous keys that have ever been generated. This can be achieved by computing the final key $\mathsf{K}_{\mathrm{AB}}^{i}$ using the previous session key $\mathsf{K}_{\mathrm{AB}}^{i-1}$, reconciled weak key k_{w}^{i}, the recovered random number r and the iteration number i as:

$$\begin{aligned} \mathsf{K}_{\mathrm{AB}}^{i} &= H(\mathsf{K}_{\mathrm{AB}}^{i-1} || f_{\mathrm{i}} || r || i) \\ f_{\mathrm{i}} &= H(f_{\mathrm{i-1}} || \mathsf{k}_{\mathrm{w}}^{i}) \end{aligned} \tag{4.12}$$

Here, f_{i} is a hash chain of reconciled weak keys, with $f_0 = H(\mathsf{k}_{\mathrm{w}}^0)$. f_{i} is used to ensure that only a node with knowledge of the previous key generation session can generate the next session key. After n_{alice} and n_{bob} have derived a key, challenge-response authentication can then be undertaken to make sure the two generated keys agree. If the two keys do not agree then key generation can be attempted again. This will prevent key error propagation, where one error leads to more errors in the subsequent keys. The agreed $\mathsf{K}_{\mathrm{AB}}^i$ can then be used for communication between n_{alice} and n_{bob}.

4.7 Implementation, Evaluation and Comparison

In order to evaluate the practicality of our proposed PLSKG scheme and compare it with the most relevant existing method in [1] as will be elaborated later, we have implemented them using the NesC language [54] on a pair of TelosB WSN nodes running the TinyOS operating system. Experiments in a line-of-sight (LOS), indoor office setting were run over a number of distances and at a number of transmit power levels in order for the correlation between these factors and the Successful Key Reconciliation (SKR) to be evaluated. The nodes were static during the key generation process and the environment was an office working environment. The RSS values used are the ones measured and reported by the CC2420 transceiver that constitutes the TelosB nodes under test. The CC2420 has a stated RSS dynamic range of 100dB and a stated RSS accuracy of ±6dB with RSS linearity of ±3dB. The antennas in use were omnidirectional antennas. The graphs showing the observed SKR vs distance relations can be seen in figures 4.2 - 4.3. The graphs also show the distance between nodes versus the SKR rate at different transmit powers. Each curve in the graphs is a third order polynomial best fit curve of the data points.

From the graphs, it is clear to see that the SKR is very high (near 100%) at short distances but decreases with the distance and also decreases with lower node transmit powers. These results show that PLSKG can be a suitable alternative for the implementation of soft key generation in WSN nodes. In particular, a key can be generated and used to refresh session keys of a WSN node and in so doing help to enforce the forward security of the WSN node. This would make it very hard for an

attacker who does not have all the keys generated over all previous key generation sessions to discover the current key.

As the distance between n_{alice} and n_{bob} increases, the signal-to-noise (SNR) ratio at the receiving end decreases. This decrease in SNR makes the estimation of the reciprocal component of the channel (and hence the RSS) harder. From the RSS we have $RSS \approx P_{\text{r}} + P_{\text{n}} \pm 3$. Here the ± 3 dB component is due to the stated linearity in the calculation of RSS on the TelosB's transceiver [2]. As the distance increases (or the transmit power reduces) the receive power (P_{r}) reduces. This reduction in P_{r} causes the share of the non-reciprocal component of the RSS ($P_{\text{n}} \pm 3$) to increase as a proportion of the total RSS. This then causes bigger disagreements in measured RSS between n_{alice} and n_{bob}. This then reduces the SKR rate. This is clearly visible by looking at the shapes of the curves in figures 4.2 - 4.3.

In order to evaluate the performance of the proposed PLSKG scheme, we have implemented the most relevant and representative PLSKG scheme for WSNs, which is the scheme in [1], for comparison. Current state-of-the-art PLSKG schemes for WSNs usually use a form of iterative quantisation to achieve key reconciliation. A popular representative example is the scheme proposed in [1]. Its key reconcilement proceeds as follows. n_{alice} chooses a value t called a tolerance value and then quantises the observed RSS samples with a quantisation level of $\Delta L = 2t$ (i.e. rounds off each sample to the nearest multiple of ΔL). Node n_{alice} then sends the quantised values, tolerance, difference between the quantised values, and observed values to n_{bob}. n_{bob} then uses the received information to quantise and then reconcile its key with n_{alice}.

After the key reconciliation stage, n_{alice} and n_{bob} trade a challenge-response message to ascertain if they have successfully generated a common key. If the challenge-response fails, the value of ΔL is incremented and the key generation process then loops back to the beginning of the key reconciliation stage. This means that the quantisation interval is increased with each iteration. This process continues until n_{alice} and n_{bob} establish a common cryptographic key.

The PLSKG scheme proposed in [1] was implemented and experiments were conducted with the quantisation interval fixed at ΔL, where $\Delta L \in \{3, 4\}$. The results of the experiments are shown alongside the results of the proposed scheme in figures 4.2 - 4.3. From the graphs it is clear that the proposed scheme performs better than the scheme in [1] for $\Delta L = 3$ but slightly underperforms [1] for $\Delta L = 4$. The scheme proposed in [1] (and other similar iterative quantisation PLSKG schemes) generally performs better as ΔL is increased but in the following section we will prove that increasing values of ΔL in these schemes reduces key entropy and thus stands to compromise security.

The graphs in figures 4.2 - 4.3 show plot points along with the second order best fit curve. Due to the nature of the curve fitting process, the maximum point in the best fit curve can exceed 1 even though the points themselves will never exceed 1. The unit used for displaying the power is dBm, which can be converted to power in units of watts using the formula $10^{(x-30)/10}$, where x is the power in dBm. The graphs show that when using purely quantisation schemes, the key generation success rate reduces with increasing transmit power with the rate at power level 0dBm and quantisation interval 4 falling to around 0.9 at 10m and the rate at -3dBm falling to around 0.8 at 10m. The graphs also show that in quantisation schemes the key generation Also note that the key generation success rate decreases exponentially with increasing distance and so if the distance was, for instance, doubled, then we would expect the key generation success rate to fall by a factor greater than 2.

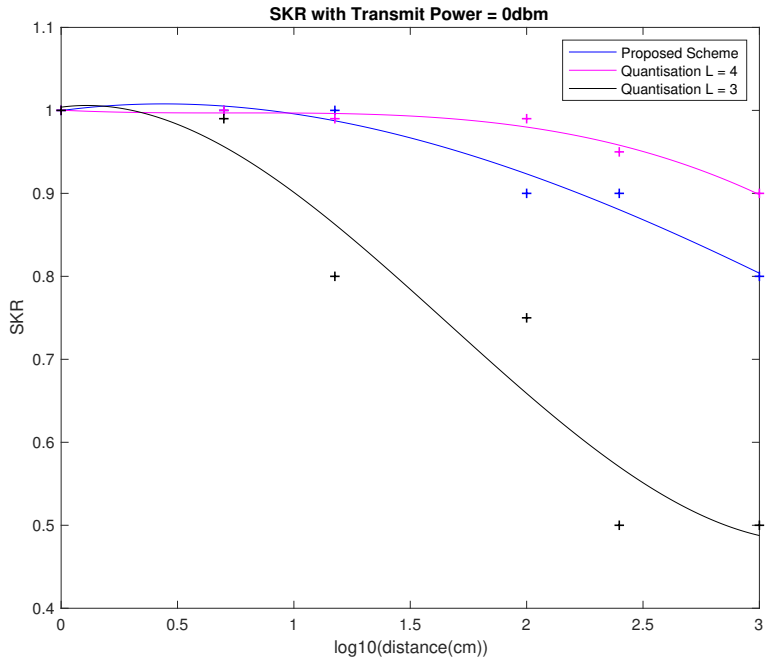


Figure 4.2: Successful Key Generation Rate (SKR) versus Node Distance at 0dBm for proposed scheme and for the scheme proposed by Wilhelm et. al [1] at different quantisation levels.

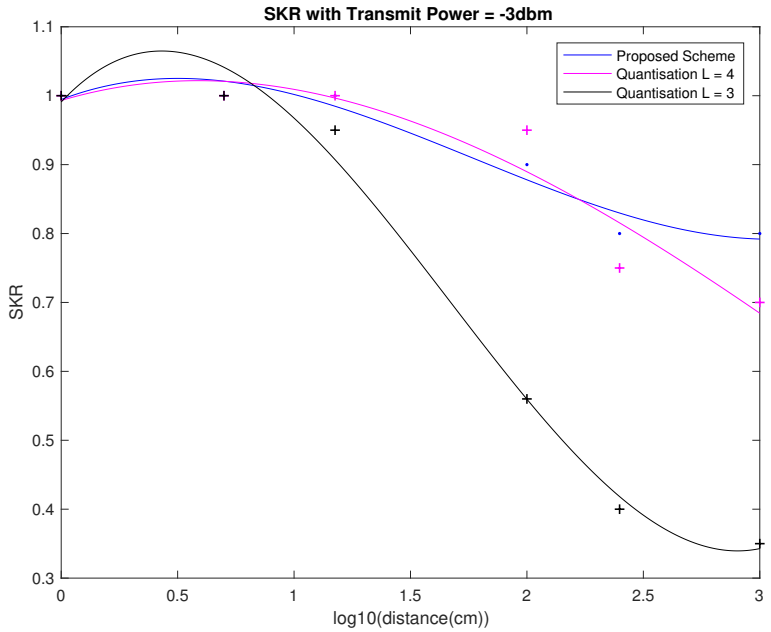


Figure 4.3: Successful Key Generation Rate (SKR) versus Node Distance at -3dBm for proposed scheme and for the scheme proposed by Wilhelm et. al [1] at different quantisation levels

There have been a few proposed key generation schemes over the past few years such as the one in [1] but our scheme differs substantially from all the other prac-

tical schemes for WSN nodes in a number of ways. Firstly, the biggest difference is the use of ECCs to improve the error correcting capability of key generation. ECCs have been proposed by a number of papers for 802.11 networks but no practical implementation of the approach has surfaced in the 802.15.4 landscape. The use of ECCs improves the scheme in a number of ways. It allows the key reconciliation layer to be designed and benchmarked separately to give a true layered design approach. It also allows different ECCs to be removed and placed depending on the power of the WSN node in question. For example, if a more powerful WSN node is used, the designer might opt to replace the hamming code used in our scheme with a slightly more powerful ECC without running the risk of breaking the system.

Secondly, unlike the other schemes, our approach does not quantise the RSS samples directly, so a mismatch with the transmit power does not alter the key generation capability. The only thing that matters in the key generation is the high frequency components (the movements in the RSS sequence) but not the DC component (the average RSS). This means that nodes do not have to be set to the exact same transmit power in the channel sampling stage for successful key generation and refreshment to occur.

Thirdly, the scheme provides a mechanism on which to generate new keys which uses not only the current state of the physical layer but also previous session keys (and hence indirectly using previous physical layer states). This helps make the scheme forward and backward secure.

In other 802.15.4 PHY layer schemes such as in [1], a single bit disagreement in the quantised RSS samples causes the final key to disagree. These schemes combat this by increasing the distance between quantisation levels until the two keys agree, with no limit on how much the maximum distance can be. This poses a security risk as the distance between quantisation levels could potentially get very big. In our scheme, errors are corrected a lot more efficiently with the exact capability of the error decoding process being clearly quantifiable.

4.8 Security Analysis

4.8.1 Security Comparison of Key Reconciliation in Proposed Scheme and State-Of-the-Art PLSKG Schemes

In this subsection we show that using iterative reconciliation is inefficient and potentially insecure because the entropy of a sequence quantised with quantisation level ΔL ($X_{\Delta \mathrm{L}}$) is lower than the entropy of the original observed sequence (X). To do this we need to derive and analyse the value of $\alpha_{\Delta \mathrm{L}}$ for increasing ΔL, where we define $\alpha_{\Delta \mathrm{L}}$ as the ratio defined in equation 4.13 below. The ratio of $H(X)$ to $H(X_{\Delta \mathrm{L}})$ should be 1 to $\alpha_{\Delta \mathrm{L}}$. We first begin by getting an expression for $\alpha_{\Delta \mathrm{L}}$ for the case where X is drawn from a uniform distribution. After this, we will proceed to deriving $\alpha_{\Delta \mathrm{L}}$ in the general case and then finally provide an expression of $\alpha_{\Delta \mathrm{L}}$ in the case where X has RSS values sampled from the RSS channel. In all the cases we will prove that $\alpha_{\Delta \mathrm{L}}$ is less than 1 and hence the entropy of the generated key is inversely proportional to ΔL.

$$\alpha_{\Delta \mathrm{L}} = \frac{H(X_{\Delta \mathrm{L}})}{H(X)} \tag{4.13}$$

In the case where X is drawn from a uniform distribution and $\mathcal{X}$ is the set of all possible outputs (i.e. the range of X). The number of elements in $\mathcal{X}$ is the cardinality of $\mathcal{X}$ ($|\mathcal{X}|$). The entropy of X can be evaluated as:

$$\begin{aligned} H(X) = -\sum_{\mathrm{x}} p(x) \log p(x) &= -\sum_{\mathrm{x}} \frac{1}{|\mathcal{X}|} \log \frac{1}{|\mathcal{X}|} \\ &= \log |\mathcal{X}| \end{aligned} \tag{4.14}$$

where $p(x)$ is the probability distribution of x. The quantisation operation rounds off values into the nearest multiple of ΔL and so it maps $\mathcal{X}$ to a set we denote as $\mathcal{X}_{\Delta \mathrm{L}}$. This means that the number of elements in $\mathcal{X}_{\Delta \mathrm{L}}$ is $|\mathcal{X}|/\Delta L$. The entropy of $\mathcal{X}_{\Delta \mathrm{L}}$ is then:

$$H(X_{\Delta \mathrm{L}}) = \log \frac{|\mathcal{X}|}{\Delta L} \tag{4.15}$$

This then implies that $\alpha_{\Delta \mathrm{L}} = (1 - \log_{|\mathcal{X}|} \Delta L) < 1$ for the case when X is drawn

from a uniform distribution. The fact that $\alpha_{\Delta \mathrm{L}}$ will always be less than 1 is due to the fact that $|\mathcal{X}| > \Delta L > 1$, which causes the value of $\log_{|\mathcal{X}|} \Delta L$ to always take a value in the range (0,1).

To get the expression of $\alpha_{\Delta \mathrm{L}}$ in the general case we first define an integer L_{m} as being equal to $|\mathcal{X}|/\Delta L$ and $\mathcal{X} = \{1, 2, ..., |X|\}$. Quantising values in the range [1,L_{m}] yields ΔL and quantising values in the range [$L_{\mathrm{m}} + 1$, $2L_{\mathrm{m}}$] yields $2\Delta L$. Equation 4.16 shows how the quantisation process maps $\mathcal{X}$ to $\mathcal{X}_{\Delta \mathrm{L}}$ in the general case.

$$\begin{aligned} \mathcal{X} &\longrightarrow \mathcal{X}_{\Delta \mathrm{L}} \\ \{1, ..., L_{\mathrm{m}}, L_{\mathrm{m}} + 1, ..., 2L_{\mathrm{m}}, ...\} &\longrightarrow \{\Delta L, 2\Delta L, ...\} \end{aligned} \tag{4.16}$$

The probability distribution over the set $\mathcal{X}_{\Delta \mathrm{L}}$ is $p_{\Delta \mathrm{L}}(x)$, where $p_{\Delta \mathrm{L}}(x)$ can be calculated from $p(x)$ using equation 4.17. We can then calculate the value of $H(X_{\Delta \mathrm{L}})$ as shown in equations 4.18 and 4.19.

$$\begin{aligned} p_{\Delta \mathrm{L}}(x = n\Delta L) &= p((n-1)L_{\mathrm{m}} + 1) + ... + p(nL_{\mathrm{m}}) \\ &= \sum\nolimits_{\mathrm{x}=(\mathrm{n}-1)\mathrm{L_m}+1}^{nL_{\mathrm{m}}} p(x) \end{aligned} \tag{4.17}$$

$$H(X_{\Delta \mathrm{L}}) = -\sum_{\mathrm{x} \in \mathcal{X}_{\Delta \mathrm{L}}} p_{\Delta \mathrm{L}}(x) \log p_{\Delta \mathrm{L}}(x) \tag{4.18}$$

$$\begin{aligned} H(X_{\Delta \mathrm{L}}) = -(P_1 + ... + P_{\mathrm{L_m}}) \log(P_1 + ... + P_{\mathrm{L_m}}) - ... \\ -(P_{(\Delta \mathrm{L}-1)\mathrm{L_m}+1} + ... + P_{|\mathcal{X}|}) \log(P_{(\Delta \mathrm{L}-1)\mathrm{L_m}+1} + ... + P_{|\mathcal{X}|}) \end{aligned} \tag{4.19}$$

$$\begin{aligned} \alpha_{\Delta \mathrm{L}} &= \frac{-(P_1 + ... + P_{\mathrm{L_m}}) \log(P_1 + ... + P_{\mathrm{L_m}}) - ...}{-P_1 \log P_1 - P_2 \log P_2 - ...} \\ &= \frac{\log(P_1 + ... + P_{\mathrm{L_m}})^{(P_1 + ... + P_{\mathrm{L_m}})} + ...}{\log P_1^{P_1} P_2^{P_2} ...} \\ &= \frac{\log \prod_{n=1}^{\Delta L} \left(\sum_{\mathrm{i}=(\mathrm{n}-1)\mathrm{L_m}+1}^{nL_{\mathrm{m}}} P_{\mathrm{i}} \right)^{\left(\sum_{\mathrm{i}=(\mathrm{n}-1)\mathrm{L_m}+1}^{nL_{\mathrm{m}}} P_{\mathrm{i}} \right)}}{\log \prod_{\mathrm{i}=1}^{|\mathcal{X}|} P_{\mathrm{i}}^{P_{\mathrm{i}}}} \end{aligned} \tag{4.20}$$

The fact that $\alpha_{\Delta \mathrm{L}}$ is less than 1 is a consequence of the mathematical inequality shown in equation 4.21 for $n_{\mathrm{i}} \in (0, 1)$. This proves that quantisation will always

reduce the entropy of the original sequence, with the ratio of the original entropy to the quantised entropy being $1 : \alpha_{\Delta \mathrm{L}}$.

$$\frac{(n_1 + ... + n_{\mathrm{M}}) \log(n_1 + ... + n_{\mathrm{M}})}{n_1 \log n_1 + ... + n_{\mathrm{M}} \log n_{\mathrm{M}}} < 1 \tag{4.21}$$

To illustrate this point, a graphical representation of this phenomenon is shown in figure 4.4. Figure 4.4 shows a sequence of integers in the range $[-4, 4]$ and graphs that result from the quantisation of the sequence with $\Delta L = \{2, 3, 4\}$. From the graphs it is clear to see how using large quantisation intervals is detrimental to security as the entropy and hence the sequence variability of quantised signals is dramatically reduced.

The linear received signal power is log-normally distributed [55] and so its discrete form can be approximated by the log binomial distribution [56]. The probability distribution $p(x = i) = P_{\mathrm{i}}$ in the case when the RSS is what we are sampling can thus be expressed as follows:

$$P_{\mathrm{i}} = \binom{n}{i} p^i (1 - p)^{n-i} \tag{4.22}$$

where

$$n = \text{Number of RSS samples} \tag{4.23}$$

$$\sigma = \text{Standard Deviation} = \sqrt{np(1 - p)} \tag{4.24}$$

The standard deviation (σ) varies depending on the exact wireless communication environment but empirical studies have estimated σ to be in the range of 5 to 12dB depending on the environment [57].

The analysis above shows that iterative quantisation will negatively affect entropy with increasing ΔL, with the rate at which entropy degrades not only being a function of ΔL but also dependent on the actual probability distribution of RSS values. This is in contrast to our ECC based reconciliation which forces node n_{bob}

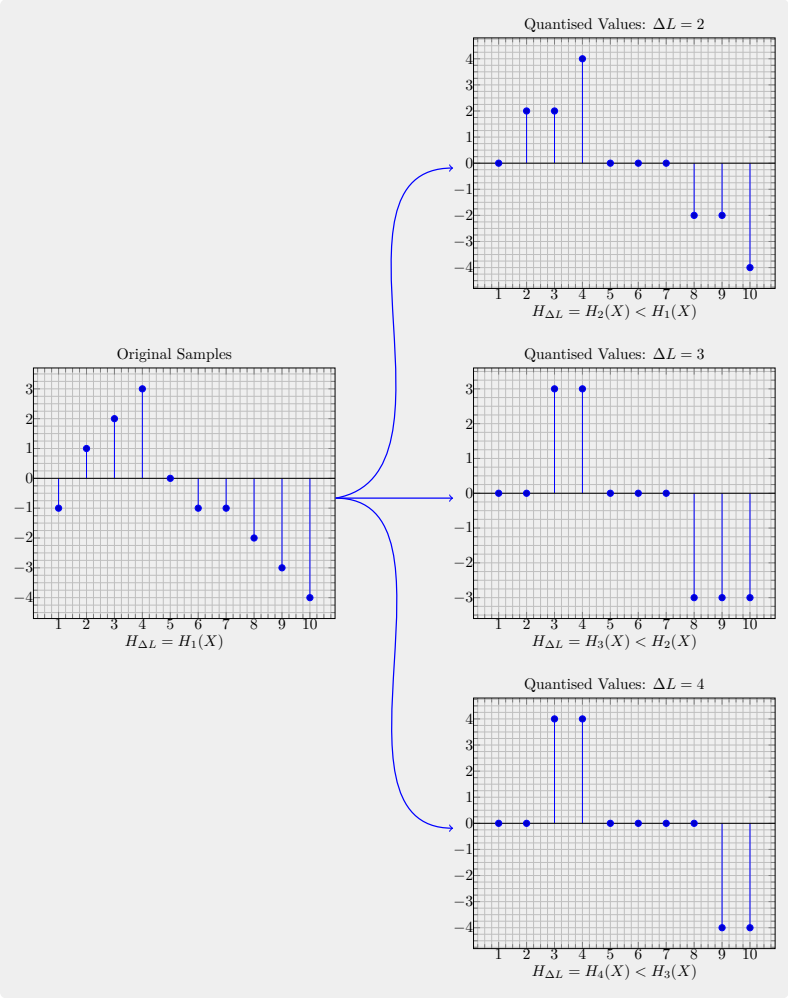


Figure 4.4: Figure showing how the quantisation level ΔL affects entropy

to reconcile its key to the original key measured by node n_{alice} and so does not necessarily reduce the entropy of the key in the reconcilement stage.

4.8.2 Randomness testing

In order for keys to be used for cryptographic processes it must be ensured that the generated keys have properties of randomness. The randomness of the generated keys was tested using the discrete frequency spectrum test, which is a part of a standardised cryptographic randomness test [58]. NIST is a standards agency which produces standardised randomness testing tools suitable for use in cryptographic purposes.

The spectral test works by taking the discrete fourier transform (DFT) of a bit stream and testing if the spectrum is similar to the spectrum that would be obtained

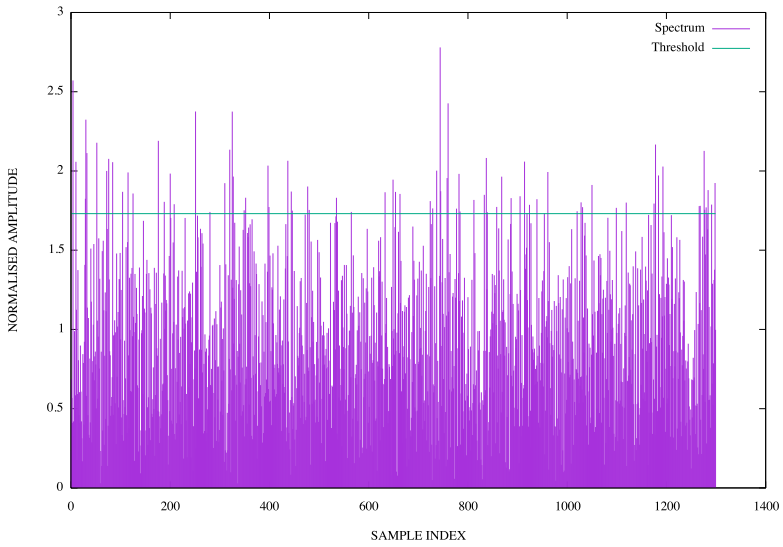


Figure 4.5: The spectrum of the key sequence and the location of the 95% threshold

from a perfect true randomness source. If a sequence is truly random, then its spectrum will approximately be flat because there will be no dominant frequency components. In addition to this, if a spectrum is random, then 95% of the frequency domain samples will be less than a threshold T where $T = \sqrt{n \log(20)}$ and n is the length of the sequence. The test works by first calculating N_{r}, the mean number of samples in the spectrum that would be below T in a truly random sequence and N_{s}, the number of samples in the spectrum that are below T in the sequence under test. The probability that the sequence under test is truly random is then calculated using the deviation of N_{s} from N_{r}. If that probability is over 99%, then sequence under test is deemed to be random [58]. The test proceeds as follows:

$$\text{KeyStream} = \{k_0 \| k_1 \| ... \| k_{\mathrm{N_k}-1}\} = x = \{0,1\}^n \Leftrightarrow \{-1,1\}^n \tag{4.25}$$

$$X = \mathrm{DFT}(x) = \sum_{\mathrm{k}=1}^{n} x e^{(j2\pi(k-1)/n)} = \sum_{\mathrm{k}=1}^{n} f(k) \tag{4.26}$$

$$\text{p-value} = 1 - \mathrm{erf}\left(\frac{|d|}{\sqrt{2}}\right) \tag{4.27}$$

where

$$j = \sqrt{-1} \tag{4.28}$$

$$N_{\mathrm{k}} = (\text{Number Of Keys Under Test}) \tag{4.29}$$

$$n = N_{\mathrm{k}} \times (\text{Number Of Bits Per Key}) \tag{4.30}$$

$$d = \frac{N_{\mathrm{r}} - N_{\mathrm{s}}}{\sqrt{0.05 N_{\mathrm{s}}/2}} \tag{4.31}$$

$$N_{\mathrm{r}} = 0.95 \times n \tag{4.32}$$

$$N_{\mathrm{s}} = \sum_{\mathrm{k}=1}^{n} \begin{cases} 1 & |f(k)| < T \\ 0 & |f(k)| > T \end{cases} \tag{4.33}$$

$$\mathrm{erf}(u) = \frac{1}{2\pi} \int_0^u e^{-t^2} dt \tag{4.34}$$

A small p-value indicates strong evidence against our null hypothesis (our null hypothesis is that the sequence under test is random). The NIST standard advises us to accept a sequence with a p-value greater than 0.01 as being random [58]. In order to test randomness, 75 different keys were computed and used to form a bit sequence of length 2400 bits. The sequence was then tested using the spectral test. The obtained spectrum can be seen in figure 4.5. The resultant p-value of the tested bit stream was 0.589 which means the sequence has randomness properties.

The keys were also tested to see if they are well correlated in a static environment. In this test, 75 keys were computed with a key refreshment period of one minute. For each key, the correlation between itself and each of the other keys was computed and the result plotted on the heat map shown in figure 4.6. The correlation coefficient between two keys is obtained by computing the cross correlation between the keys and then taking the maximum correlation coefficient (CC) from the resultant vector (this can be seen in equation 4.35). From figure 4.6 it is clear that different keys do not correlate highly with each other between sessions on the vast majority of occasions. The points in the map are keys which were generated by legitimate parties n_{alice} and n_{bob} in the same session. Out of all key correlations, there is only one rare occasion when subsequently generated keys correlated highly and only one

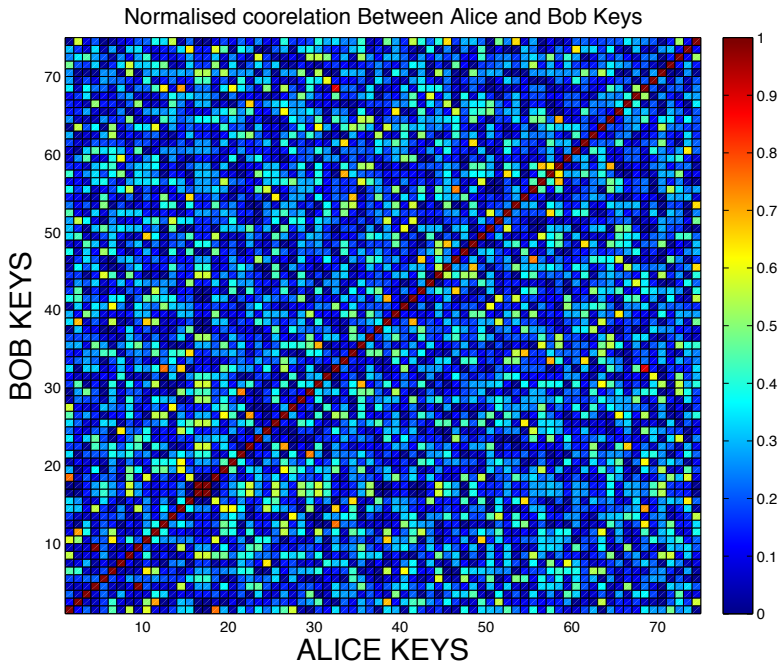


Figure 4.6: Correlation Coefficients between Keys (only keys with the same index should correlate highly)

other case of high correlation between key number 4 and key number 9. These relatively rare high correlations could have been caused by channel conditions not changing adequately enough between key generation intervals.

$$\begin{aligned} \text{keyA} = x = \{0,1\}^n \Leftrightarrow \{-1,1\}^n \\ \text{keyB} = y = \{0,1\}^n \Leftrightarrow \{-1,1\}^n \\ \text{CC} = |\rho_{\text{xy}}| \end{aligned} \tag{4.35}$$

where

$$\rho_{\text{xy}} = \frac{E[xy] - E[x]E[y]}{\sqrt{E[x^2] - [E[x]]^2}\sqrt{E[y^2] - [E[y]]^2}} \tag{4.36}$$

4.8.3 Security Analysis against Common Attacks

It is also important to evaluate the security of the scheme in order to understand the additional security benefits that our scheme brings in relation to existing key refreshment and generation schemes. The biggest threat facing sensor networks is arguably brought about by the fairly recent drive to connect them to the Internet

to create what is known as the Internet of Things (IoT). Connecting devices in this way leaves WSNs (which can have indirect access to the Internet via sinks / base stations) vulnerable to a wide range of attacks from remote users who have access to much more powerful computational resources.

It is important to make sure that if the key material that was originally deployed with the WSN and/or key material used in a particular session is compromised by a remote user, that user cannot use that information to discover any key material used in any other session. In other words, we want our scheme to achieve perfect forward and backward security.

A type of attack that WSNs are particularly vulnerable to is man-in-the-middle (MITM) attacks. The most direct way an adversary could try and compromise the process is by trying to snoop on communications between n_{alice} and n_{bob} and then running through the key generation process to generate a key. Tests were done with a third node, n_{eve}, being a distance of 2m away from n_{bob}. n_{alice} and n_{bob} went through the key generation process 75 times with n_{eve} also trying to generate a key from messages sent from n_{alice}. The correlation coefficient of the keys obtained from n_{bob} and n_{eve} are shown in figure 4.7. From the figure it is clear that the correlation between keys generated at n_{alice} and n_{eve} was not high, showing that this scheme can be used even in relatively dense WSN deployments.

The case outlined above is the case where n_{eve} is passive. In an active case, n_{eve} could try to and inject/broadcast to n_{alice} and/or n_{bob}. In this case, allowing only packets that have been appended with a message authentication code (MAC) to be used in the key reconciliation process will prevent malicious packets from being injected by n_{eve}. In the event that n_{eve} tries to influence the wireless environment by flooding the channel with malicious packets and in so doing raising the RSS of packets received in particular time intervals, n_{alice} and n_{bob} need to monitor the quality of the link between them by looking at the link quality indicator. An increase in RSS should correspond to an increase in Link Quality Index (LQI), so any inverse relationship between RSS and LQI will indicate some possible malicious activity.

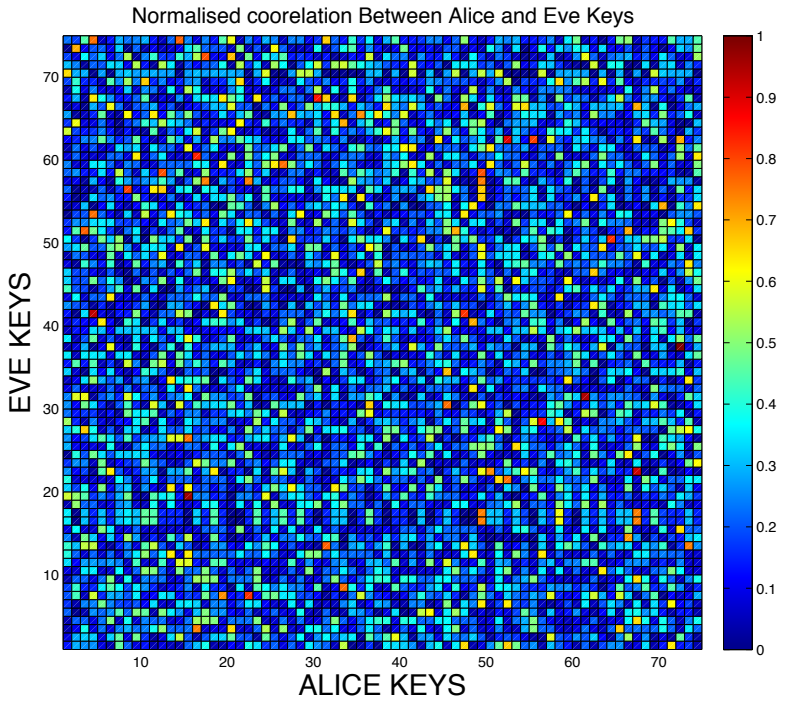


Figure 4.7: Correlation Coefficients between Legitimate and Adversary Keys, D = 2m

There is also the possibility of an adversary trying to disrupt the process by jamming the wireless channel. This could be done in two ways: i) the jammer might just jam the particular 802.15.4 channel being transmitted on by transmitting high power noise or ii) the attacker could flood the channel with 802.15.4 compliant packets. In the first case the 2.4GHz 802.15.4 PHY layer provides an inbuilt defence for this using direct sequence spread spectrum (DSSS) technology. DSSS works by encoding the bits to be sent with another pseudo random spreading bit sequence of a substantially higher data rate than the data sequence and then transmitting the result instead of transmitting the bits directly. Doing this has the effect of spreading the information sent over a large bandwidth and in so doing preventing narrowband indiscriminate jammers.

In the event of an adversary flooding the channel with 802.15.4 compliant packets, the use of DSSS will not prevent the attack. Relying on the MAC only will not prevent the attack as each malicious packet will need to be received, leading to a denial-of-service attack. In this type of the attack, actions need to be taken at the physical layer to minimise the impact. When the PHY layer transmits a packet, it prepends the packet with a synchronisation (SYNC) header. The header does not

contain any application specific information, it is used to make sure the transceiver's communicating can synchronise before the actual packet information starts. The SYNC header used in the CC2420 is shown in figure 4.8.

In order to minimise the impact of flooding attacks, legitimate parties could switch from the default 802.15.4 synchronisation header value of 0x00007A to a different header unique value on a per packet basis when generating keys. A unique SYNC header value will prevent flooding attacks because packet information (e.g destination address, packet length, etc.) will not even be read if the expected SYNC header preamble and received SYNC header preamble differ by more than a set threshold of bits (this threshold is configurable on the transceiver). The SYNC value is sent as plaintext and so would need to change synchronously on a per packet basis in order to foil attacks from sophisticated denial-of-service attackers who are snooping on SYNC headers and also flooding at the same time.

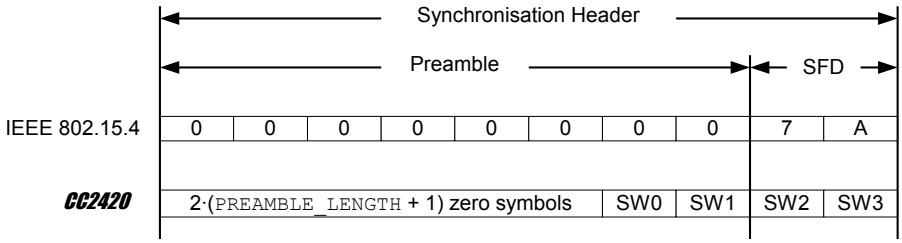


Figure 4.8: Synchronisation Header [2]

We now consider another attack scenario where the current session key is K_{j} and an adversary compromises the key material from a previous session K_{i}. If we had just hashed the key (together with other deterministic information) when moving to new sessions in order to refresh keys, as is the case with current WSN deployments, then session keys used before K_{i} would not be computable because a hash is irreversible but all session keys that come after K_{i} (including K_{j}) could be easily computed from K_{i} by just applying successive hashes. This is a major problem because at any time in the future, if the original key material K_0 that the WSN was deployed with is compromised, then all the session keys that have ever been used would be at risk. If, on the other hand, our proposed scheme is used, then the compromise of any session key will not compromise any other session key as the adversary will not be able to evaluate the physical layer values used to refresh session keys. This

is because the adversary would have no access to the values of f_i in equation 4.12, assuming that the advisory is not collocated with any one of the legitimate nodes. A big advantage of a PLSKG scheme is that an adversary who has not been locally there throughout the entire lifespan of the WSN network will find it very difficult to compromise a key, even if they know all the key information loaded on nodes at deployment.

4.9 Summary

In summary, this chapter has proposed a physical layer key generation scheme for resource constrained wireless sensor networks. The scheme allows low power devices to generate key securely using the RSS power channel that lies between them. This is accomplished by hopping through a variety of frequencies in order to increase variability and by using error correcting codes in the key reconciliation stage to correct any errors which may exist between the legitimate parties.

The proposed scheme is aimed at stationary nodes and uses off-the-shelf WSN devices. This is in contrast with most state-of-the-art schemes which require nodes to be moving during the key generation process. The scheme uses a combination of quantisation and error coding control to manage the key reconciliation stage of the key generation process. The thesis also provides a new proof that existing schemes that achieve key reconciliation by iterative quantisation reduce the entropy of the resultant key for every iteration with no known method of bounding the entropy loss, making it hard to estimate the entropy of the final generated key. If entropy is lost in the key reconciliation stage but there exists no way to evaluate or bound how much entropy is lost, then it is not possible to know how much entropy the final generated key has or even ascertain if it has any entropy at all relative to an adversary.

Randomness tests are conducted in order to ensure that the generated keys satisfy basic randomness properties. From the implementation and testing of the pairwise PLSKG scheme, we have shown that the generated keys indeed exhibit properties of randomness.

Chapter 5

Group Physical Layer Key Generation

5.1 Group Key Generation Fundamentals

Group key generation is concerned with the establishment of a common cryptographic key between more than two legitimate nodes. Group secure key generation aims to achieve this by using the physical layer wireless channel as a common source of randomness. The established key should not be computable by an adversarial node making channel observations via some degraded channel.

The key mechanism that allows for physical layer schemes to work is the presence of some environmental advantage that allows legitimate nodes to estimate parameters better than adversarial nodes. A condition which assures this (or the class of adversarial channel that guarantees this) is a physically degraded adversarial channel. A channel, h_{E} is said to be degraded with respect to another channel h_{Y} if the observations over that channel follow a distribution E which is equal to the sum of a distribution Y (the distribution of observations over h_{Y}) and another distribution which is independent from Y. A channel which is physically degraded with respect to another channel is guaranteed to be noisier. Thus if an adversarial channel is degraded with respect to all legitimate (or main) channel(s), then the main channel(s) is said to have some secrecy capacity, i.e. some key generation capacity.

As mentioned in chapter 2, in a lot of cases it can be difficult to ascertain if an adversarial channel is indeed degraded, but in these cases we showed that minimum

guarantees regarding the secrecy level can still be achieved.

The design of GPLSKG requires some modifications to the pairwise PLSKG scheme defined in the previous chapter (particularly regarding the key reconciliation stage) to enable the scheme to be more efficient in a group setting.

Let R_{L} be power lost when transmitting between two nodes n_{A} and n_{B} that are a distance d_{AB} apart. If the two nodes transmit probes at a transmit power level of P_{T}, then the receive power at each of the nodes will be $P_{\mathrm{RX}} \approx P_{\mathrm{T}} - R_{\mathrm{L}}(d_{\mathrm{AB}}) \pm P_{\mathrm{N}} = P_{\mathrm{T}} + Z_{\mathrm{AB}}$. Where $R_{\mathrm{L}}(d_{\mathrm{AB}})$ is the loss at distance d_{AB} and P_{N} is noise power over the channel, causing measurement inaccuracies, non-linearities etc. at the receiver. The distribution of the loss distribution of $R_{\mathrm{L}}(d_{\mathrm{AB}})$ with distance d_{AB} is dependent on the environment but has been shown in chapter 2 to follow a random Gaussian distribution in most circumstances.

Let an adversary, n_{E}, be a distance d_{AE} from n_{A} and a distance d_{BE} from n_{B}. The received power at n_{E} from n_{A} is $P_{\mathrm{E}} \approx P_{\mathrm{T}} - R_{\mathrm{L}}(d_{\mathrm{AE}}) \pm P_{\mathrm{N}} = P_{\mathrm{T}} + Z_{\mathrm{AE}}$. In general, if there exists a parameter P which can be estimated with uniform inaccuracy $\pm a$ by legitimate parties but with an accuracy of $\pm b$ by adversaries then there is secrecy or key generation capacity if $b > a$. This statement is true in general. If the distribution of the noise experienced on the main channel is N_1 and the distribution of the noise experienced on the degraded adversarial channel is at best N_2, then the key generation capacity per sample can be shown to be at least $H(Z_{\mathrm{AE}}) - H(Z_{\mathrm{AB}})$. The key generation capacity is the theoretical maximum number of bits that can be generated per sample, and this capacity is generally hard to achieve in practice. The generated key length has to be at most equal to the key generation capacity for the key to have an entropy that is equal to its bit length.

A general diagram that shows the relation between the channels of concern in key generation ($Z_{\mathrm{AB}}, Z_{\mathrm{AC}}$ and Z_{AE}) can be seen in figure 5.1a. It can also be shown that the secrecy capacity per channel sample, C_{s}, is:

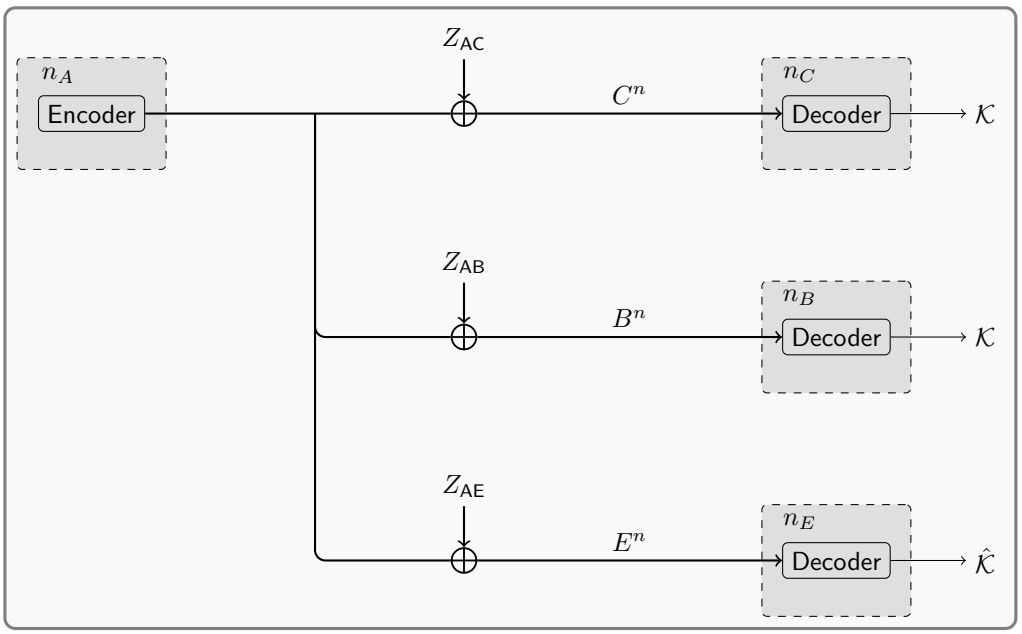
$$C_{\mathrm{s}} \geq C^{*}_{\mathrm{m}} - C^{*}_{\mathrm{e}} \tag{5.1}$$

where C^{*}_{m} is the capacity between legitimate nodes and C^{*}_{e} is the capacity between any node and the adversarial node.

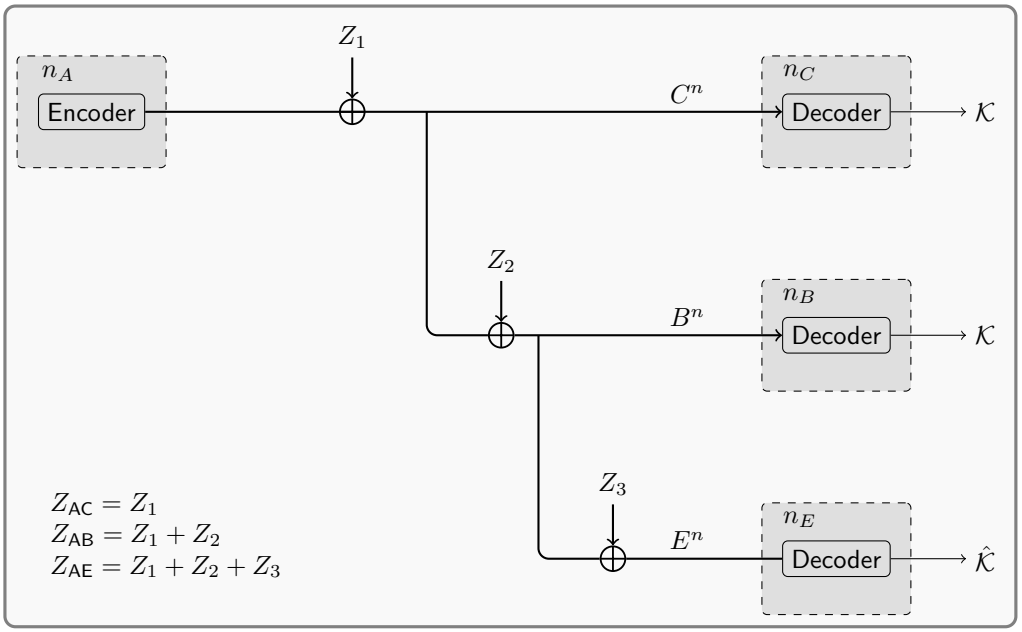
A random variable X_{i} is the distribution of the values observable at node n_{i}. In the special case where the adversarial channel is degraded with respect to legitimate channels as in figure 5.1b then the secrecy capacity can be shown to evaluate to satisfy:

$$C_{\mathrm{s}} \geq C^{d}_{\mathrm{s}} \tag{5.2}$$

A proof of the equation 5.2 can be found in [8]. It is difficult to compute the secrecy capacity of a particular generic wireless system but computing the degraded secrecy capacity gives a minimum secrecy rate C^{d}_{s} that the secrecy capacity C_{s} has to at least equate to. We can thus know that exchanging n messages between nodes taking part in group key generation can possibly extract a key of entropy nC^{d}_{s}.



(a) General Group PLSKG between three legitimate nodes and one adversary



(b) Group PLSKG between three legitimate nodes with degraded channels and one adversary

Figure 5.1: GPLSKG between three legitimate nodes and one adversary (n_{E})

5.2 Group Key Generation

This section gives an overview of the proposed GPLSKG scheme, including its key stages. The scheme is used to generate a common key between three legitimate nodes n_{A}, n_{B} and n_{C}. Figure 5.4 illustrates its operational process.

The three main stages of key generation in the group case are the same as those

of the pairwise key generation case presented in the chapter 4, namely, i) randomness sharing stage, ii) the key reconciliation stage and the iii) privacy amplification stage. Similar to the situation in the pairwise key generation, probes need to be transmitted between legitimate nodes so that they can determine what losses were experienced in each of the channels of concern at given sampling times and use those sets of measurements to generate a correlated sequence of values which they can then use to generate a common key. Unlike the pairwise PLSKG, in GPLKSG has the option of having some of the legitimate nodes being more active than the others.

We will first look at group key generation in the scenario where we have three legitimate nodes,n_{A}, n_{B} and n_{C}, which wish to generate a common cryptographic key in the presence of an adversary n_{E} who is a distance of least d away from all the legitimate nodes. In the first instance, we will consider the nodes in a straight-line topology where n_{C} is equidistant from each of n_{A} and n_{B}. Due to the legitimate nodes being on the straight line we naturally have the distance between n_{A} and n_{B} (d_{AB}) being twice the distance between n_{A} and n_{C}. A figure showing the topology under consideration can be seen in figure 5.2.

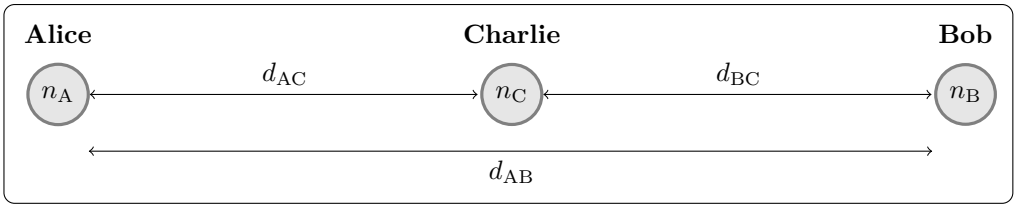


Figure 5.2: Three legitimate nodes ($n_{\mathrm{A}}, n_{\mathrm{B}}$ and n_{C}) in a straight line topology

If n_{A} broadcasts a probe at power level P_{T}, node n_{C} will receive the probe at power level $R_{\mathrm{m,AC}}$ where $R_{\mathrm{m,AC}} \approx P_{\mathrm{T}} - L_{\mathrm{f}}(p_{\mathrm{AC}})$ and n_{B} will receive the probe at power level $R_{\mathrm{m,AB}}$ where $R_{\mathrm{m,AB}} \approx P_{\mathrm{T}} - L_{\mathrm{f}}(p_{\mathrm{AB}})$. The quantity of $L_{\mathrm{f}}(p_{\mathrm{ij}})$ is the power loss incurred over the transmission channel on path p_{ij} from node n_{i} to node n_{j}. Note that $p_{\mathrm{ij}} = p_{\mathrm{ji}}$. Similarly, if n_{C} transmits a probe at power level P_{T}, n_{A} will receive the probe at power level $R_{\mathrm{m,CA}}$ where $R_{\mathrm{m,CA}} \approx P_{\mathrm{T}} - L_{\mathrm{f}}(p_{\mathrm{AC}})$ and n_{B} will receive the probe at power level $R_{\mathrm{m,CB}}$ where $R_{\mathrm{m,CB}} \approx P_{\mathrm{T}} - L_{\mathrm{f}}(p_{\mathrm{CB}})$.

If n_{A} and n_{B} broadcast probes, then it is clear that n_{A} and n_{B} can compute a

common key but it is not immediately clear that third node n_{C} can also join the key generation process to obtain a unique shared key established among n_{A}, n_{B} and n_{C}.

If n_{C} wants to generate a common key with n_{A} and n_{B} primarily using information obtained from the probes it has received from n_{A} and n_{B}, then it (together with n_{A} and n_{B}) need to perform necessary computations on its measured values in order to ensure that the resulting values it obtains are highly correlated to the measured values at n_{A} and n_{B}. This has implications on the restrictions put on the location of adversary n_{E} because if n_{E} is collocated with respect to any of the three legitimate nodes, then n_{E} could also generate the common key.

We will refer to n_{A} and n_{B} as active nodes as they participate fully in all the stages of the group key generation process, and refer to n_{C} as a partially passive node as it will participate fully only in particular stages of the key generation process.

5.2.1 Wireless Channel Model

Consider two nodes, n_A and n_B, that wish to communicate wirelessly. The waveform radiated from n_A to n_B traverses through space toward n_B along some set of signal paths $\{p_l\}$. En-route to n_B, various wireless phenomena distort the waveform. It can be shown that the channel is approximately symmetrical [19]. This is due in part to the fact that the signal traversal paths from n_A to n_B are the same as the traversal paths from n_B to n_A.

Let the transmitted waveform be $s(t)$ and the received waveform be $r(t)$. The average power of the radio waveform at the point of leaving the transmitter, expressed as $\mathbb{E}[s^2]$, will differ from the average power of the received waveform, denoted as $\mathbb{E}[r^2]$. This difference is due to a number of dispersive losses such as spatial dispersion which cause the radiated waveform to disperse in space, resulting in the waveform traversing from n_A to n_B to lose power at a rate proportional to the node separation distance, raised to some environmental dependent exponent. This is the linear path loss, L_{PL} [19].

In addition, there are also time dispersive losses, which are predominately caused by the fact that the wave propagates along different paths whilst traversing from n_A to n_B. These different paths have different lengths, which cause different copies of the waveform to arrive at different times - a situation commonly referred to as time dispersion. This then leads to a Gaussian distributed power loss [12].

The final main class of losses are frequency dispersive losses which are brought about by there being some relative velocity between communicating nodes. Due to Doppler shifts, the receiver perceives a signal, which has a frequency of f_{TX}, as actually having a frequency of f_{RX}, where $f_{\mathrm{RX}} = f_{\mathrm{TX}} + \frac{\Delta v}{\lambda} f_{\mathrm{TX}}$, Δv is the relative velocity, and λ is the wavelength. Such losses usually follow a Rayleigh distribution [12].

Let $P_{\mathrm{TX}} = \mathbb{E}[s^2]$ denote the transmit power, $P_{\mathrm{RX}} = \mathbb{E}[r^2]$ the receive power, L the total loss from the transmitter to the receiver, L_{SF} the slow fading time dispersive losses, L_{FF} frequency dispersive losses, and σ^2 the variance of the Gaussian distributed loss component. We then have the relations below:

$$\begin{aligned} P_{\mathrm{RX}} &= P_{\mathrm{TX}} - L \\ &\approx P_{\mathrm{TX}} - (L_{\mathrm{PL}} + L_{\mathrm{SF}} + L_{\mathrm{FF}}) \\ &\approx P_{\mathrm{TX}} + Z \end{aligned}$$

Where $Z = -(L_{\mathrm{PL}} + L_{\mathrm{SF}} + L_{\mathrm{FF}})$. If the nodes are stationary or near stationary, then we have $L_{\mathrm{FF}} \approx 0$, leading to the simplification below:

$$P_{\mathrm{RX}} \cong P_{\mathrm{TX}} - L_{\mathrm{PL}} - L_{\mathrm{SF}} \tag{5.3}$$

P_{TX} and L_{PL} are constants. L_{SF} follows a Gaussian distribution of mean 0 and variance σ^2. P_{RX} thus gfollows a Gaussian distribution with mean $(P_{\mathrm{TX}} - L_{\mathrm{PL}})$ and variance σ^2 [12].

Due to the fact that the power loss channel like the channel impulse response channel is approximately symmetrical, the power loss Z_{AB} from n_A to n_B is approximately be equal to the power loss Z_{BA} from n_B to n_A. Also there are some factors that introduce asymmetry to these channels and a detailed review of these can be seen in [59]. The received power profiles, X_{AB} and X_{BA}, can thus be modelled as being

of the following form:

$$\begin{aligned} X_{\mathrm{BA}} &\dot{\sim} P_{\mathrm{TX}} + Z_{\mathrm{BA}} \\ &\dot{\sim} P_{\mathrm{TX}} + \mathcal{N}(-L_{\mathrm{PL}}, \sigma^2_{\mathrm{BA}}) + N_1 \\ &\dot{\sim} \mathcal{N}(P_{\mathrm{TX}} - L_{\mathrm{PL}}, \sigma^2_{\mathrm{BA}}) + N_1 \\ &\dot{\sim} \mathcal{N}(P_{\mathrm{TX}} - L_{\mathrm{PL}}, \sigma^2_{\mathrm{BA}}) + \mathcal{N}(0, \sigma^2_N) \end{aligned} \tag{5.4}$$

and

$$\begin{aligned} X_{\mathrm{AB}} &\dot{\sim} P_{\mathrm{TX}} + Z_{\mathrm{AB}} \\ &\dot{\sim}\dot{\sim} P_{\mathrm{TX}} + \mathcal{N}(-L_{\mathrm{PL}}, \sigma^2_{\mathrm{AB}}) + N_2 \\ &\dot{\sim} \mathcal{N}(P_{\mathrm{TX}} - L_{\mathrm{PL}}, \sigma^2_{\mathrm{AB}}) + N_2 \\ &\dot{\sim} \mathcal{N}(P_{\mathrm{TX}} - L_{\mathrm{PL}}, \sigma^2_{\mathrm{AB}}) + \mathcal{N}(0, \sigma^2_N) \end{aligned} \tag{5.5}$$

Here, N_1 and N_2 are independent additive Gaussian distributed noises. $N(\mu, \sigma^2)$ is a Gaussian distributed random variable with a mean of μ and a variance of σ^2. σ^2_{AB} and σ^2_{BA} are the variances of the Gaussian distributed slow fading losses experienced on the channel from n_B to n_A and n_A to n_B respectively.

Let $I(X_{\mathrm{AB}}; X_{\mathrm{BA}})$ denote the mutual information between X_{AB} and X_{BA}. It can be shown [11] that if X_{AB} and X_{BA} are Gaussian distributed, then we have:

$$I(X_{\mathrm{AB}}; X_{\mathrm{BA}}) = -\frac{1}{2}\log(1 - \rho^2_{\mathrm{AB}}) \tag{5.6}$$

where ρ_{AB} is the Pearson correlation coefficient. Let $\mathrm{x}^n_{\mathrm{AB}}$ and $\mathrm{x}^n_{\mathrm{BA}}$ be long vectors formed by sampling X_{AB} and X_{BA} n times, where n tends to ∞. Then the relation in equation 5.7 holds [11]:

$$\rho^2_{\mathrm{AB}} \approx \mathrm{corr}(\mathrm{x}^n_{\mathrm{AB}}, \mathrm{x}^n_{\mathrm{BA}}) \tag{5.7}$$

where

$$\rho_{\mathrm{xy}} = \frac{E[xy] - E[x]E[y]}{\sqrt{E[x^2] - [E[x]]^2}\sqrt{E[y^2] - [E[y]]^2}} \tag{5.8}$$

5.2.2 Adversarial Model Physical Layer Key Generation Problem Statement

Adversary n_E receiving a set of waveforms x_{EA}^n and x_{EB}^n from n_A and n_B respectively will have access readings that follow the distributions of X_{EA} and X_{EB} respectively. Its mutual information relative to the main channel between n_A and n_B takes the higher value of $I(X_{\mathrm{AB}}; X_{\mathrm{EA}})$ and $I(X_{\mathrm{AB}}; X_{\mathrm{EB}})$.

The diagrams in figures 5.1a and 5.1b show the various channel models that are applicable to the adversarial set-up.

Figure 5.1a shows the general channel model when a node n_A is transmitting and nodes n_B, n_C and n_E are receiving. n_A transmits packets at power level P_{TX}. The packets then traverse through the wireless environment and arrive at the receiving nodes n_C, n_B and n_E at receive power levels $P_{\mathrm{TX}} - L_{\mathrm{CA}}$, $P_{\mathrm{TX}} - L_{\mathrm{BA}}$ and $P_{\mathrm{TX}} - L_{\mathrm{EA}}$, respectively. These receive powers follow random variables denoted as X_{CA}, X_{BA} and X_{EA} respectively. L_{ij} is the power loss over the channel from node n_j to node n_i. These channel losses are random variables denoted as Z_{CA}, Z_{BA} and Z_{EA} respectively.

The objective of key generation is to generate a key between the legitimate nodes with a high entropy relative to n_E. This entropy of the generated key can be shown to be maximum when the observations at the adversary are completely independent and completely uncorrelated to the legitimate observations. It can also be shown that the entropy of the generated key will be zero when the adversarial channel is completely correlated with the legitimate channels.

In the case where the adversarial channel's entropy is greater than the legitimate channel's entropy, it can be shown that the entropy of the generated key is minimised when the general model in figure 5.1a can be represented in the degraded form shown in figure 5.1b [8]. In the figure Z_1, Z_2 and Z_3 in the figure are independent random variables.

It can be shown that if the secrecy capacity of a scheme is denoted C_s, then C_s is at least equal to C_s^D, where C_s^D denotes the secrecy capacity of a communication scheme when the adversarial node's observations are degraded with respect to the main (i.e legitimate) channel observations as shown in figure 5.1b [8]. The secrecy

capacity can thus be bounded as follows:

$$C_s \geq C_s^D = C_m - C_{\mathrm{E}} \tag{5.9}$$

Where C_m is the capacity of the main channel and C_E is the capacity of the adversarial degraded channel. The degraded channel model allows us to conservatively analyse the secrecy capacity of a scheme by using a lower bound for the maximum achievable secrecy [8]. The degraded channel model is only applicable in cases where $H(Z_{\mathrm{AE}}) > H(Z_{\mathrm{BE}}) > H(Z_{\mathrm{CE}})$, where $H(X)$ is the entropy of the random variable X. This property can be satisfied in practice by restricting the adversary to be some distance away from the legitimate nodes. In this thesis we will be concerned with key generation in the presence of an adversary n_E making observations via a degraded channel.

We now state the physical layer key generation problem for group key generation with three nodes: n_A, n_B and n_C. Given sets of sampled sequences with length n, drawn from the correlated distributions observable at the three legitimate nodes, define a key generating function f such that:

- A common key $K_{\mathrm{ABC}} = f(\mathrm{x}_{\mathrm{AB}}^n, \mathrm{x}_{\mathrm{AC}}^n) = f(\mathrm{x}_{\mathrm{BA}}^n, \mathrm{x}_{\mathrm{BC}}^n) = f(\mathrm{x}_{\mathrm{CA}}^n, \mathrm{x}_{\mathrm{CB}}^n)$ can be generated.
- There exists no function g such that $K_{\mathrm{ABC}} = g(\mathrm{x}_{\mathrm{EA}}^n, \mathrm{x}_{\mathrm{EB}}^n, \mathrm{x}_{\mathrm{EC}}^n)$. Where $\{\mathrm{x}_{\mathrm{EA}}^n, \mathrm{x}_{\mathrm{EB}}^n, \mathrm{x}_{\mathrm{EC}}^n\}$ are observations available to an adversary n_E, which are correlated to the main observations to some degree $\rho_E = \max\{\rho_{\mathrm{EA}}, \rho_{\mathrm{EB}}, \rho_{\mathrm{EC}}\}$. Without losing generality we will take $\rho_E = \rho_{\mathrm{EA}}$.
- Let g be the best function for estimating K_{AB} with $\mathbb{P}(g(\mathrm{x}_{\mathrm{EA}}^n, \mathrm{x}_{\mathrm{EB}}^n, \mathrm{x}_{\mathrm{EC}}^n) = K_{\mathrm{ABC}}) \approx 2^{-H(K_{\mathrm{ABC}})}$.

For simplicity, only three legitimate nodes are considered in this paper, but the scheme can be extended to deal with a much larger number of nodes, which will be discussed in section 5.4.3.

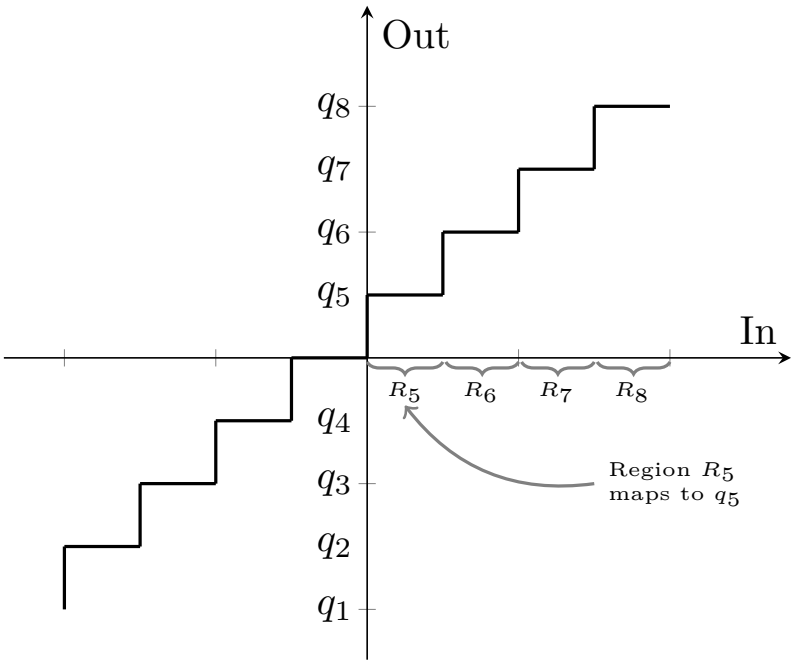
5.3 Quantisation

Quantisation involves sampling some continuous time signal and rounding the samples off to the nearest number in some predetermined discrete set of numbers. The set of all valid output samples is the valid set of quantisation points. The set of valid quantisation points is $\{p_i\}$, which is also denoted as a quantisation vector Q (i.e. $Q = \{q_i\}$).

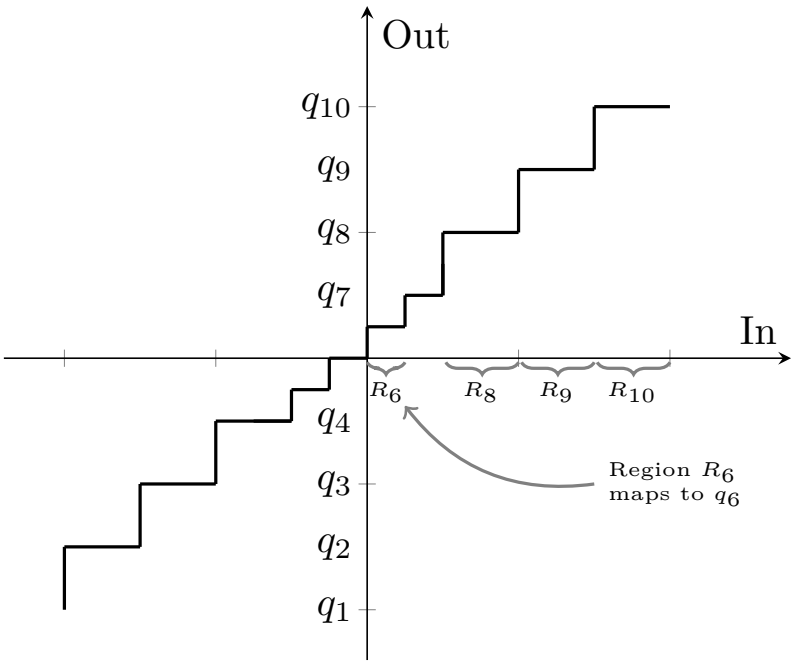
If the separation between the quantisation points is fixed and uniform, then the quantisation scheme used is denoted as Q_d^u where d is the separation between any two adjacent quantisation points. In a uniform quantisation scheme, d is also the average separation between the set of valid quantisation points. The range on which the quantisation points all fall is $[-R, +R)$, which can be explicit when needed by denoting Q_d^u as $Q_{d,[-R,+R)}^u$. In a uniform scheme Q_d^u the quantisation points, $\{q_i\}$ are defined by $q_i = -R + d/2 + (i-1)d$.

Let $R_{m,i}$ be some set of values measured by a node i. The quantisation of $R_{m,i}$ using the quantisation vector Q_d^u is denoted as $Q_d^u(R_{m,i})$ and evaluates to some set of quantised values. The concatenation of these quantisation values yields a weak key $K_{i,u,d}$. In the case where the quantisation vector is a standard quantisation vector which is Q_1^u, this key can just be denoted K_i.

A non-uniform quantisation vector consists of valid quantisation points which are not uniformly spaced. A non-uniform quantisation scheme with quantisation points $\{q_i\}$ is denoted as $Q_{\{q_i\}}^n$. The range on which the quantisation points all fall is $[-R, +R)$, which can be explicit when needed by denoting $Q_{\{q_i\}}^n$ as $Q_{\{q_i\},[-R,+R)}^n$. Figures 5.3a and 5.3b show a set of different quantisation vectors. The quantisation levels in figure 5.3a are all uniform and those in 5.3b are nonuniform.



(a) Uniform Quantiser



(b) Non-Uniform Quantiser

Figure 5.3: Main Types Of Quantisers

A quantisation vector can be expressed as the union on two quantisation vectors. If a quantisation vector is expressed in this manner then the quantisation vector has quantisation points from both quantisation vectors. For instance, the quantisation vector which is the union of $Q^u_{1,[-R,+R)}$ and $Q^u_{1,[-R,-R+1)}$ is denoted $Q^u_{1,[-R,+R)} \bigcup Q^u_{1,[-R,-R+1)}$.

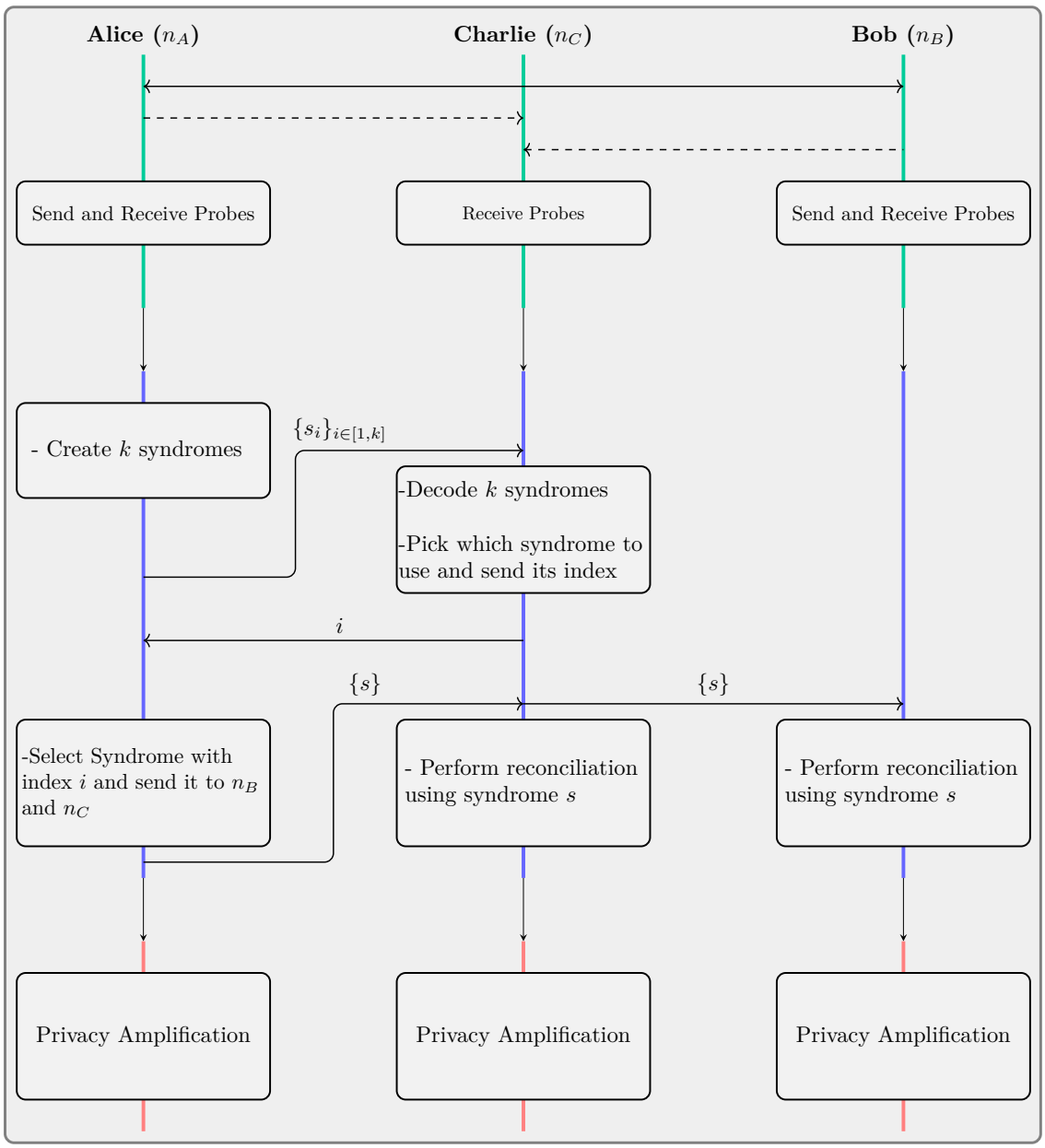


Figure 5.4: Overview of Group PLSKG Scheme

5.4 Novel Physical Layer Group Key Generation Scheme

5.4.1 Randomness Sharing

The randomness sharing stage of the proposed scheme involves participants sending probes over a set of pre-defined frequencies. In the pairwise setting, the participating nodes n_A and n_B trade N_i probes at each centre frequency. The frequency hopping sequence is determined by some schedule S, which has N_j distinct centre frequencies per round. The average RSS per centre frequency is then taken as the measured RSS value over the frequency $f_j \in S$. The set of measured RSS values

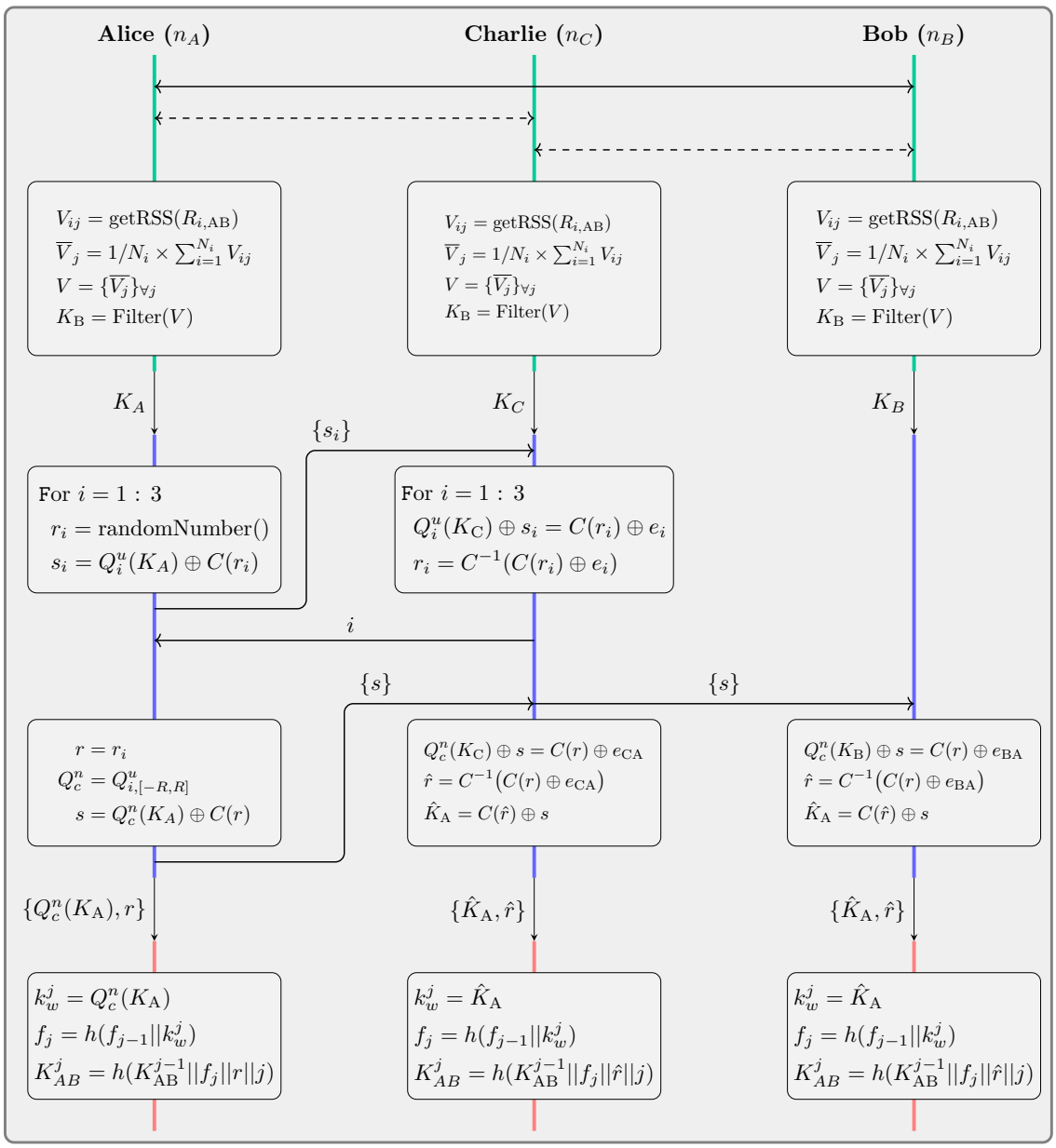


Figure 5.5: Detailed Overview of Group PLSKG Scheme

are then concatenated to form a measured RSS vector of size N_j.

The values are then passed through a zero order filter in order to gain a zero mean vector. The resulting measured vector and the mean of the vector are then passed to the second stage which is the key reconciliation stage.

In the group setting a third legitimate node n_C is also present. In our proposed scheme, n_C joins the scheme by also measuring all probes sent over its channel by the other legitimate nodes, but it sends only one message to other legitimate nodes per frequency to facilitate frequency hopping synchronisation and to confirm that it has successfully completed a round of RSS measurement.

Nodes n_A and n_B are active nodes as they actively send probes fully in the randomness sharing stage as in the pairwise case. n_C is partially passive i.e. it does not participate fully in the randomness sharing stage. n_C gets its values by estimating the losses by observing the probes received from n_A and n_B. Due to the scheme having one of the nodes being partially passive, the key generation rate is lower, which puts additional restrictions on the adversary as its channel is degraded with respect to n_A, n_B and importantly n_C. The topology of the three legitimate nodes under consideration is a straight-line topology with n_C equidistant from n_A and n_B. This straight-line topology can be seen in figure 5.2.

Note that the scheme proposed in this paper can be applied to a non straight-line topology with nodes n_A, n_B and n_C. To justify this applicability, let n_A and n_B once again be the active nodes in our GPLSKG scheme and let n_C be the partially passive legitimate node. Further to this, let $d_{ij} = d_{ji}$ be the distance between n_i and n_j.

The active nodes have access to RSS measurements $\mathrm{x}^n_{\mathrm{AB}}$ and $\mathrm{x}^n_{\mathrm{BA}}$ which can be thought of as having been drawn from probability distributions X_{AB} and X_{BA} respectively. Node n_C has access to measurements $\mathrm{x}^n_{\mathrm{CA}}$ and $\mathrm{x}^n_{\mathrm{CB}}$ which can be thought of as having been drawn from probability distributions X_{CA} and X_{CB}. In order to generate a common key, all three nodes use their measurements to reconcile a common key.

Node n_C uses a function $f_C(\cdot)$ to estimate the measurements at n_A ($\hat{\mathrm{x}}^n_{\mathrm{AB,C}}$) and node n_B uses a function $f_B(\cdot)$ to estimate the measurements at n_A ($\hat{\mathrm{x}}^n_{\mathrm{AB,B}}$). The

mutual information between n_A, n_B and n_C will then be at least proportional to $\rho = \min\{\rho_{\mathrm{AB,C}}, \rho_{\mathrm{AB,B}}\}$ where ρ is the Pearson coefficient and the relation between ρ and the mutual information will be lower bounded by the relation in equation 5.6. Here, $\rho_{\mathrm{AB,C}} = \mathrm{corr}(\hat{\mathrm{x}}^n_{\mathrm{AB,C}}, \mathrm{x}^n_{\mathrm{AB}})$ and $\rho_{\mathrm{AB,B}} = \mathrm{corr}(\hat{\mathrm{x}}^n_{\mathrm{AB,B}}, \mathrm{x}^n_{\mathrm{AB}})$.

The function f_{C} is the best-known estimator. In other words, f_{C} has to ideally be chosen to satisfy:

$$\left\{ f_{\mathrm{C}}(\mathrm{x}^n_{\mathrm{CA}}, \mathrm{x}^n_{\mathrm{CB}}) : \min_f |f(\mathrm{x}^n_{\mathrm{CA}}, \mathrm{x}^n_{\mathrm{CB}}) - \mathrm{x}^n_{\mathrm{AB}}| \right\} \tag{5.10}$$

The function f_{C} is thus the best known function for estimating the measured values at n_A from n_C. The best known general function for estimating the loss at distance d_j using the measured path loss at distance d_i (where $d_j > d_i$) is ([12]):

$$\hat{L}_{\mathrm{PL}} = L(d_j) = L(d_i) + 20\log_{10}(d_j - d_i) \tag{5.11}$$

If the nodes are not in a straight-line topology, equation 5.11 can thus be used to estimate the RSS values measured at the active nodes using the passive nodes measured values. The estimator in equation 5.11 uses distance information and so can only be used in cases where all nodes are aware of the network topology beforehand.

The equation outlined in equation 5.11 estimates the path loss over the channel and has an error which is known to vary depending on a wide variety of factors such as the specific deployment environment and also known to increase with increasing values of d_j. The formulation of an exact, closed form expression for this error is still an open problem. The received power can be estimated as:

$$\hat{P}_{\mathrm{RX}} = P_{\mathrm{TX}} - \hat{L}_{\mathrm{PL}} \tag{5.12}$$

The difference between estimated power $\hat{P}_{\mathrm{RX}}$ and actual power P_{RX} is:

$$\begin{aligned} |P_{\mathrm{RX}} - \hat{P}_{\mathrm{RX}}| &\approxeq |(P_{\mathrm{TX}} - L_{\mathrm{PL}} - L_{\mathrm{SF}} - L_{\mathrm{FF}}) - \hat{P}_{\mathrm{RX}}| \\ &\approxeq |(P_{\mathrm{TX}} - L) - (P_{\mathrm{TX}} - \hat{L}_{\mathrm{PL}})| \\ &= |\hat{L}_{\mathrm{PL}} - L| \end{aligned} \tag{5.13}$$

Here, $L = L_{\mathrm{PL}} + L_{\mathrm{SF}} + L_{\mathrm{FF}}$. Equation 5.13 gives the error obtained when a path loss model is used to estimate the received power. Although no exact closed form expression exists for this error, different bounds have been investigated by different works. The work in [60] empirically investigates the bounds on errors between actual losses and losses predicted by loss models. The paper looks at a number of loss models in different deployment environments and provides estimates of the mean error of estimation in a set of sample environments.

If $L_{\mathrm{PL}} \approx \hat{L}_{\mathrm{PL}}$, then the equation in 5.13 can be bounded as follows:

$$\begin{aligned} |P_{\mathrm{RX}} - \hat{P}_{\mathrm{RX}}| &\cong |\hat{L}_{\mathrm{PL}} - L| \\ &\approx |L_{\mathrm{SF}} + L_{\mathrm{FF}}| \end{aligned} \tag{5.14}$$

As mentioned earlier, the frequency selective losses (L_{SF}) are approximately zero mean Gaussian distributed whilst the time selective losses (L_{FF}) are either Rician or Rayleigh distributed. In the case where nodes are approximately stationary we have $L_{\mathrm{FF}} \approx 0$. The average absolute deviation of the error is then approximately equal to the mean absolute deviation (or mean absolute error) from a normal distribution which can be shown to be equal to:

$$\mathbb{E}\left[|L_{\mathrm{SF}}|\right] = \sqrt{\frac{2\sigma^2}{\pi}} \tag{5.15}$$

Here, σ^2 is the variance of the distribution followed by the random variable L_{SF}. Further to this, note that the entropy of a Gaussian distribution computes to [11]:

$$H(X) = \frac{1}{2}\log_2 2\pi e\sigma^2 \tag{5.16}$$

Where X is a Gaussian distribution with variance σ^2 and e is a mathematical constant that is approximately equal to 2.71828. From equation 5.16 we can evaluate σ^2 to be:

$$\sigma^2 = \frac{2^{2H(X)}}{2\pi e} \tag{5.17}$$

which implies that

$$\begin{aligned}\mathbb{E}\left[|X|\right] &= \sqrt{\frac{2\sigma^2}{\pi}} \\ &= \sqrt{\frac{2^{2H(X)}}{\pi^2 e}}\end{aligned} \tag{5.18}$$

If we restrict ourselves to adversarial channels that are degraded relative to the legitimate channels then by definition we have restricted ourselves to cases where $H(Z_E) > \max\{H(Z_A), H(Z_B), H(Z_C)\}$. Thus, in equation 5.18, we can see that the mean absolute error will be greater for the adversary if the adversary has a higher entropy (i.e. noisier) channel.

If the difference in distance between n_A and n_C (d_{AC}) is significantly smaller than the distance from n_A to n_B (d_{AB}), then the RSS values at n_C can be corrected by 6dB. This is because the path loss between two nodes is related to the distance separating them according to the approximate relation outlined in equation 5.20 [12] which computes to approximately 6dB if $d_{\text{AB}} = 2d_{\text{AC}}$. This relation is only valid for large separations and thus in the experiments this correction was done when $d_{\text{AB}} \geq 10m$.

$$\begin{aligned}P_L(d_{\text{AB}}) &\approx 20\log_{10}(d_{\text{AB}}) + 10\log_{10}\left(\frac{4\pi f}{c}\right) \\ &= 20\log_{10}(d_{\text{AB}}) + \text{Constant}\end{aligned} \tag{5.19}$$

$$\begin{aligned}P_L(d_{\text{AB}}) - P_L(d_{\text{AC}}) &\approx 20\log_{10}(d_{\text{AB}}) - 20\log_{10}(d_{\text{AC}}) \\ &= 20\log_{10}(d_{\text{AB}}/d_{\text{AC}})\end{aligned} \tag{5.20}$$

Here, f is the frequency of the transmitted wave and c is the speed of light. If the proposed scheme is not used in a straight-line topology, the relation in equation 5.20 can be used to provide an estimate of the RSS at the partially active node. This is possible if the legitimate nodes are all aware of the deployment topology. The adversarial node still needs to have an RSS channel that is degraded with respect to all legitimate nodes in order to ensure that there is some secrecy capacity. Similarly the scheme can be applied to a larger group of more than 3 nodes for group key generation.

5.4.2 Key Reconciliation

The aim of key reconciliation is to reconcile the measured vectors at all legitimate nodes into one common key whilst denying some adversary the opportunity to generate the common key. This is only possible if the legitimate nodes have keys which correlate with each other to a degree higher than the correlation with the adversarial key. In light of this, restrictions have to be put on the adversarial nodes that our scheme is secure against. In practice this translates to location limitations on the adversary i.e. the adversary cannot be collocated relative to the legitimate nodes.

The proposed key reconciliation stage consists of two main steps, i) quantisation and ii) error correction with error correcting codes. Quantisation with uniform quantisation point distances has been proposed in the past for key reconciliation schemes. A key limitation of this approach is the coarse manner in which it reduces key entropy and the difficultly in establishing the maximum acceptable quantisation interval, as detailed in [4].

This thesis proposes the use of a different binary encoding method from the standard 2's complement for the quantiser, which we call uniform binary encoding. We then show how such encoding allows us to have more knowledge about the structure of the binary output from the encoder. This allows an ideal error threshold to be quantitatively evaluated.

Once we have an error threshold, we can then use an error correcting procedure to correct all errors in measured RSS values given that the number of errors is less than the threshold. More importantly, we can ensure that an adversary is unlikely to be able to reconcile a key if their number of errors passes the given threshold. In the following sub-sections, we first consider a pairwise case and then extend it to a group case.

5.4.3 Quantisation and Encoding

5.4.3.1 Quantisation

Consider a standard uniform quantiser used to quantise sequences x_A^n and x_B^n, where every value in each sequence is some RSS value in the range $[-R, R]$, where R is a real number.

The objective of quantisation is to quantise the measured physical layer values and then formulate a binary sequence. This is done at both legitimate ends (i.e. at nodes n_A and n_B). In addition we can assume that the adversary can also perform the same operation if it wishes. A quantiser takes some real value as input and rounds it to the nearest valid quantisation point. The index of the quantisation point is the binary output from the quantiser. Here, the output binary sequence is typically the 2's complement binary representation of the quantisation point's index.

The result of this process is that the output binary sequence will be of length $n_o = n \log_2 B$, where n is the input sequence length and $\log_2 B$ is the bit depth (i.e. the number of binary bits per sample), where the bit depth of the quantiser with B levels is $\log_2 B$.

The legitimate nodes now have to reconcile the key without the adversary being able to reconcile their key. However, given that the correlation between the legitimate nodes is ρ_{AB} and the correlation between the adversary's key and any of the legitimate keys is at most ρ_{EA}, where $\rho_{\mathrm{EA}} < \rho_{\mathrm{AB}}$, if the binary encoding used is the 2's complement or even Gray coding, it is not possible to usefully bound the bit error between the legitimate and adversarial sequences.

This is largely due to the fact that decimal samples which vary by large differences can have very small bit differences in their bit representations. This is true for both 2's complement and Gray coding. Gray coding reduces the problem slightly, as it ensures that decimal numbers adjacent to each other have exactly one bit difference but it does not guarantee large bit differences for large deviations. Tables 5.1, 5.2 and 5.3 show index encoding using 2's complement, Gray coding and the proposed uniform encoding scheme respectively. Note from the tables that for the 2's complement and Gray coding, a large difference in decimal numbers does not necessarily

translate to a high bit disagreement between their binary representations, a key issue which makes error threshold bounding difficult.

Index	2's Complement
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Table 5.1: Standard Binary Encoding

Index	Gray Code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100

Table 5.2: Gray Encoding

Index	Uniform Distance
0	0000000
1	0000001
2	0000011
3	0000111
4	0001111
5	0011111
6	0111111
7	1111111

Table 5.3: Uniform Distance Encoding

A classical approach to tackling the issue of low successful key generation rates (SKGR) in quantisation only key generation schemes is to reduce the fidelity of the quantiser by iteratively reducing the number of quantisation points in the uniform quantiser up until the point where n_A and n_B can reconcile a common key.

If a quantiser, $Q_L(x)$, is uniform with spacing between L quantisation levels, then there are $2R/L$ quantisation points, so the quantiser has a bit depth of $\log_2(2R/L)$.

Doubling the spacing to form another quantiser $Q_{2L}(x)$ increases the SKGR, but doing so raises a number of issues. The first one is the loss of entropy of the generated key - an issue discussed in [4]. This results in more RSS measurements being required.

A much more serious issue is that there is no clear way of knowing at what point the separation between the quantisation points is so large that there is no secrecy capacity left (i.e. the point at which the adversary would be able to theoretically reconcile the key).

In order to tackle these issues, this paper proposes a uniform binary encoder ($\varphi(\cdot)$) which ensures that given decimal numbers a and b, equation 5.21 holds:

$$H_D\Big(\varphi(a), \varphi(b)\Big) = |a - b| \tag{5.21}$$

Here, $H_D(i, j)$ is the hamming distance between binary numbers i and j. If we use the uniform binary encoding, then we know that:

$$\begin{aligned} |\mathrm{x}_A^n - \mathrm{x}_B^n| &= \sum_{i=1}^{n} |\mathrm{x}_A^i - \mathrm{x}_B^i| \\ &= \sum_{i=1}^{n} H_D\Big(\varphi(\mathrm{x}_A^i), \varphi(\mathrm{x}_B^i)\Big) \\ &= e_{\mathrm{AB}} \end{aligned} \tag{5.22}$$

Thus, we can find the expected value of $\mathrm{x}_A^n - \mathrm{x}_B^n$ and then bound the total expected number of bit errors between the quantised binary sequences x_A^n and x_B^n (e_{AB}). Similarly we can bound the total expected number of bit errors between the quantised binary sequences at the adversary and n_A, x_E^n and x_A^n ($e_{\mathrm{E}} = \min\{e_{\mathrm{AE}}, e_{\mathrm{BE}}\}$). We can then choose an error correcting code with a threshold e_{T} where $e_{\mathrm{T}} \in [e_{\mathrm{AB}}, e_{\mathrm{E}})$.

We now proceed to defining the difference distribution X_D which is a random variable $X_D = X_{AB} - X_{BA}$. We first evaluate the expected value of $|X_D|$ and then relate it to expected bit errors between the quantised binary sequences x_A^n and x_B^n.

Given two normal random variables X and Y where

$$X \sim \mathcal{N}(\mu_X, \sigma_X^2)$$
$$Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$$

with a Pearson coorelation ρ_{AB}. Here, μ_X and μ_Y are the means of the random variables X and Y respectively. σ_X^2 and σ_Y^2 are the variances of X and Y. The random variable $Z = X - Y$ is:

$$Z \sim \mathcal{N}(\mu_X - \mu_Y, \sigma_X^2 + \sigma_Y^2 - 2\rho_{XY}\sigma_X\sigma_Y) \tag{5.23}$$

From equation 5.23, we thus know

$$X_D \sim \mathcal{N}(\mu_D, \sigma_D^2) \tag{5.24}$$

where $\mu_D = \mu_{\mathrm{AB}} - \mu_{\mathrm{BA}}$ and $\sigma_D^2 = \sigma_{\mathrm{AB}}^2 + \sigma_{BA}^2 - 2\rho_{\mathrm{AB}}\sigma_{\mathrm{AB}}\sigma_{\mathrm{BA}}$. The distribution of a random variable $|X|$ is the folded normal distribution if X is normal. The mean of the folded normal $|X|$ is $\sigma_X\sqrt{2/\pi}$, where σ_X is the standard deviation of X. Applying this to X_D yields the mean of the distribution $|X_D|$ as:

$$\mathbb{E}\{|X_D|\} = \sigma_D\sqrt{2/\pi}$$

By the weak law of large numbers, we have $\mathbb{E}\{|X_D|\} \to \mu_{|D|} = \sigma_D\sqrt{2/\pi}$. This is enough to show that the expected value of $|X_D|$ tends to be $\mu_{|D|}$ as n tends towards infinity. e_{AB} can then be estimated for large n as:

$$\begin{aligned} e_{\mathrm{AB}} &\approx n\mathbb{E}\{|X_D|\} \\ &= n\sigma_D\sqrt{2/\pi} \\ &= \sqrt{(\sigma_{\mathrm{AB}}^2 + \sigma_{\mathrm{BA}}^2 - 2\rho_{\mathrm{AB}}\sigma_{\mathrm{AB}}\sigma_{\mathrm{BA}})} \times n\sqrt{2/\pi} \end{aligned} \tag{5.25}$$

Similarly we expect the adversary to have e_{E} errors where e_{E} is:

$$e_{\mathrm{EA}} \approx \sqrt{(\sigma_{\mathrm{EA}}^2 + \sigma_{\mathrm{AE}}^2 - 2\rho_{EA}\sigma_{\mathrm{EA}}\sigma_{\mathrm{AE}})} \times n\sqrt{2/\pi} \tag{5.26}$$

$$e_{\mathrm{EB}} \approx \sqrt{(\sigma_{\mathrm{EB}}^2 + \sigma_{\mathrm{BE}}^2 - 2\rho_{\mathrm{EB}}\sigma_{\mathrm{EB}}\sigma_{\mathrm{EB}})} \times n\sqrt{2/\pi} \tag{5.27}$$

$$e_{\mathrm{E}} = \min\{e_{\mathrm{EA}}, e_{\mathrm{EB}}\} \tag{5.28}$$

The error threshold used in the key reconciliation stage can therefore be set to a value e_{T} where $e_{\mathrm{T}} \in [e_{\mathrm{AB}}, e_{\mathrm{E}})$. Where the values of e_{AB} and e_{E} are given by equations 5.25 and 5.28 respectively.

If two different numbers, a and b, are quantised by a quantiser with a uniformly spaced quantiser with a spacing of l, using the uniform distance binary encoding yields two binary numbers, a_l and b_l, where $H_D(0, \mathrm{a_l} \oplus \mathrm{b_l}) = e_l$ and $e_l \geq (a-b)/l+1$. A quantiser with spacing l is a function that takes a binary number x_1 and outputs a binary number $x_{1,l}$ where:

$$x_{1,l} = Q_l^u(x_1) = \left\lfloor \frac{x_1}{l} + \frac{1}{2} \right\rfloor \tag{5.29}$$

$$\begin{aligned} \implies Q_l^u(x_1) - Q_l^u(x_2) &= \left\lfloor \frac{x_1}{l} + \frac{1}{2} \right\rfloor - \left\lfloor \frac{x_2}{l} + \frac{1}{2} \right\rfloor \\ &\leq \left(\frac{x_1}{l} + \frac{1}{2} \right) - \left(\frac{x_2}{l} - \frac{3}{2} \right) \\ &= \frac{x_1 - x_2}{l} + 2 \end{aligned} \tag{5.30}$$

$$\begin{aligned} \implies \mathbb{E}\left[(Q_l^u(x_1) - Q_l^u(x_2)) \right] &\leq \mathbb{E}\left[\frac{x_1 - x_2}{l} + 2 \right] \\ &= \frac{1}{l} \mathbb{E}\left[x_1 - x_2 \right] + 2 \end{aligned} \tag{5.31}$$

Further to the above, consider a quantisation regime where half of the range is quantised by a quantiser with spacing l_1 and the other half is quantised by a quantiser with spacing l_2, where $l_1 < l_2$, i.e.

$$Q_d^n(x) = \begin{cases} Q_{l_1}^u(x) & x \in [-R, 0) \\ Q_{l_2}^u(x) & x \in [0, +R] \end{cases} \tag{5.32}$$

$$
\begin{aligned}
&\mathbb{E}\left[|(Q_d^n(x) - Q_d^n(y))|\right] \\
&\quad = \mathbb{P}(x < 0, y < 0)\mathbb{E}(|Q_{l_1}^u(x) - Q_{l_1}^u(y)|) \\
&\quad + \mathbb{P}(x \geq 0, y \geq 0)\mathbb{E}(|Q_{l_2}^u(x) - Q_{l_2}^u(y)|) \\
&\quad + \mathbb{P}(x \geq 0, y < 0)\mathbb{E}(|Q_{l_2}^u(x) - Q_{l_1}^u(y)|) \\
&\quad + \mathbb{P}(x < 0, y \geq 0)\mathbb{E}(|Q_{l_1}^u(x) - Q_{l_2}^u(y)|) \qquad (5.33)
\end{aligned}
$$

If $\mathbb{P}(x \geq 0, y < 0)$ and $\mathbb{P}(x < 0, y \geq 0)$ are small, then

$$
\begin{aligned}
&\mathbb{E}\left[|(Q_d^n(x) - Q_d^n(y))|\right] \\
&\quad \approx \mathbb{P}(x < 0, y < 0)\mathbb{E}(|Q_{l_1}^u(x) - Q_{l_1}^u(y)|) \\
&\quad + \mathbb{P}(x \geq 0, y \geq 0)\mathbb{E}(|Q_{l_2}^u(x) - Q_{l_2}^u(y)|) \\
&\quad \approx 0.5\mathbb{E}(|Q_{l_1}^u(x) - Q_{l_1}^u(y)|) \\
&\quad + 0.5\mathbb{E}(|Q_{l_2}^u(x) - Q_{l_2}^u(y)|) \\
&\quad \leq \frac{1}{2l_1}\mathbb{E}\left[x - y\right] + \frac{1}{2l_2}\mathbb{E}\left[x - y\right] + 2 \qquad (5.34)
\end{aligned}
$$

So if the non-uniform quantiser being used is of the form in equation 5.32, the expected difference can be computed in general using equation 5.33 or approximated by equation 5.34 in cases where the chance that the measured values at legitimate nodes fall in different ranges is deemed to be small.

5.4.3.2 Error Correction Coding

The error coding stage aims to correct up to e_{A} errors in the measured binary sequences. This ideally has to be done using an error correcting code of block size n which will correct all errors in the measured binary sequences with a total number of errors less than e_{A} and more importantly not allow the adversary with e_{E} errors to reconcile them. The analysis of this requirement begins below with the pairwise situation and then extends to the group setting.

The error threshold to be set for the scheme, e_{T}, thus has to be within the integer set $\{[e_{\mathrm{A}}, e_{\mathrm{E}})\}$. Note that if the error threshold is e_{T} then the entropy of the reconciled key is at most $nH((e_{\mathrm{E}} - e_{\mathrm{T}})/n)$, a point which can be proved. Note that the adversary's sequence has e_{E} errors relative to any of the legitimate sequences, so

there exists some n bit sequence of hamming weight $e_{\mathrm{E}} - e_{\mathrm{T}}$ that can be XOR'ed ($\oplus$) with the adversarial sequence to yield a sequence with a bit disagreement of e_{T} relative to a legitimate sequence. That sequence has an entropy of $nH((e_{\mathrm{E}} - e_{\mathrm{T}})/n)$ where $H(\cdot)$ is the binary entropy function.

Once a suitable error correcting code has been identified, the error correction can proceed as detailed in [4]. The process involves node n_A generating a random number r, padding it with n_A's binary sequence x_A to form a new binary sequence called a syndrome s, and sending s to node n_B. n_B can then reconcile x_A using its sequence x_B by following the processing sequence below:

$$
\begin{aligned}
& n_A : \\
& \quad s = C(r) \oplus \mathrm{x}_A \\
& n_B : \\
& \quad \hat{r} = \mathrm{Decode}(s \oplus \mathrm{x}_B) \\
& \qquad = \mathrm{Decode}(C(r) \oplus \mathrm{x}_A \oplus \mathrm{x}_B) \\
& \qquad = \mathrm{Decode}(C(r) \oplus e) \\
& \qquad = r \quad (\text{if } H_D(e, \mathbf{0}) \leq e_T) \\
\implies & \hat{x}_A = C(\hat{r}) \oplus s
\end{aligned}
\tag{5.35}
$$

Once again let n denote the length of binary sequences x_A at n_A and x_B at n_B. In order to reconcile the sequences with e_T errors, we ideally need some error correcting code , $\mathcal{C} = (n, \lambda)$, with a block size n and a rate $R = \lambda/n$, which corrects all errors up to e_T. Such a code is called a perfect code of block size n and error correcting capability e_T. The main problem is that there are only finitely many perfect codes. Table 5.4 lists all known binary perfect codes.

Code	n	e_T
Repetition codes (odd block length)	$2m + 1$	1
Hamming Code	$2^m - 1$	1
Binary Golay Code	23	3

Table 5.4: Binary Perfect Codes ($m \in \mathbb{Z}^+$)

The variable m can take the value of any positive integer (i.e. $m \in \mathbb{Z}^+$). There are of course other error correcting codes such as low-density parity check (LDPC) codes and Turbo codes with large block sizes but these codes do not have a set

error threshold e_T above which we can be confident that they will not correct errors [61]. They are also usually only optimal for large block sizes, which makes their implementation in software very resource-intensive and thus not usually suitable for low resource devices.

A possible way to get around this problem is to divide the binary sequences into blocks of size n_c, create a syndrome for each block and transmit them over the channel to reconcile keys. Note that the number of blocks is $\beta = n/n_c$. The error correction would be accomplished by using a block code of size n_c which can correct up to e_T/β errors. In order for key reconciliation to be successful, every block should have at most e_T/β errors, which is a stricter requirement than just e_T errors over n bits.

We now proceed to estimate the probability of successful key reconciliation using the method of types. A binary sequence of length n with n_1 ones and n_2 zeros is referred to as being a sequence of type class $P = \{n_1, n_2\}$, expressed as $T_n(P)$. In this paper we will denote a type, $T_n(P)$ with $P = \{n_1, n_2\}$, simply as $T_n(n_1)$ to shorten the notation. Note that the number of sequences of type class $T_n(n_1)$ is:

$$|T_n(n_1)| = \binom{n}{n_1} \tag{5.36}$$

We will define an n bit sequence to be of the form $F_{n,n_c}(\{n_1, n_2, ..., n_\beta\})$, if the first block of n_c bits has exactly n_1 ones, the second block of n_c bits has exactly n_2 ones, etc. Another way of seeing F_{n,n_c} is as a sequence of length n which is formed by concatenating β (where $\beta = n/n_c$) sequences of size n_c each, where the first sequence has a hamming weight of n_1, the second one has a hamming weight of n_2, etc. A function $\gamma(i, \{n_1, n_2, ..., n_\beta\})$ returns the number of elements in the set $\{n_1, n_2, ..., n_\beta\}$ which are equal to i. Note that if $i > n_c$ then $\gamma(i, \{n_1, n_2, ..., n_\beta\})$ returns 0 because of $\max\{n_1, n_2, ..., n_\beta\} \leq n_c$.

Let $e_T = e_\mathrm{A} + \alpha$, where sequence $\mathrm{x_A}$ with size n has e_A bits of disagreement relative to another sequence $\mathrm{x_B}$ (i.e. e_A errors). α is an integer combining l non-negative integers that sum up to α. The l-partition of α is defined as the number of combinations of l non-negative integers that sum up to α. Also, let $\mathcal{S}^\alpha$ signify the set of all partitions of α with each having length β.

Assume an error correcting code of block size n_c that can correct up to e_T/β errors is used. Error correction will be possible if $\mathrm{x_A}$ is divided into n_c blocks and there are at most e_T/β errors per block.

If $e_T = e_\mathrm{A} + \alpha$ then:

$$\begin{aligned} e_A &= e_T - \alpha \\ (n_1^{e_A} + ... + n_\beta^{e_A}) &= e_T - (n_1^\alpha + ... + n_\beta^\alpha) \\ &= \underbrace{\left(\frac{e_T}{\beta} + ... + \frac{e_T}{\beta}\right)}_{\beta \text{ times}} - (n_1^\alpha + ... + n_\beta^\alpha) \\ &= \left(\frac{e_T}{\beta} - n_1^\alpha\right) + ... + \left(\frac{e_T}{\beta} - n_\beta^\alpha\right) \end{aligned} \tag{5.37}$$

Where, $(n_1^{e_A} + ... + n_\beta^{e_A}) = e_A$ and $(n_1^\alpha + ... + n_\beta^\alpha) = \alpha$ are partitions of e_A and α in line with the β blocks of $\mathrm{x_A}$ respectively. Equation 5.37 allows us to express the set of numbers in a partition of e_A in the following form:

$$\left\{\left(\frac{e_T}{\beta} - n_1^\alpha\right), ..., \left(\frac{e_T}{\beta} - n_\beta^\alpha\right)\right\} \tag{5.38}$$

which is a relation we will use to simplify computations later on.

For simplicity, we use the shorthand notations $F_{n,n_c} = F_{n,n_c}(\{n_1, n_2, ..., n_\beta\})$, and $\gamma(i) = \gamma(i, \{n_1, n_2, ..., n_\beta\})$. Note that the number of sequences in F_{n,n_c}, $|F_{n,n_c}|$, is:

$$|F_{n,n_c}| = |T_{n_c}(n_1)||T_{n_c}(n_2)|... \times |T_{n_c}(n_\beta)|$$

If every element in $\{n_1, n_2, ..., n_\beta\}$ is different then the total number of permutations of the numbers in the set $\{n_1, n_2, ..., n_\beta\}$ is $\beta!$. If there are $\gamma(1)$ numbers in the set that take on the same value as 1, $\gamma(2)$ numbers in the set that take on the same value as 2 etc., then the total number of possible permutations of the numbers in $\{n_1, n_2, ..., n_\beta\}$ is:

$$\binom{\beta}{\gamma(1), \gamma(2)..., \gamma(n_c)} = \frac{\beta!}{\gamma(1)!\gamma(2)!...\gamma(n_c)!} \tag{5.39}$$

Thus the total number of sequences of form $F_{n,n_c}(\{k_1^{e_A}, k_2^{e_A}, ..., k_\beta^{e_A}\})$ where $\{k_1^{e_A}, k_2^{e_A}, ..., k_\beta^{e_A}\}$

is a permutation of $\{n_1^{e_A}, n_2^{e_A}, ..., n_\beta^{e_A}\}$ is:

$$\begin{aligned} &\binom{\text{Num Of Sequences of form}}{F_{n,n_c}(\{k_1, k_2, ..., k_\beta\})} \\ &= \binom{\text{Num Of Permutations}}{\text{of } \{n_1, n_2, ..., n_\beta\}} \times \binom{\text{Num Of Sequences of}}{\text{form } F_{n,n_c}(\{n_1, n_2, ..., n_\beta\})} \end{aligned} \tag{5.40}$$

The error correcting process involves dividing the n bit sequence into β sub-blocks and then using an error correcting code which can correct up to e_T/β errors on each block. To this end, an n bit sequence is only correctable if each of the individual sub-blocks has less than e_T/β errors. Using this method, it is thus possible for some sequences with errors totalling e_A to be reconcilable. The number of sequences which are reconcilable is:

$$\begin{aligned} N_{\text{CORRECTABLE}} = \\ &= \sum_{\boldsymbol{s}\in\mathcal{S}^\alpha}^{|\mathcal{S}^\alpha|} \binom{\beta}{\gamma(1)\gamma(2)...\gamma(n_c)} \times \\ &\qquad |F_{n,n_c}(\boldsymbol{v} - \boldsymbol{s})| \\ &= \sum_{s\in\mathcal{S}^\alpha}^{|\mathcal{S}^\alpha|} \binom{\beta}{\gamma(1)\gamma(2)...\gamma(n_c)} \times \\ &\qquad \left| F_{n,n_c}\left(\left\{\left(\frac{e_T}{\beta} - n_1^\alpha\right), ..., \left(\frac{e_T}{\beta} - n_\beta^\alpha\right)\right\}\right)\right| \\ &= \sum_{s\in\mathcal{S}^\alpha}^{|\mathcal{S}^\alpha|} \binom{\beta}{\gamma(1)\gamma(2)...\gamma(n_c)} \times \\ &\qquad \left|T_{n_c}\left(\frac{e_T}{\beta} - n_1^\alpha\right)\right| ... \left|T_{n_c}\left(\frac{e_T}{\beta} - n_\beta^\alpha\right)\right| \end{aligned} \tag{5.41}$$

Where $\boldsymbol{v} = \{e_T/\beta\}^\beta = \{e_T/\beta, ..., e_T/\beta\}$ and $\boldsymbol{s} = \{n_1^\alpha, n_2^\alpha, ..., n_\beta^\alpha\}$. $\boldsymbol{s}$ is a partition of α. The probability that a sequence is correctable is:

$$\begin{aligned} \mathbb{P}_C &= \frac{N_{\text{CORRECTABLE}}}{N_{\text{TOTAL}}} \\ &= \frac{N_{\text{CORRECTABLE}}}{|T_n(e_A)|} \\ &= \frac{1}{|T_n(e_A)|} \sum_{s\in\mathcal{S}^\alpha}^{|\mathcal{S}^\alpha|} \binom{\beta}{\gamma(1)\gamma(2)...\gamma(n_c)} \times \\ &\left|T_{n_c}\left(\frac{e_T}{\beta} - n_1^\alpha\right)\right| ... \left|T_{n_c}\left(\frac{e_T}{\beta} - n_\beta^\alpha\right)\right| \end{aligned} \tag{5.42}$$

Here, $\mathbb{P}_C$ is the probability that a sequence of length n with e_A errors is correctable

by using error correcting codes of block size n_c and error correcting capability e_T/β to reconcile the keys.

5.4.3.3 Interleaving

Interleaving a bit sequence involves randomly switching the positions in a bit sequence. It is typically used in wireless communications to combat burst errors. In the proposed scheme, after a measured sequence is quantised into binary bits, the resulting sequence is interleaved. This is for two purposes to help in combating burst errors and aiding key reconciliation.

As outlined earlier, error correction in key reconciliation will only be successful if the error vector between the two sequences that need to be reconciled, $\overline{\boldsymbol{e}} = H_D(\mathrm{x}_A^n, \mathrm{x}_B^n)$, is of the form $F_{n,n_c}(\{n_1, n_2, ..., n_\beta\})$ where $\max\{n_i\} \leq e_T/\beta$. In other words, the first n_c bits in the sequences have less than n_1 bits mismatching, the second block of n_c bits in the sequences have less than n_2 bits mismatching, etc. Where all blocks have less than e_T/β bits mismatching. The quantity $\sum_i n_i = e_A$ is the number of mismatching bits or errors.

The probability that some arbitrary pair of sequences which differ by e_A bits and have a form described above, is $\mathbb{P}_\mathrm{C}$ and thus the probability that they cannot be reconciled is $(1 - \mathbb{P}_\mathrm{C})$. If we have a pair of sequences, x_A^n and x_B^n, which have $H_D(\mathrm{x}_A^n, \mathrm{x}_B^n) < e_T$ but are not reconcilable, then synchronously interleaving the sequences once could make the resulting sequence reconcilable. The probability for this is at most $\mathbb{P}_\mathrm{C}$.

If interleaving is used iteratively k times and all those k permutations of the binary sequences are random, then the probability that all of those permutations are unreconcilable is $(1 - \mathbb{P}_\mathrm{C})^k$ and thus the probability that one of them is reconcilable is $1 - (1 - \mathbb{P}_\mathrm{C})^k$. A syndrome can thus be created for each permutation of the binary sequence that needs to be reconciled and those syndromes are sent wirelessly to other legitimate nodes. These nodes can then permute their sequences in a similar manner and attempt reconciliation with all syndromes. It is important to note that if multiple syndromes are used, every syndrome should be generated using different random numbers. The multiple syndromes can then be sent as one packet.

Iteratively permuting the sequences can thus aid in key reconciliation although care

needs to be taken so that the scheme remains efficient. This is because having too many iterations may prove to be energy inefficient as it increases the number of bytes transmitted in the key reconciliation stage.

5.4.3.4 Extension to the Group Setting

In the group setting, not all legitimate nodes have sequences that are correlated to the same degree. This poses additional challenges that need to be considered when designing the relevant group key generation scheme. In particular, passive node n_C that gets its measurements by combining RSS measurements from the active nodes' probes will have readings that do not correlate as highly as with the active nodes and thus will not be able to jointly reconcile its key if the threshold has been set as in the pairwise case.

To this end, threshold e_T has to be set at a lower level. The key reconciliation process can thus proceed in the same way as in the pairwise case but with the syndrome sent to third node n_C as well. The third node, knowing the positions of the legitimate nodes, tries to approximate readings as closely as possible. The proposed GPlSKG scheme shows how this key generation process can be achieved with three nodes - n_A, n_B and n_C.

In cases where there are more than three nodes the scheme extended to account for the additional legitimate nodes although some important factors need to be taken into account. When used in an expanded group PLSKG set-up (i.e. used in situations where there are more than 3 nodes), the node that should be chosen to act in the exact same way as n_C should be the node with the most degraded channel (i.e the node with the 'noisiest' channel with respect to the active nodes). All arguments made in this section would then still hold. Care needs to be taken that the error threshold (e_T) is set to a threshold value that is lower than the expected number of errors at the adversarial node (n_E). If any of the legitimate nodes have wireless channels which are noisier than the adversary's channel then key generation can not take place as there would be no secrecy capacity. If the adversary is not collocated with respect to the legitimate nodes then it can be shown that there will be secrecy capacity.

The analysis above shows that the size of the key the nodes need to agree when

using the scheme is dependent on the lowest long term correlation between any two legitimate channels. This means that we would expect the key generation rate to fall as the number of nodes in a group increases. Also as members leave the group, new keys would not be able to be generated if the nodes still remain collocated with respect to the nodes that are left to the group. If, on the other hand, nodes leave the area, then their keys can be refreshed using the pairwise key generation and refreshment scheme detailed in the previous chapter. Similarly, when new nodes move into the area to join the group, new keys can be established using the scheme. In addition, the maximum number of nodes in a group, beyond which our scheme would not work effectively, together with the location distribution of the nodes still needs further investigation, which this research project does not have sufficient sensor nodes to carry out.

5.4.4 Privacy Amplification

The key idea of privacy amplification is to conduct further processing on the key in order to ensure that the reconciled key is fit for use for cryptographic purposes. In order for privacy amplification to be successful, the legitimate nodes need to have the same sequence, so privacy amplification is carried out after the error correction stage. The main operation in this stage is the hashing of the reconciled sequence (together with other information) to form the key used for communication. If the key generation scheme is used for key refreshment, then the reconciled key can be hashed with a previous key in order to form a new session key. The privacy amplification stage in the scheme is conducted similarly to the pairwise scheme in [4], which is outlined below for the case of key refreshing.

Let the reconciled key in the current i^{th} session be denoted as K^i, the previous session key as K^{i-1}, the recovered random number from the key reconciliation stage as $\boldsymbol{r}_i$, the reconciled sequence as x_i, the i^{th} session's secret information as f_i and a secure hash function as $h(\cdot)$. The current key is computed by each node as:

$$f_i = h(f_{i-1}||\mathrm{x}_i) \tag{5.43}$$

$$K^i = h(K^{i-1}||f_i||\boldsymbol{r}_i||i) \tag{5.44}$$

After the key has been generated, the nodes can check whether the key generation

has been successful by trading challenge-response packets. If the key generation process has been successful, then the nodes can proceed to use the generated key.

5.5 Detailed Overview Of GPLSKG

A detailed overview of the GPLSKG process can be seen in figure 5.5, which includes three main stages as outlined in earlier sections. In the randomness sharing stage, the nodes measure the receive power of incoming packets across the different frequency channels. The nodes need to agree on what sequence of frequency channels to use beforehand. This information is called the schedule. The i^{th} measured value over the j^{th} channel in the schedule S is V_{ij}. The measured vector consists of the concatenation of the mean RSS values over each channel. The measured vector is filtered by computing the first order difference, which makes the resulting vector zero mean. The vector is then passed onto the key reconciliation stage.

In the key reconciliation stage, n_A computes a number of candidate syndromes with each syndrome produced with a different random number and a sequence formed by quantising the sequence passed on from the randomness sharing stage with a different quantisation vector. The candidate syndromes are transmitted to n_C and n_C sends the index of the syndrome to be used back to n_A. n_A can then proceed to use the syndrome that corresponds to the chosen index for key reconciliation. The chosen syndrome is then broadcast to all legitimate nodes that perform key reconciliation using the procedure in equation 5.35.

The nodes could carry out key reconciliation by just generating keys with each candidate syndrome and then choosing the successfully generated key with the highest entropy in the privacy amplification stage but this would waste computational resources. Instead, the node n_C picks the index of the specific syndrome to use and then sends it to n_A as outlined above. The syndrome chosen is the successfully reconciled syndrome which has been computed using the quantiser with the lowest average separation between quantisation levels.

A simple way to ascertain which syndrome to use is to choose the random numbers used in the key reconciliation stage such that they form a hash chain (i.e. $r_1 = H(r_0), r_2 = H(r_1), ..., r_n = H(r_{n-1})$). Note that if a node can perform reconciliation

with a syndrome that has been computed using a sequence that has been quantised with a vector $Q^u_{L_1}$, then the node can also reconcile using a syndrome that was formulated with $Q^u_{L_2}$, where $L_2 > L_1$. The node can thus just pick the syndrome with the lowest separation distance that has also been deemed as being successfully reconciled. Where the successful reconciliation of the syndrome s_i with index i is ascertained by checking if $r_{i+1} = H(r_i)$.

If the key reconciliation stage is performed with an ECC of length n and decoding threshold e_{T} and also the uniform quantisation is used, then the lowest acceptable average separation between quantisation points can be set to a value no less than $2R/L$. Where the range of the RSS values is $[-R, R]$ and L is determined such that:

$$\begin{aligned} \log_2 L &> I(X_A; X_E) + \frac{1}{n} e_{\mathrm{T}} \\ &= -\frac{1}{2}\log_2(1 - \rho^2_{\mathrm{AE}}) + \frac{1}{n} e_{\mathrm{T}} \end{aligned} \tag{5.45}$$

Here ρ_{AE} is the maximum correlation between the adversarial node and the legitimate node, which the scheme is designed to be secure against. Note that if nodes n_{A} and n_{E} both sample random variables X_{A} and X_{E} n times respectively to form two sequences $\mathrm{x}^n_{\mathrm{A}}$ and $\mathrm{x}^n_{\mathrm{E}}$, then the amount of information they share is dependent on the mutual information between X_{A} and X_{E} (i.e. $I(X_{\mathrm{A}}; X_{\mathrm{E}})$). If nodes n_{A} and n_{E} want to derive a common bit sequence from their respective observations, then the longest sequence they can derive with an arbitrarily small bit error probability is $nI(X_{\mathrm{A}}; X_{\mathrm{E}})$ information bits long. It then follows that if the sequences $\mathrm{x}^n_{\mathrm{A}}$ and $\mathrm{x}^n_{\mathrm{E}}$ both have $n \log_2 L$ information bits then at least $n \log_2 L - nI(X_{\mathrm{A}}; X_{\mathrm{E}})$ bits would need to be corrected in order to ensure that the bit sequences reconcile. It then follows that if only e_{T} disagreeing bits are corrected in a sequence of length $n \log_2 L$, then n_{A} and n_{E} will not be able to reconcile their sequences if $n \log_2 L - nI(X_{\mathrm{A}}; X_{\mathrm{E}}) > e_{\mathrm{T}}$. This leads to the relation in 5.45.

After the key has been reconciled by all the nodes, the key is passed onto the privacy amplification that is conducted in the way outlined in section 5.4.4.

5.6 Implementation, Evaluation and Comparison

5.6.1 Implementation of Proposed GPLSKG Scheme

In order to evaluate the performance and practicality of the secure group key generation scheme proposed in this paper, we implemented the scheme on wireless sensor nodes using the NesC programming language. The nodes ran the TinyOS operating system, which is an operating system for low power communication devices. Three nodes, n_A, n_B and n_C, follow a straight-line topology similar to the one shown in figure 5.2, with n_C equidistant from n_A and n_B.

Note that because n_C's measurements are estimated observations, they will not correlate with n_A's measurements to the exact same degree as the measurements at n_B. This leads to i) lower entropy keys and ii) additional restriction on the adversary relative to the pairwise case. The later condition relates to ensuring that the adversarial channel is degraded (and hence worse than n_C's channel). Experiments were in a line-of-sight, indoor setting ran over different separation distances with a random frequency hopping schedule (i.e nodes pick a sequence in which to cycle through different frequencies randomly before the key generation process starts).

Previous GPLSKG schemes have been designed and targeted for use by mobile nodes and they thus measure the values of various properties of mobile channels and use that information to generate keys. The schemes in [47][48] for instance use the fading trend in mobile channels to generate a common key between legitimate nodes. Due to the fact that our proposed scheme is meant for stationary nodes, a direct comparison of SKGR between the two is not straightforward. However, comparisons can still be made of the various components of the different schemes (e.g. the quantisation and error correction operations) and the case for why our proposed scheme allows keys to be generated in a more secure manner.

The proposed scheme will therefore be compared in sub-section 5.6.2 with state-of-the-art GPLSKG schemes which are known to have practical implementations. The most relevant of these are the schemes in [47][48].

Group key generation was conducted using i) various average separations of the

quantisation points in the quantiser and ii) two different binary encoding procedures - the 2's complement encoding and the proposed uniform distance encoding procedure. Figures 5.6 - 5.11 show the achieved SKGR for different separation distances and the two encoding procedures. Further comparison with the current state-of-the-art practical GPLSKG schemes is given in subsection 5.6.2.

When using 2's complement encoding, increasing the quantisation levels increases the SKGR. The SKGR also reduces with increasing separation distance between nodes. When using uniform encoding, the SKGR remains near 1 for low separation distances but then starts to decrease as the separation distance also increases. In terms of the effect of quantisation intervals, it is clear from the figures that increasing the quantisation interval increases the SKGR in most instances, which one would naturally expect, but what the figures also show is that using the uniform encoder provides improvements in SKGR as illustrated in figures 5.6 - 5.7. This is due to the fact that for the proposed uniform distance encoder, if two different sequences with a set difference of $\overline{\boldsymbol{d}}$ are quantised, then the output binary sequences will also vary by $\overline{\boldsymbol{d}}$.

This means that if the expected difference between two vectors is d, then the difference between the two vectors will still be d after they are quantised with the uniform distance binary encoder. In the case of 2's complement encoding, the two quantised vectors could have a difference of more or less than d. In PLSKG, we know that the legitimate nodes have access to RSS measurements that are more closely correlated to each other than the measurements available at the adversarial node and thus have a lower expected difference to each other relative to the adversarial node. The key goal here is to ensure this property is maintained even after the quantisation process. The key reconciliation process can then be designed in a manner that only allows errors less than a given threshold to be corrected. Thus the threshold has to be set to a value greater than the expected number of errors at the legitimate nodes and less than the expected number of errors between a legitimate node and the adversarial node.

This advantage is in addition to the main advantage outlined earlier, which guarantees that the adversary will have a higher expected number of errors in their quantised observations provided that i) the measured sequences length in use is high and ii) the adversary's channel is degraded with respect to the legitimate channels. The

trade-off of using the uniform distance encoder is the binary sequence length, with every quantisation index encoded by the 2's complement scheme being shorter by a factor of $(\log M)/M$, where M is the number of quantisation points.

The above fact means that if a measured RSS value is an integer with a range of M (i.e. $M = RSS_{\max} - RSS_{min}$), then the length of the quantised binary sequence will be proportional to $\log M$ if the 2's complement encoding is used and proportional to M if the uniform distance encoder is used. The uniform distance encoder is thus suitable for low variability channels experienced by resource constrained devices where the range of RSS values taken is not very high.

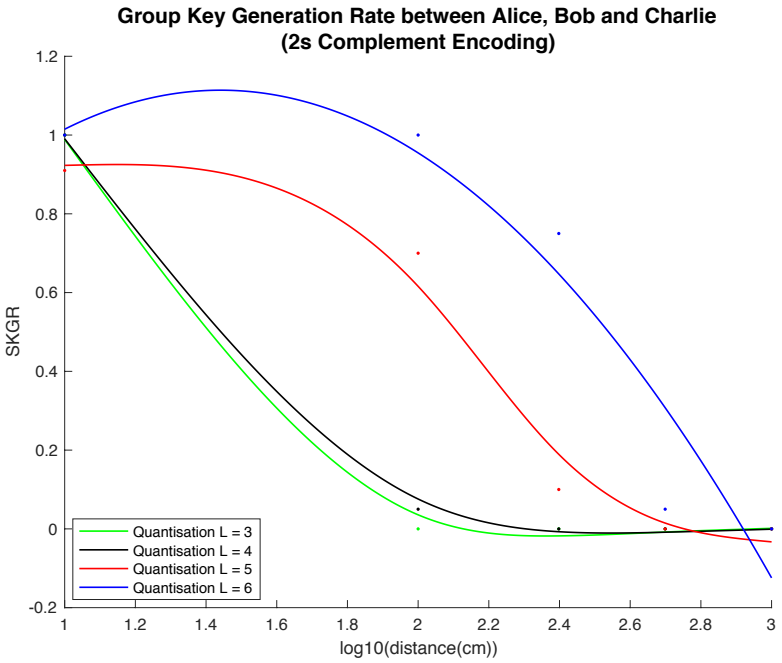


Figure 5.6: Key Generation Success Rate (SKGR) with standard encoding and separation distances $\in [3, 6]$

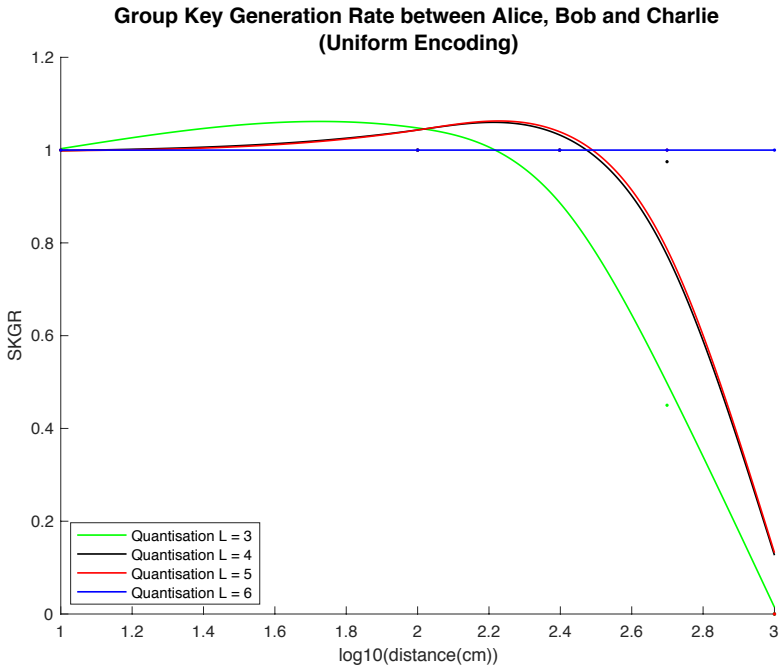


Figure 5.7: Key Generation Success Rate (SKGR) with uniform encoding and separation distances $\in [3, 6]$

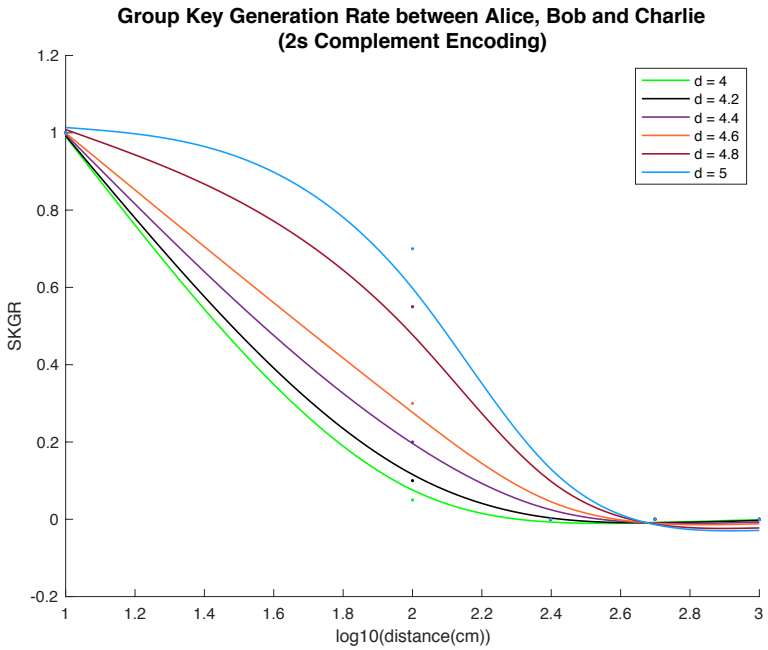


Figure 5.8: Key Generation Success Rate (SKGR) with standard encoding and separation distances $\in [4, 5]$

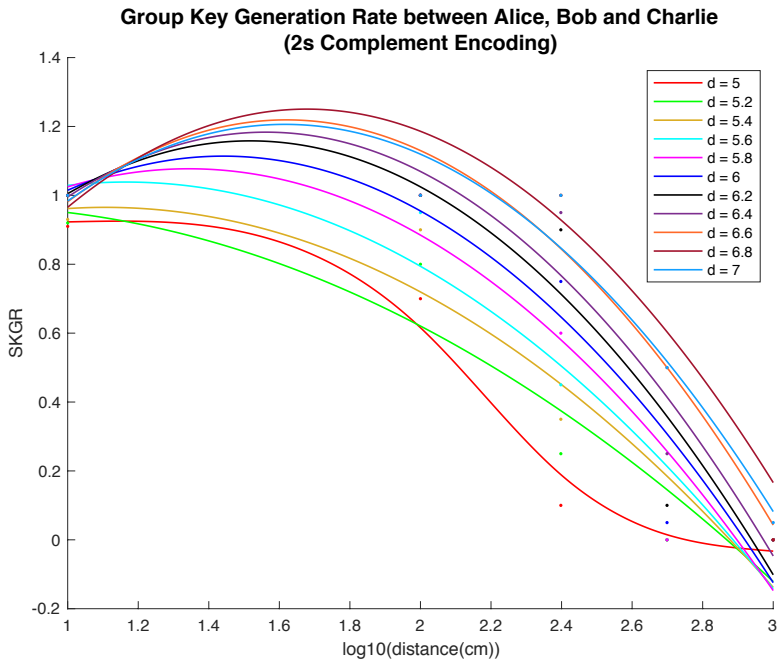


Figure 5.9: Key Generation Success Rate (SKGR) with standard encoding and separation distances $\in [5, 7]$

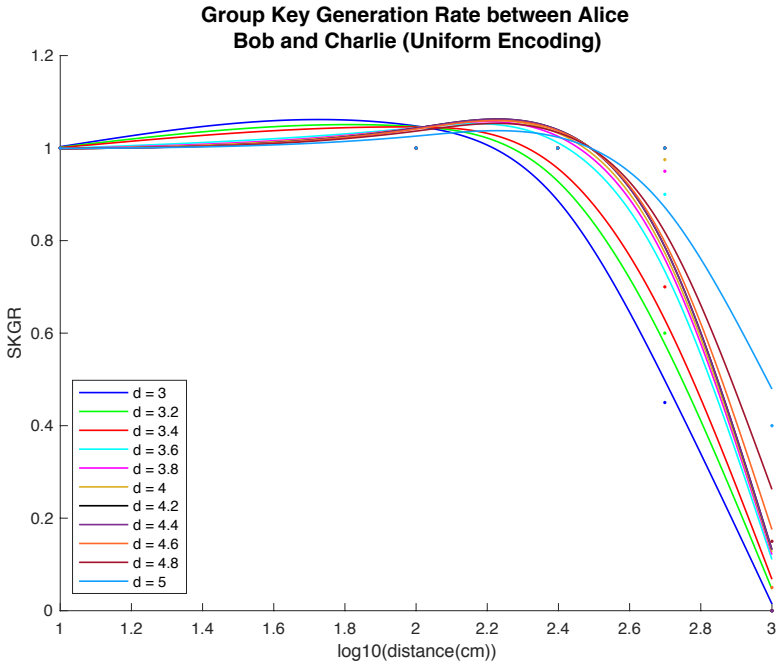


Figure 5.10: Key Generation Success Rate (SKGR) with uniform encoding and separation distances $\in [3, 5]$

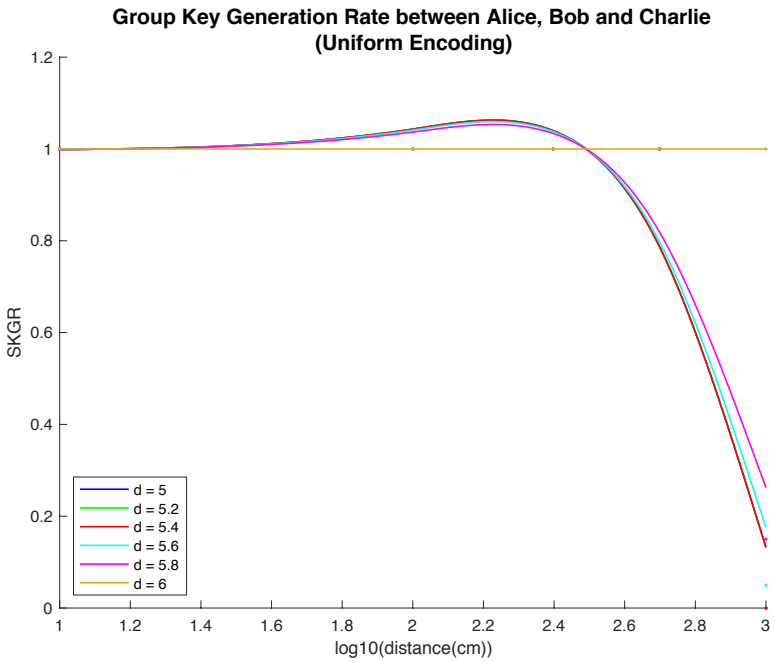


Figure 5.11: Key Generation Success Rate (SKGR) with uniform encoding and separation distances $\in [5, 6]$

The figures on uniform encoding relate to results that have been obtained from the scheme proposed in this thesis whilst the figures that show results relating to the most relevant scheme in literature are labelled as being 2's complement. In the figures, the quantisation interval of a uniformly spaced quantiser, L, is the distance between any two adjacent quantisation points. In a non-uniformly spaced quantiser, the average distance d is the average distance between adjacent quantisation points. Due to the fact that quantisation points might not be uniformly spaced, the value of d might not be an integer even if all quantisation points are integers.

The figures show the SKGR plotted against the logarithm of the distance for different quantisation mappings. The best fit curves are then fitted onto each sequence of points. The figures show that the SKGR is strictly inversely proportional to the logarithm of the distance. The crossings at the beginning and end of each of the best fit curves is due to the curves being restricted to being second order (i.e. of the form $y = a_2x^2 + a_1x + a_0$).

The use of the best fit curves in the graphs allows the general trend in SKGR to be better understood but can cause slight crossing between curves even if the points that correspond to one curve are always lower than or equal to the points in another curve. This can be seen in figure 5.7 where the curve for $L = 3$ crosses the curves

for $L = 4$ and $L = 5$. Fitting a higher dimensional curve could remedy this issue but it leads to the curves being overfitted to the data. A horizontal line, such as the line for $d = 6$ in figure 5.11, illustrates a situation where the quantisation interval used is so large that the quantised key at both ends is always reconciled correctly.

From the results, it is clear that uniform encoding improves the SKGR when compared with normal 2's complement encoding. This improvement is more pronounced at higher separation distances where the SKGR rate increases by a factor of approximately 2. This can be seen particularly at low distances where uniform encoding achieves an almost perfect key generation success rate whilst the key generation success rate falls quickly with distance as separation increases. This is due to the quantisation effects discussed earlier.

5.6.2 Comparison with State-of-The-Art Practical Group Physical Layer Key Generation Schemes

GPLSKG schemes have been proposed for a variety of different wireless networks. This subsection will detail how our proposed scheme differs from other state-of-the-art schemes and highlight its strengths and weaknesses.

The majority of existing GPLSKG schemes have been aimed at high resource networks and rely on channel impulse response measurements although other schemes have been proposed for low power networks. The most relevant practical GPLSKG scheme is the one outlined in [47][48] for low power networks. The scheme is a purely quantisation-based scheme and therefore it suffers from the limitation that some errors cannot be corrected at the end of the quantisation stage. The scheme is relay-based and is aimed at mobile nodes, with the key generation taking place by nodes alternately sending probes to each other and using those probes to try and estimate the losses across one pre-chosen channel.

The key generation of the scheme is done by using two different methods: i) an RSS fading trend and median threshold (RTM) and ii) RSS fading trend and quantisation (RTQ). The fading trend is a sampled, moving average value of the RSS. It is used in lieu of direct RSS measurements in the scheme because the scheme was targeted at mobile networks. In such networks the relative velocity between the nodes causes

frequency dispersion in the channel which then causes time-selectivity (i.e. causes the channel to change quickly in time).

The RTM method involves the nodes using n measured RSS values as a n binary sequence, where each binary digit at position i indicates whether the i^{th} RSS value received was below or above the median of all the RSS values received. The RTQ is a thresholding procedure which involves thresholding using multiple thresholds. The formulation of these thresholds is done by using the probability distribution of the channel between nodes, so the distribution has to be known a-priori or estimated. The thresholds are then set in such a manner that the probability of an RSS value falling between any two adjacent thresholds is constant.

Let $p(x)$ be the probability distribution of the RSS channel between two legitimate nodes, n_A and n_B. Further to this, let $\mathcal{F}_p(x)$ denote the cumulative distribution function of $p(x)$ and $\mathcal{F}_p^{-1}(x)$ its inverse. The thresholds are set in such a manner that the probability that the RSS will fall between any two adjacent thresholds is $1/m$. The set of m thresholds $\{t_i\}$ are the set of values such that

$$t_i = \mathcal{F}_p^{-1}(i/m) \tag{5.46}$$

Formulating the threshold as shown in equation 5.46 ensures that the bit disagreement between two measured RSS values is equal to the number of thresholds that lie between the RSS values. No error coding control is included in the scheme. Nevertheless the scheme is the most representative state-of-the-art scheme for practical key generation.

One of the main drawbacks of the scheme is the need to estimate the probability distribution and corresponding inverse cumulative probability distribution of the RSS channel. This remains quite difficult to accomplish on practical nodes. A node generating different keys with many other nodes will need to estimate the distributions of each RSS power channel and then maintain a store of the sequence of thresholds for each of those channels.

Further to this, in a scheme for stationary or near-stationary nodes where probes are taken over a set of different channels (where each channel is centred around a different frequency), it is not efficient to adapt the scheme to this situation.

If the distribution of the RSS channel for each of the possible centre frequencies is estimated, then the scheme would quickly become exponentially more resource intensive.

The scheme works in a group key generation setting by having nodes estimate the losses incurred over one channel. Consider a case where 4 nodes, n_0, n_1, n_2 and n_3, want to generate a common key corresponding to the channel conditions over the channel between n_0 and n_1. Each node has access to the channel conditions near to them. Let $L_{i,j}$ denote the loss from n_i to n_j, and node n_i has access to $L_{i,i+1}$ and $L_{i-1,i}$. The node n_1 can thus compute the difference in loss between n_0 and n_1 by computing $\delta_{1,2} = L_{1,2} - L_{0,1}$, where $\delta_{1,2}$ is the difference between the losses. $\delta_{1,2}$ can then be passed onto n_2 in order for n_2 to estimate $L_{0,1}$. This process can then continue in a similar manner until all nodes have estimates of measurements of the channel between n_0 and n_1, which they use for key generation. This is the broad idea of the relay model proposed in [47].

This randomness sharing process is perfectly suitable when each node is only in the transmission range of a pair of adjacent nodes and no two nodes are collocated but in other situations a broadcast-based group key strategy works better. This is because the number of probes sent by the scheme scales linearly with the number of nodes, whereas a broadcast-based scheme such as the one proposed in this paper increases sub-linearly with an increasing number of nodes.

Consider the situation where there are L nodes, n_1 to n_L, and the nodes n_2 to n_{L-1} are collocated. Further let these collocated nodes be in the transmission range of n_1 and n_L. In this situation, allowing these collocated nodes to receive broadcast packets from n_1 and n_L is a more energy efficient strategy. Let $\rho_{i,j}$ denote the correlation between the observations at nodes n_i and n_j, $\hat{\rho}_{1,L}$ denote the correlation between the measurements at n_1 from n_L. There are also the measurements estimated by each of the nodes n_2 to n_{L-1}.

In the randomness sharing stage of our proposed scheme, the theoretical number of probes that would need to be exchanged by each of the passive nodes (i.e. n_2 to n_{L-1}) is 0 and the number of probes exchanged by each of the two active nodes (i.e. n_1 and n_L) to generate a key K of entropy H is:

$$n_{\text{Active}} = \frac{H(K)}{-\frac{1}{2}\log_2(1-\hat{\rho}_{1,L}^2)} = -\frac{2H(K)}{\log_2\left(1-\hat{\rho}_{1,L}^2\right)} \tag{5.47}$$

Due to the fact that there are two active nodes, the total probes needed in the randomness sharing thus becomes:

$$n_{\text{Active, T}} = -\frac{2 \times 2H(K)}{\log_2\left(1-\hat{\rho}_{1,L}^2\right)}$$

In the randomness sharing stage of the relay model [47], the number of probes is the same per node at:

$$n_{\text{Relay}} = \frac{H(K)}{-\frac{1}{2}\log_2\left(1-\min\{\rho_{1,2}^2, \rho_{2,3}^2, ..., \rho_{L-2,L}^2\}\right)} = -\frac{2H(K)}{\log_2\left(1-\min\{\rho_{1,2}^2, \rho_{2,3}^2, ..., \rho_{L-2,L}^2\}\right)} \tag{5.48}$$

Due to the fact that there are L nodes, the total number of probes is:

$$n_{\text{Relay, T}} = -\frac{2L \times H(K)}{\log_2\left(1-\min\{\rho_{1,2}^2, \rho_{2,3}^2, ..., \rho_{L-2,L}^2\}\right)}$$

$\implies n_{\text{Active, T}} \leq n_{\text{Relay, T}}$, if

$$\left(1-\hat{\rho}_{1,L}^2\right) \leq \left(1-\min\{\rho_{1,2}^2, \rho_{2,3}^2, ..., \rho_{L-2,L}^2\}\right)^{2/L}$$
$$\hat{\rho}_{1,L}^2 \geq 1-\left(1-\rho_{\min}^2\right)^{2/L} \tag{5.49}$$

Where $\rho_{\min}^2 = \min\{\rho_{1,2}^2, \rho_{2,3}^2, ..., \rho_{L-2,L}^2\}$. In the relay-based scheme, its performance is essentially limited by whichever link (or channel) is the weakest. Equation 5.49 outlines the condition that has to be met for our proposed scheme to require fewer probes than the model in [47]. This model is intended for mobile channels and uses fading trends to generate keys, where the variability of the channels sampled by the model is much higher than that faced by our scheme. Hence our scheme achieves variability by changing the centre frequency in use.

In the case with L nodes and all the channels being similar (i.e. where $\rho_{ij} = \rho$ for all i and j), then $n_{\text{Relay, T}} = 0.5L \times n_{\text{Active, T}}$. So having some nodes being active

reduces the required probes by a factor of $0.5L$.

In a home setting where long term correlation between the legitimate channels is for instance 0.98, the number of probes that would be required to generate a group key would be 28 in order to generate a key of length 32. If used in a factory setting and the long term correlation drops to 0.97 then the number of probes required would rise to 32. This illustrates the importance of the long-term correlations between legitimate observations in physical layer key generation.

5.7 Security Analysis

5.7.1 Randomness Testing

In order for keys to be suitable for cryptographic purposes, they need to have the property of randomness. This is to ensure that the keys do not have any underlying correlations between them, which can be exploited by an adversary to compromise the keys. The randomness testing was done using the industry standards for randomness testing which were explained in subsection 4.8.2.

The DFT spectral test was conducted on generated keys, which resulted in a p-value of 0.457, meaning that our scheme can produce random keys. Another way to analyse the similarity of keys is time. This analysis helps in assessing how the generated keys vary between sessions. In order to assess this, 75 keys were generated and each of the keys was correlated with other keys. The correlation measurement used in this case was based on the Pearson correlation coefficient. Figure 5.12 shows the correlation between the keys reconciled by three nodes n_A, n_B and n_C. From this figure, it is clear that in the vast majority of cases, the correlation between the keys is very low. The diagonal in figure 5.12 refers to the autocorrelation between the keys.

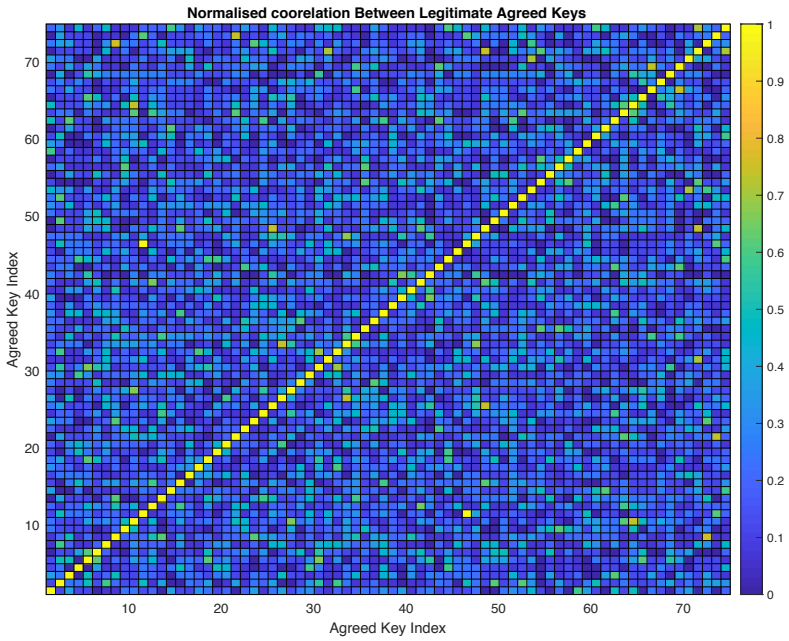


Figure 5.12: Correlation between common keys among legitimate nodes n_A, n_B and n_C

5.7.2 Security Against Common Attacks

It is important for any scheme to be analysed in light of possible attacks that the scheme might face. Physical layer schemes face attack vectors just like computational security schemes.

The attack of utmost relevance to our proposed scheme is the one where the adversary receives probes sent for the purposes of key generation by the legitimate nodes and then attempts to reconcile a common key with them without their knowledge. In order to better assess this risk, experiments were conducted by adding the fourth node n_E as the adversary. Similar to the previous experiments, the three legitimate nodes, n_A, n_B and n_C, were in the straight-line topology starting with n_A. The distance between n_A and n_B was 4m and n_C was equidistant between n_A and n_B. n_E had a distance of 6m from n_B, 10m from n_A and 8m from n_C.

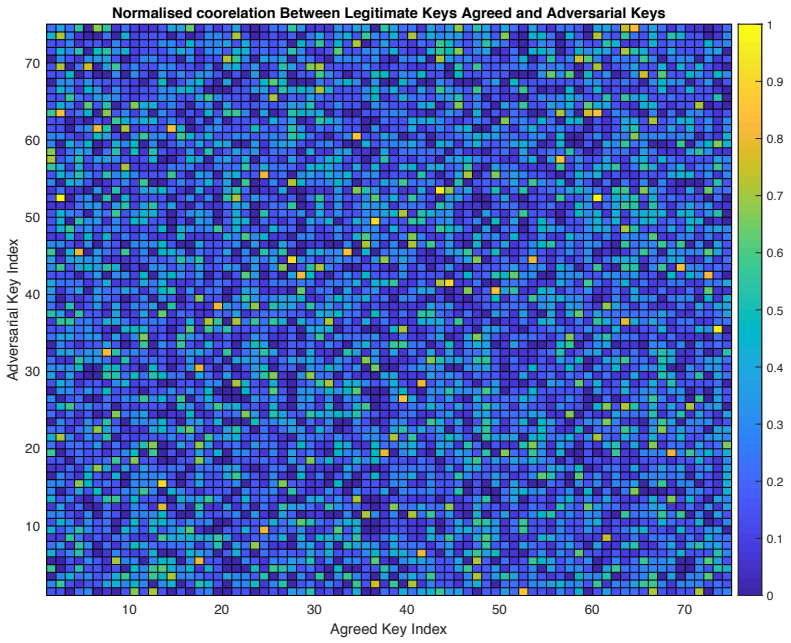


Figure 5.13: Correlation between common keys among n_A, n_B & n_C and n_E's keys

A total of 75 keys were computed by both the legitimate nodes and the adversarial one. The frequency schedule changed between key generation rounds (i.e. each key is generated over a different set of centre frequencies). A correlation map for the legitimate keys versus the adversarial keys can be seen in figure 5.13. From the figure, it is clear that a legitimate key correlates very lowly with an adversarial key.

5.8 Summary

This chapter proposed a key generation that can be used for pairwise and group key generation purposes. The scheme improves on previous state-of-the-art schemes by providing a quantisation method that assures that adversaries with measurements that correlate less strongly with the correlation between legitimate parties get more bit errors in their quantised binary sequences. The chapter also provides an analysis of the probability of legitimate nodes successfully reconciling keys in the case where they use segmented block coding techniques.

The analysis of i) an achieved successful key generation rate between legitimate nodes, ii) cryptographic properties of the generated keys and importantly iii) the successful key generation rate of an adversarial node is presented and evaluated to demonstrate the scheme's advantages over relevant existing approaches. The scheme

also reduces the number of probes required to generate a common GPLSKG key. This increases the energy efficiency of the scheme.

The chapter then presents the results of experiments conducted on an implementation of the proposed scheme in a straight-line topology. The results show that the scheme can achieve high success rates at short distances and that the keys generated correlate lowly between sessions and also correlate lowly with those of adversaries that are no collocated with respect to the legitimate nodes.

Chapter 6

Design of Practical Physical Layer Key Generation Schemes

This chapter introduces and discusses typical wireless communication hardware used in resource-constrained networks and then discusses a selection of practical issues that need to be taken into account when designing and deploying physical layer key generation schemes on real hardware.

6.1 Resource Constrained Network Hardware

A WSN node consists of a micro-controller connected to a transceiver and a set of sensors. The micro-controller controls the entire operation of the device and most of the device logic is stored in it. The transceivers job is to facilitate communication and send packets from one node to another. The wireless sensor node used for experiments in this thesis is the TelosB sensor node which consists of a CC2420 transceiver [2], a MSP430 microcontroller [62] and various sensors. This node provides communication over the low-power 802.15.4 standard. A figure showing the components that make up a TelosB wireless sensor node can be seen in figure 6.1.

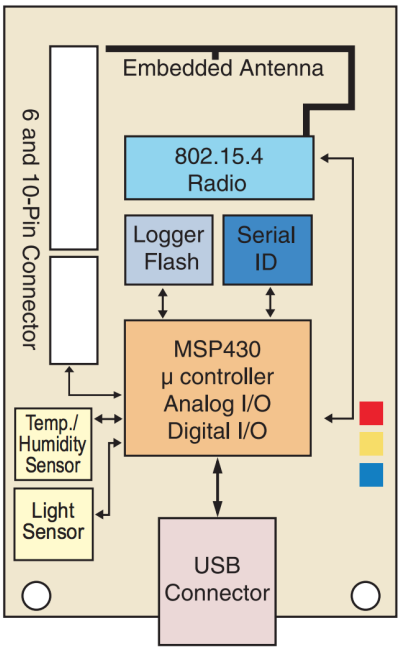


Figure 6.1: TelosB Wireless Sensor Node [3]

6.2 Considerations on Practical Implementations of Physical Layer Security Schemes

A key aspect affecting the secureness of a physical layer security key scheme is the transceiver's characteristics. It is the transceiver that provides the physical layer measurements used to generate the cryptographic keys, so it is very important for the transceiver to be characterised and analysed. This is especially important with regards to the received power as it is the key channel measearement used by low power physical layer key generation schemes.

The received power (RSS) is provided to the sensor node by the transceiver. There are two key characterisations given on a transceiver data sheet that characterise the RSS reading. These are i) the RSS value linearity and the ii) the RSS value's accuracy. The RSS accuracy specifies the range of values around the true RSS value that the transceiver will report and the linearity specifies how linear the readings that are computed by the transceiver will be. The linearity is always lower than the accuracy.

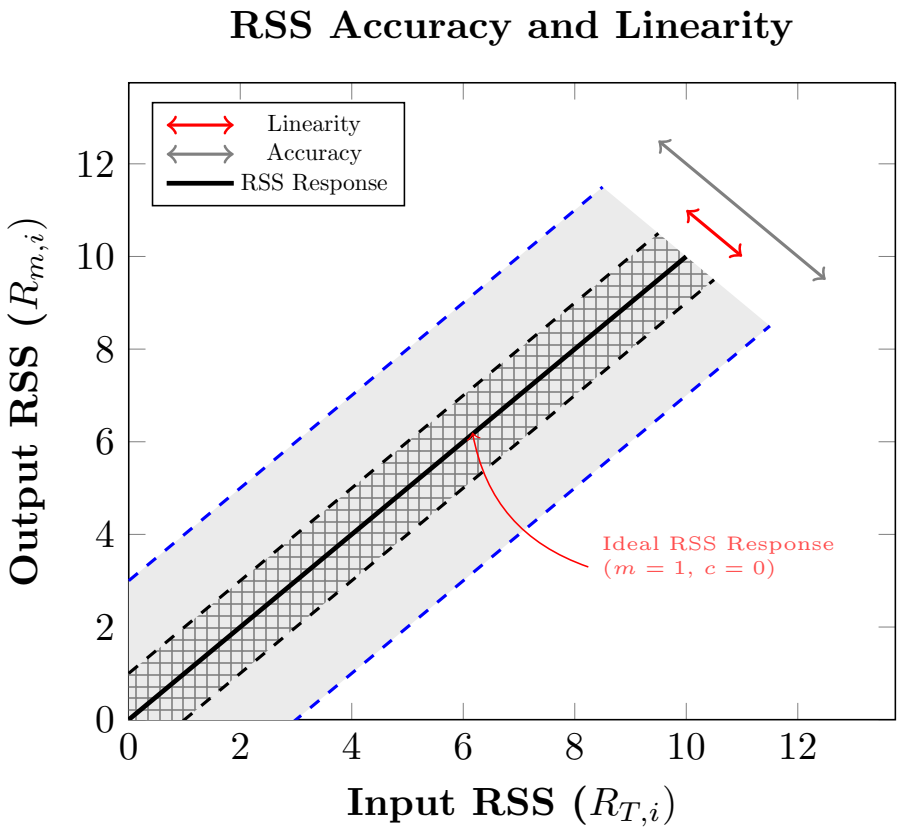
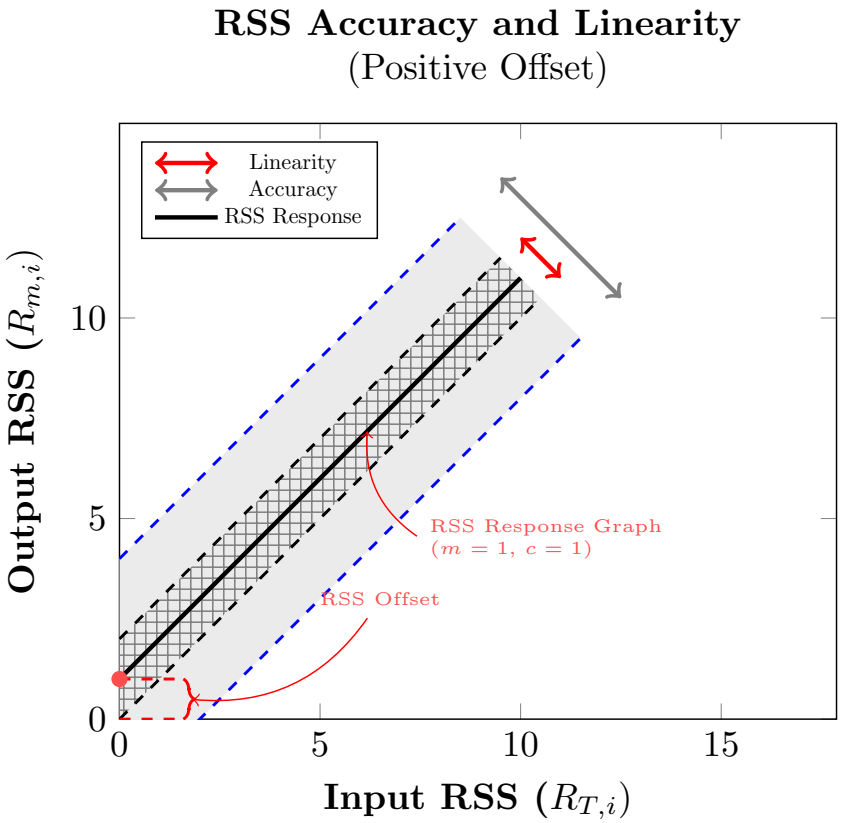
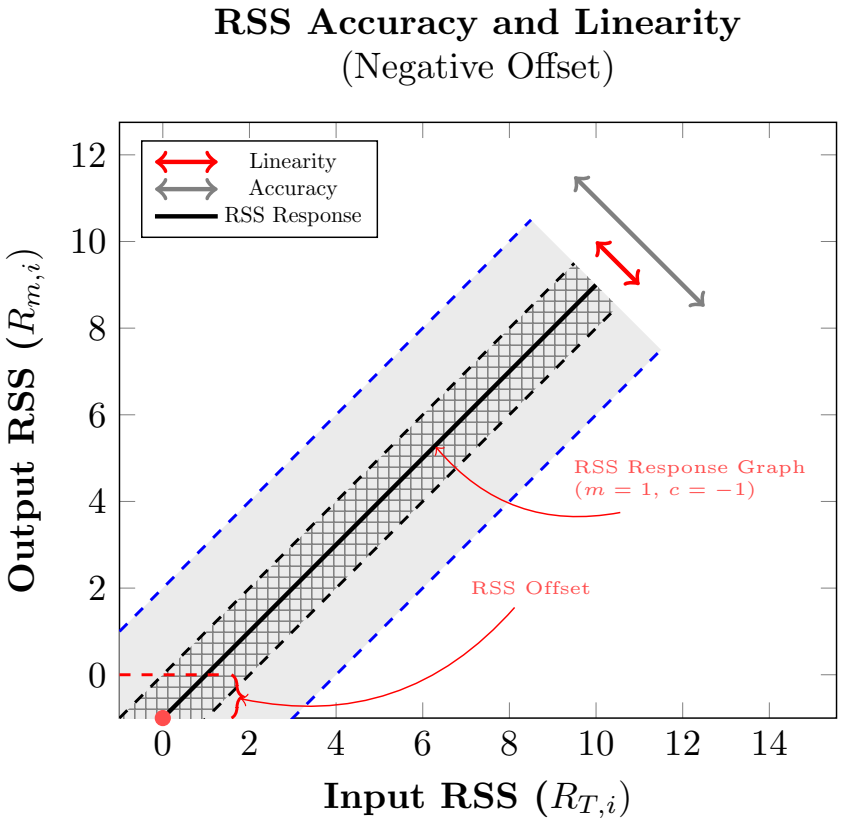


Figure 6.2: Ideal RSS Response Graph



(a) RSS Response Graph with Positive Offset



(b) RSS Response Graph with Negative Offset

Figure 6.3: RSS Response Graphs

The RSS response graph in figure 6.2 illustrates how the RSS, its accuracy and linearity are related. The output RSS, $R_{m,i}$, is related to the input RSS, $R_{\mathrm{T},i}$, according to relation $R_{m,i} = mR_{\mathrm{T},i} + c + r$. Here, m and c are constants, $r \in [-l, l]$, where l is a constant called the linearity. The best fit line between a plot of $R_{m,i}$ versus $R_{\mathrm{T},i}$ is thus a straight line with a gradient of m and y-axis offset of c. The RSS response graphin figure 6.2 shows the graph in the ideal case of $m = 1$ and $c = 0$.

In an ideal case, an actual RSS value and its measured one would be exactly the same. If there is an inaccuracy of a, then the measured RSS value would differ from the actual RSS by at most a dB. In addition to this, if a linearity l is also specified, then there exists a straight line that is at most l dB away from each of the possible measured RSS values. We will refer to the average power of a packet being received by node A at time index i as the input RSS at node A, denoted as $R^A_{\mathrm{T},i}$. We will refer to the measurement of $R^A_{\mathrm{T},i}$ by A as the output RSS at A, denoted as $R^A_{m,i}$.

The presence of some variability in measuring RSS in practical devices adds additional complexity to physical layer key generation schemes as it makes the channel more asymmetric. This needs to be taken into account in order to improve key generation rates and also maintain the security. The existence of a given linearity level in RSS values means that for any given transceiver there are many different RSS response graphs that can be observed. Figures 6.3a and 6.3b show the different RSS response graphs that are possible for a given accuracy level.

In order to maintain the symmetry of the channel as much as possible, it is important to make sure that the non-linearity of the RSS values affects the key generation process in the most minimal way possible. If the accuracy of RSS reading is stated as being $\pm a_m$ then for a received power value $R_{\mathrm{T},i}$ at time index i, the measured RSS value $R^A_{m,i}$ at node A will be:

$$R^A_{m,i} = R_{\mathrm{T},i} + a_A \tag{6.1}$$

where $a_A \in (-a_m, a_m)$. Similarly, for the same $R_{T,i}$, the measured RSS value $R^B_{m,i}$ at node B will be:

$$R^B_{m,i} = R_{\mathrm{T},i} + a_B \tag{6.2}$$

where $a_B \in (-a_m, a_m)$. The important thing to note here is that the difference between the measured values at nodes A and B will be at most $2a_m$ even for the same received power. This can be seen by noting that the value of $|R^A_{m,i} - R^B_{m,i}| \leq 2 \cdot a_m$.

A linearity of $\pm l$ at each receiver means that the measured value and the actual RSS values are related by the relation below:

$$R^A_{m,i} = m_A R_{\mathrm{T},i} + C_A + l_{A,i} \tag{6.3}$$

where m_A and C_A are constants, and $l_{A,i} \in (-l_m, l_m)$. The special case where l_m is 0 is the case when the measured value is perfectly linear and there is no non-linearity. This case which is not seen in practical transceivers. Similarly the measured value at node B will be:

$$R^B_{m,i} = m_B R^B_{\mathrm{T},i} + C_B + l_B \tag{6.4}$$

If there is another received power level $R_{T,j}$, then the measured values at nodes A and B will be $R^A_{m,j}$ and $R^B_{m,j}$ respectively. The difference between $R^A_{m,i}$ and $R^A_{m,j}$ is:

$$\begin{aligned} d^A_{ij} &= |R^A_{m,j} - R^A_{m,i}| \\ &= |m_A R_{\mathrm{T},i} + C_A + l_{A,i} - m_A R_{\mathrm{T},j} - C_A - l_{A,j}| \\ &= |m_A (R_{\mathrm{T},i} - R_{\mathrm{T},j}) + (l_{A,i} - l_{A,j})| \end{aligned} \tag{6.5}$$

The expression in equation 6.5 gives us a relation of the difference between two measured values $R^A_{m,i}$ and $R^A_{m,j}$ when the node A receives two different waveforms with two different received powers $R_{\mathrm{T},i}$ and $R_{\mathrm{T},j}$. Similarly, if node B receives the exact same waveforms then it will measure the receive powers as $R^B_{m,j}$ and $R^B_{m,j}$. The difference between them is

$$d^B_{ij} = |m_B (R_{\mathrm{T},i} - R_{\mathrm{T},j}) + (l_{B,i} - l_{B,j})| \tag{6.6}$$

We want to look at how the expressions d^A_{ij} and d^B_{ij} change in practical transceivers

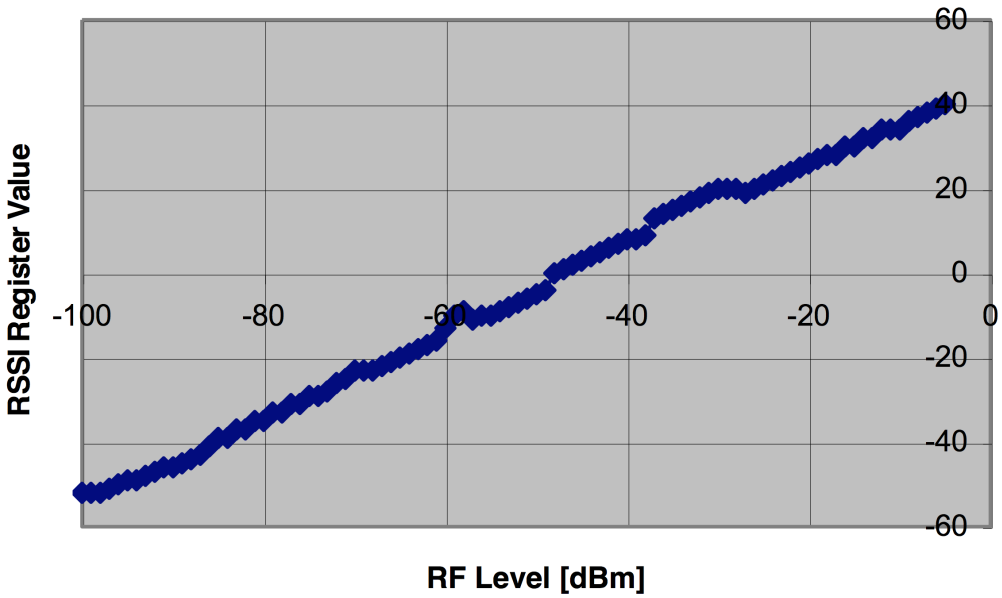


Figure 6.4: Plot of Output RSS vs Input RSS for CC2420 Transceiver. Source [2]

when $R_{\mathrm{T},i}$ is close to $R_{\mathrm{T},j}$. Taking the CC2420 as an example tranceiver, we can look at how the RSS response changes over small intervals. The RSS response graph (i.e. the plot of the received power versus the measured power) can be seen in figure 6.4. The plot has been taken from the CC2420 datasheet in [2].

It can be seen from figure 6.4 that the graph is not perfectly linear (i.e not all points fall strictly in a straight line) but over intervals that are not very long, the response can be taken to be approximately linear. In cases where $R_{\mathrm{T},i}$ is close to $R_{\mathrm{T},j}$, we can thus approximate the measured values $R_{m,i}$, $R_{m,j}$ and all RSS values between $R_{m,i}$ and $R_{m,j}$ to fall in the same straight line. The measured RSS values can then be represented in the form:

$$Y = mX + C \tag{6.7}$$

where X is the receive power, Y is the measured receive power and m & C are constants. The measured RSS values can then be represented in the form shown below

$$R^A_{m,i} = m_A R_{\mathrm{T},i} + C_A$$
$$R^A_{m,j} = m_A R_{\mathrm{T},j} + C_A$$
$$R^B_{m,i} = m_B R_{\mathrm{T},i} + C_B$$
$$R^B_{m,j} = m_B R_{\mathrm{T},j} + C_B$$

Now the expressions for differences $d^A_{i,j}$ and $d^B_{i,j}$ defined earlier become:

$$\begin{aligned} d^A_{i,j} &= |R^A_{m,i} - R^A_{m,j}| \\ &= |m_A R_{\mathrm{T},i} + C_A - (m_A R_{\mathrm{T},j} + C_A)| \\ &= |m_A (R_{\mathrm{T},i} - R_{\mathrm{T},j})| \end{aligned} \tag{6.8}$$

$$\begin{aligned} d^B_{i,j} &= |R^B_{m,i} - R^B_{m,j}| \\ &= |m_B R_{\mathrm{T},i} + C_B - (m_B R_{\mathrm{T},j} + C_B)| \\ &= |m_B (R_{\mathrm{T},i} - R_{\mathrm{T},j})| \end{aligned} \tag{6.9}$$

Equations 6.8 and 6.9 show that the difference between any two measured values is a value which is directly proportional to the difference between the two input RSS levels. The proportionality factor, m, is approximately 1.

This implies that if node A and node B both receive two packets which arrive at their respective transceivers with received powers $R_{\mathrm{T},i}$ and $R_{\mathrm{T},j}$ then the two nodes both compute a value d_{ij} which is not dependent on the offsets of their respective RSS response graphs, C_A and C_B. In addition to this, if the linear dependence on the gradient factors, m_A and m_B, is also close to 1 (as is the case in the CC2420 transceiver) then both nodes can fairly accurately compute the difference between the received powers ($R_{\mathrm{T},i}$ and $R_{\mathrm{T},j}$) and use that difference in the physical layer key generation process instead of using the measured RSS values directly. In the case that n packets are received, one can use the the measures RSS values for each packet to compute the difference vector $\mathbf{d}$ as:

$$\begin{aligned}\mathbf{d} &= \{R_{m,i+1} - R_{m,i}\}_{i=1}^{i=n-1} \\ &= [(R_{m,2} - R_{m,1}), (R_{m,3} - R_{m,2}), ..., (R_{m,n} - R_{m,n-1})] \\ &= [m(R_{T,2} - R_{T,1}), m(R_{T,3} - R_{T,2}), ..., m(R_{T,n} - R_{T,n-1})] \qquad (6.10)\end{aligned}$$

This is in contrast to using the measured RSS value array directly, which is in the form:

$$\mathbf{R} = \{R_{m,i}\}_{i=1}^{i=n} \qquad (6.11)$$

$$= [R_{m,1}, R_{m,2}, ..., R_{m,n}] \qquad (6.12)$$

The use of the difference array in lieu of using the sequence array directly allows us to reduce the errors that may be caused by different linear offsets being present in the individual RSS response graphs. The difference array $\mathbf{d}$ produces $(n-1)$ values for every n RSS values measured. If we wish to remove the dependence of the linear offsets in the transceivers whilst not reducing the number of distinct points that result from the measuring process then we can opt for an alternative method where we just remove DC offset from the measured RSS values. This is the method that was proposed in the key generation schemes detailed in this thesis. We will show that just removing the DC offset is also sufficient for suppressing the effects of practical linear offsets in the practical transceiver.

The removal of the DC component from an array $\mathbf{R}$ involves deducting the array mean, $\overline{\mathbf{R}}$ from each of the values in $\mathbf{R}$. In the situation we are concerned with where the RSS response is linear and the measured value at time index i is $R_{m,i}$, the array with the DC component removed can be shown to be

$$
\begin{aligned}
\mathbf{R}_{\mathrm{DC}} &= \mathbf{R} - \frac{1}{n}\sum_{j=1}^{n} R_{m,j} \\
&= \left\{ R_{m,i} - \frac{1}{n}\sum_{j=1}^{n} R_{m,j} \right\}_{i=1}^{i=n} \\
&= \left\{ (mR_{T,i} + C) - \frac{1}{n}\sum_{j=1}^{n} (mR_{T,j} + C) \right\}_{\forall i} \\
&= \left\{ (mR_{T,i} + C) - \frac{1}{n}\sum_{j=1}^{n} mR_{T,j} - \frac{1}{n}\sum_{j=1}^{n} C \right\}_{\forall i} \\
&= \left\{ mR_{T,i} - \frac{1}{n}\sum_{j=1}^{n} mR_{T,j} \right\}_{\forall i} \\
&= \left\{ m\left(R_{T,i} - \frac{1}{n}\sum_{j=1}^{n} R_{T,j} \right) \right\}_{\forall i} \\
&= m\left(\mathbf{R}_{\mathrm{T}} - \overline{\mathbf{R}}_{\mathrm{T}} \right)
\end{aligned}
\tag{6.13}
$$

Equation 6.13 shows that in the case where the RSS response is linear the received signal with the DC component removed is directly proportional to the DC component removed array of the input RSS. In addition, if the RSS response graph gradient, m, is close to 1 then the DC component removed values obtained when using measured values will be close to the DC component removed values that would be obtained if the input RSS values where used instead. This is one of the motivations for using the DC removed RSS values in the pairwise PLSKG scheme and the group PLSKG schemes proposed in this thesis.

A key advantage of removing the DC offset instead of computing the difference array as shown in equation 6.5 is that the output from the process preserves the n data points whilst in the difference array case n input data points output only $n-1$ data points. So that decreases the variability upon which one can generate keys.

Another key advantage of using the DC offset removed RSS array instead of using the measured RSS values directly is that the values move to oscillating around the zero point, so they become zero mean. The advantage that brings in data processing on resource constrained devices is that the magnitudes of the RSS being processed becomes smaller as the RSS array being processed oscillates around the zero point. This makes processing easier as it allow smaller word lengths (such as 8 bits and

16 bits) in the data processing.

6.3 Summary

This chapter discussed the key characteristics of the resource constrained hardware, outlined the key characteristics of RSS measurements taken on practical low power networks, highlighted some of the limitations commonly found on low power nodes and proposed some countermeasures to remedy those limitations.

The countermeasures discussed in this chapter help to mitigate some of the limitations that came about from the way RSS is measured and reported on actual WSN nodes. These countermeasures help to ensure that nodes have good and consistent performance on real devices at all times. If the countermeasures are not put in place, then the key disagreement rate is highly variable as it would depend highly on how similar the RSS response graphs between the nodes in use are. If the RSS response graphs are not similar then the performance will degrade, leading to high key disagreement rates. This issue can cause the very undesirable property of the key generation rate performance being highly variable and the entropy of the final key being lower than expected. The countermeasures discussed in this chapter help to mitigate this issue.

Chapter 7

Summary, Conclusion and Further Work

7.1 Summary and Conclusion

In summary, this thesis has discussed and presented proposals for a pairwise key generation scheme and a group key generation scheme for low power wireless networks using the physical layer. In summary, the novel contributions in this thesis include:

- Proposing a novel pairwise PLSKG scheme for resource constrained wireless devices. The scheme takes advantage of both the power and simplicity of classic Error Correcting Codes (ECCs) and also the diversity of frequency channels available on 802.15.4 compliant nodes to generate keys from RSS readings. This thesis has shown that our key generation and refreshment scheme can achieve a near 100% key reconciliation rate whilst also providing perfect forward and backward security.

- Proposing a novel GPLSKG scheme for resource constrained wireless devices. The proposed scheme is novel as it provides a means of evaluating and bounding the entropy of the generated key with respect to an adversary. This is possible in the case that the long-term correlations between all legitimate nodes are known.

  This information allows a concrete lower bound of the generated key's entropy to be established, something which is not possible with current GPLSKG schemes. The scheme is also suitable for stationary nodes as it relies on using

different channel frequencies to induce variability in RSS measurements. This is in contrast to current state-of-the-art schemes which depend on the fading trend between legitimate moving nodes to generate group PLSKG keys.

7.2 Further Work

There are many lines of research that need to be conducted for physical layer key generation schemes to be ready for commercial use. At the moment the limitations are having to have the topology known a-priori, finding better sources of entropy than received signal strength and estimating the amount of extractable entropy in a given channel post-deployment. More detailed discussions on these issues are given in the subsection below.

7.2.1 Localised Dynamic Conferencing with Physical Layer Key Generation

Secure dynamic conferencing refers to the possibility for any subset of a group to establish a shared key. Dynamic conferencing allows an arbitrary subset of members to form a privileged subgroup [25]. For any finitely sized group, the number of possible subgroups that may want to form a secure group rises exponentially. This poses a very difficult problem for key tree-based schemes as nodes may need to maintain many different key trees in order to be able to efficiently conduct secure conferences.

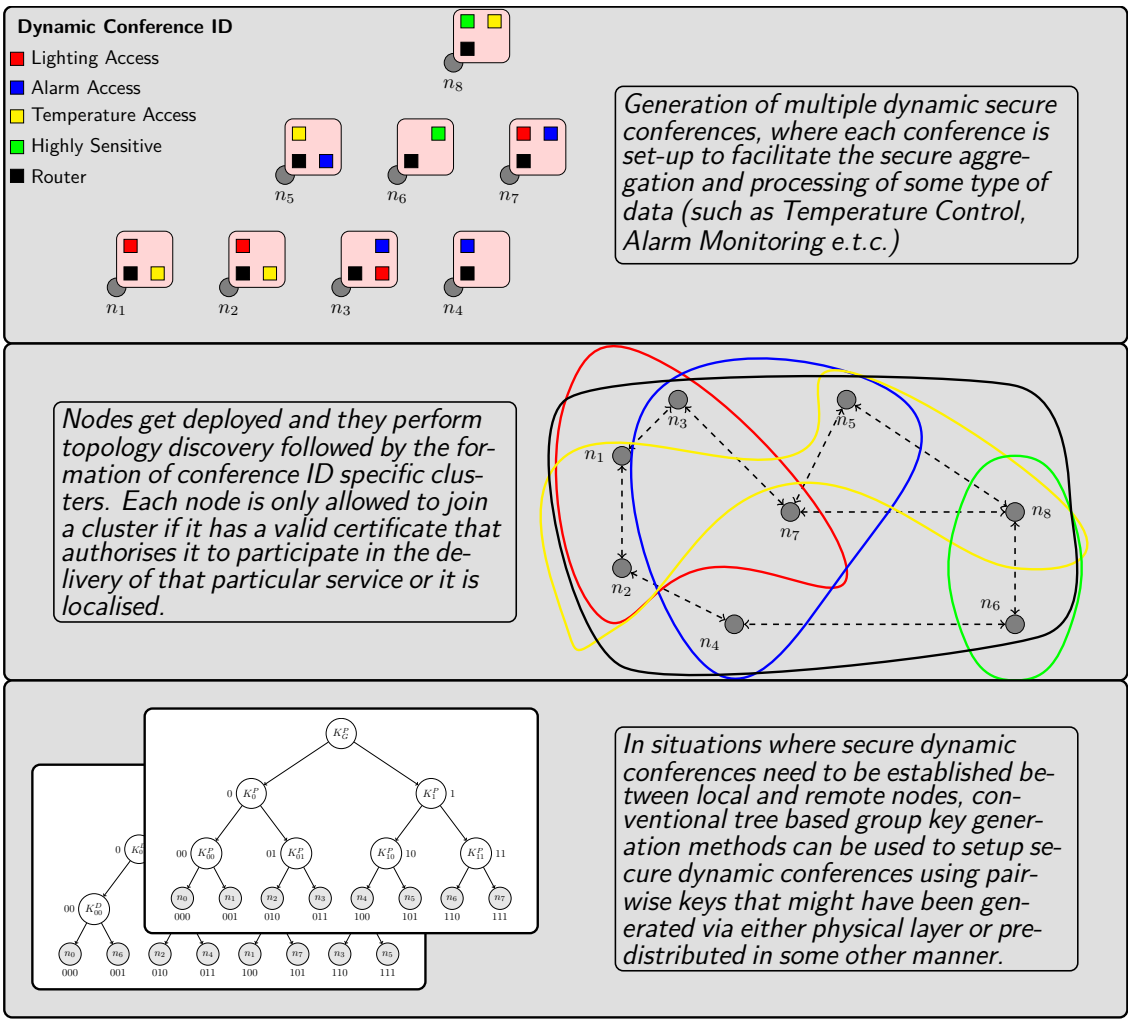


Figure 7.1: Hybrid Dynamic Conferencing

In many applications, secure conferences need to be generated in localised situations. An example application of this is in home networks, where only wireless devices within the home should be able to access particular sets of data. In order to facilitate this, physical layer-based techniques could be used together with conventional group key generation techniques to form both localised and dynamic conferences.

Physical layer key refreshment can be used to ensure that nodes that leave the area are automatically revoked as they move out of the local area and become remote nodes. In cases where a secure conference needs to be established between nodes and some of these nodes might not be localised, the conventional public key methods might need to be added. Figure 7.1 shows an example of secure dynamic conferences. In the figure, nodes granted to a particular class of data (e.g.

temperature data, alarm system data e.t.c) and in a particular area form a dynamic conference in order to securely aggregate their data and forward to other remotely located parties. When new nodes move into the local area and join the group, we would like to have a more efficient way of managing this join operation than just restarting the entire group physical layer key generation process again.

7.2.2 Alternate Sources of Entropy in The Physical Layer

A key issue with using RSS instead of the channel impulse response is that the RSS channel is a lot less variable and hence has a lot less entropy than the channel impulse response function. The reason why RSS has widespread use is that it is both easily accessible in off-the-shelf devices and can be sampled relatively slowly.

If one is to use the channel impulse response function, they would need to be able to work at very high throughout to perform channel estimation and send it over to the micro-controller at high frequency over the serial interface. This is generally not possible on off-the-shelf nodes but it might be possible to use other sources of randomness such as automatic gain control (AGC) readings and/or direct sequence spread spectrum (DSSS) chips to generate keys. These are physical layer parameters that could potentially be suitable lead to the generation of longer cryptographic keys but the manner in which they are reported is not consistent across different WSN devices. Another key difficulty in using these physical layer parameters to investigate is the very high sampling frequency that is required to receive the data.

7.2.3 Beamforming for Physical Layer Key Generation

Beamforming antennas are antennas that allow the waveform that is radiated from an antenna to be directed in some given direction. The use of beamforming allows a legitimate node to direct the waveform it is transmitting towards the other legitimate nodes it wants to generate a common key with. The capability could be leveraged by physical layer key generation schemes to i) efficiently swap probes between legitimate nodes with lower transmit power, ii) restrict the power of the waveform radiated towards the adversary and to iii) allow multiple independent key generation procedures to take place in a smaller area.

Beamforming directs the transmitted signal to a given direction and thus reduces the number of different paths a signal can take whilst traversing from the transmitter to the receiver. This reduction in multi-path propagation causes the channel to experience less slow fading. This is because slow fading is caused by waveforms that have traversed through different paths interefering at the receiver and thus multi-path propagation becomes less prominent and then fading reduces. In the case of beamforming since multipath fading - which contributes to randomness in the channel - reduces, the key generation capacity reduces. Research needs to be conducted on what extent this issue would affect the key generation capacity in practice and also conduct research on how these limitation could somehow be overcome.

Appendices

Appendix A

Appendix A

A.1 Testbed Hardware Information

The WSN nodes used in the experiments throughout this thesis were Berkeley TeloB series nodes. These nodes consist of i) a microprocessor (MSP430 series from Texas Instruments), ii) a IEEE 802.15.4 transceiver (CC2420 series from Texas Instruments), iii) 1MB external flash storage (ST M25P80 series from Micron). The operating system that ran on the nodes was TinyOS 2.0.1, running using the NesC programming language. The antennas used were omnidirectional antennas with a maximum transmit power of 0dBm (1mW).

The specific physical layer modulation and demodulation implementation used within the broader IEEE 802.15.4 standard was the 250KB/s Orthogonal Quadrature Phase Shift Keying (O-QPSK) which operates around the 2.4GHz channel band.

Bibliography

[1] M. Wilhelm, I. Martinovic, and J. B. Schmitt, "Secure key generation in sensor networks based on frequency-selective channels," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1779–1790, September 2013.

[2] T. Instruments, "CC2420 Datasheet," 2006. [Online]. Available: https://web.archive.org/web/20190419015904/http://www.ti.com/lit/ds/symlink/cc2420.pdf

[3] Memsic, "TelosB datasheet," 2013. [Online]. Available: https://web.archive.org/web/20190215211317/http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb{_}datasheet.pdf

[4] K. Moara-Nkwe, Q. Shi, G. M. Lee, and M. H. Eiza, "A novel physical layer secure key generation and refreshment scheme for wireless sensor networks," *IEEE Access*, vol. 6, pp. 11 374–11 387, 2018.

[5] A. Soni, R. Upadhyay, and A. Kumar, "Wireless physical layer key generation with improved bit disagreement for the internet of things using moving window averaging," *Physical Communication*, vol. 33, pp. 249 – 258, 2019.

[6] G. Li, C. Sun, J. Zhang, E. Jorswieck, B. Xiao, and A. Hu, "Physical layer key generation in 5g and beyond wireless communications: Challenges and opportunities," *Entropy*, vol. 21, no. 5, 2019. [Online]. Available: https://www.mdpi.com/1099-4300/21/5/497

[7] Y. Wu, A. Khisti, C. Xiao, G. Caire, K. Wong, and X. Gao, "A survey of physical layer security techniques for 5g wireless networks and challenges ahead,"

IEEE Journal on Selected Areas in Communications, vol. 36, no. 4, pp. 679–695, 2018.

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

[9] N. Zhang, X. Fang, Y. Wang, S. Wu, H. Wu, D. Kar, and H. Zhang, "Physical layer authentication for internet of things via wfrft-based gaussian tag embedding," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[10] P. Murvay and B. Groza, "Efficient physical layer key agreement for flexray networks," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[11] T. M. Cover and T. A. Joy, *Elements of Information Theory.* Wiley-Interscience, 2006.

[12] A. Goldsmith, *Wireless Communications.* New York, NY, USA: Cambridge University Press, 2005.

[13] W. Ren, F. An, D. Shen, B. Liu, and X. Shan, "Research on statistical characteristics for the part-shadowing channel model," in *2017 Sixth Asia-Pacific Conference on Antennas and Propagation (APCAP)*, 2017, pp. 1–3.

[14] S. Samanta and T. V. Sridha, "Modified slow fading channel estimation technique and fast fading channel estimation technique for ofdm systems," in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 2018, pp. 1638–1643.

[15] A. K. Permana and E. Y. Hamid, "Dft-based channel estimation for gfdm on multipath channels," in *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2018, pp. 31–35.

[16] Y. Yang, D. Fei, and S. Dang, "Inter-vehicle cooperation channel estimation for ieee 802.11p v2i communications," *Journal of Communications and Networks*, vol. 19, no. 3, pp. 227–238, 2017.

[17] H. Friis, "Simple Transmission Formula for radio transmission," *Proceedings of the IRE and Waves and Electrons*, no. 1, pp. 254–256, 1946.

[18] K. K. Mohan, S. S. Das, and P. Ray, "Link adaptation schemes based on parametric estimation of snr distribution over nakagami- m fading channels," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1537–1553, 2019.

[19] A. F. Molisch, *Wireless Communications*, 2nd ed. Wiley Publishing, 2011.

[20] B. Korenev, *Bessel Functions and Their Applications*, ser. Analytical Methods and Special Functions. CRC Press, 2002. [Online]. Available: https://books.google.co.uk/books?id=qy1GNv2ovHQC

[21] M. Mohammadi and A. Keshavarz-Haddad, "A new distributed group key management scheme for wireless sensor networks," in *2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, 2017, pp. 37–41.

[22] P. P, S. S, and N. S, "A survey on dynamic key management system in secure group communication," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 1440–1443.

[23] G. Krishnasamy, "An energy aware fuzzy trust based clustering with group key management in manet multicasting," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, 2019, pp. 1–5.

[24] S. Blackburn, K. Martin, and M. Paterson, "Key Refreshing in Wireless Sensor Networks," in *Proceedings of the 3rd International Conference on Information Theoretic Security*. Berlin, Heidelberg: Springer Science, 2008, pp. 156–170.

[25] X. Zou, B. Ramamurthy, and S. Magliveras, *Secure Group Communications Over Data Networks*. Springer New York, 2007. [Online]. Available: https://books.google.co.uk/books?id=3L0JTidK-IoC

[26] Abdullah Said Alkalbani, T. Mantoro, and A. O. M. Tap, "Comparison between rsa hardware and software implementation for wsns security schemes," in *Proceeding of the 3rd International Conference on Information and Communication Technology for the Moslem World (ICT4M) 2010*, 2010, pp. E84–E89.

[27] V. Patil, V. Kulkarni, and H. Patil, "Improvised group key management protocol for scada system," in *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, 2018, pp. 1–4.

[28] S. Sharma and C. R. Krishna, "An efficient distributed group key management using hierarchical approach with elliptic curve cryptography," in *2015 IEEE International Conference on Computational Intelligence Communication Technology*, 2015, pp. 687–693.

[29] H. Lin, M. Sun, H. Lin, and W. Kuo, "Multi-level and group-based key management for mobile ad hoc networks," in *2012 International Conference on Information Security and Intelligent Control*, 2012, pp. 164–167.

[30] B. Bryant, *Designing an Authentication System: a Dialogue in Four Scenes.* M.I.T., Project Athena, 1988.

[31] D. Liu and P. Ning, "Multi-Level microTESLA: A Broadcast Authentication System for Distributed Sensor Networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 4, pp. 800–836, 2003.

[32] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM conference on Computer and communications security - CCS '02.* New York, New York, USA: ACM Press, 2002, p. 41. [Online]. Available: http://portal.acm.org/citation.cfm?doid=586110.586117

[33] Haowen Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405).* IEEE Comput. Soc, 2003, pp. 197–213. [Online]. Available: http://ieeexplore.ieee.org/document/1199337/

[34] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," in *Advances in Cryptology.* Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 335–338.

[35] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly Secure Key Distribution for Dynamic Conferences," *Information and Computation*, vol. 146, no. 1, pp. 1–23, oct 1998. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0890540198927171

[36] M. Klonowski, M. Kutylowsk, M. Ren, and K. Rybarczyk, "Forward-Secure Key Evolution in Wireless Sensor Networks," in *Cryptology and Network Se-*

curity: 6th International Conference. Berlin, Heidelberg: Springer Science, 2007, pp. 102–120.

[37] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3776–3784, nov 2005. [Online]. Available: http://ieeexplore.ieee.org/document/1528749/

[38] A. Badawy, T. Elfouly, T. Khattab, A. Mohamed, and M. Guizani, "Unleashing the secure potential of the wireless physical layer: Secret key generation methods," *Physical Communication*, vol. 19, pp. 1–10, jun 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1874490715000713

[39] Y. E. H. Shehadeh and D. Hogrefe, "A survey on secret key generation mechanisms on the physical layer in wireless networks," *Security and Communication Networks*, vol. 8, no. 2, pp. 332–341, jan 2015. [Online]. Available: http://doi.wiley.com/10.1002/sec.973

[40] M. McGuire, "Channel Estimation for Secret Key Generation," in *2014 IEEE 28th International Conference on Advanced Information Networking and Applications*. IEEE, may 2014, pp. 490–496. [Online]. Available: http://ieeexplore.ieee.org/document/6838704/

[41] S. N. Premnath, S. Jana, J. Croft, P. L. Gowda, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "Secret Key Extraction from Wireless Signal Strength in Real Environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 917–930, may 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6171198/

[42] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948.

[43] A. D. Wyner, "The wire-tap channel," *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, Oct 1975.

[44] P. Xu, K. Cumanan, Z. Ding, X. Dai, and K. K. Leung, "Group secret key generation in wireless networks: Algorithms and rate optimization," *IEEE*

Transactions on Information Forensics and Security, vol. 11, no. 8, pp. 1831–1846, Aug 2016.

[45] C. Ye and A. Reznik, "Group secret key generation algorithms," in *2007 IEEE International Symposium on Information Theory*, June 2007, pp. 2596–2600.

[46] S. Nitinawarat, C. Ye, A. Barg, P. Narayan, and A. Reznik, "Secret key generation for a pairwise independent network model," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6482–6489, Dec 2010.

[47] H. Liu, J. Yang, Y. Wang, Y. Chen, and C. E. Koksal, "Group secret key generation via received signal strength: Protocols, achievable rates, and implementation," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2820–2835, Dec 2014.

[48] H. Liu, J. Yang, Y. Wang, and Y. Chen, "Collaborative secret key extraction leveraging received signal strength in mobile wireless networks," in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 927–935.

[49] Y. Liu, H. H. Chen, and L. Wang, "Physical layer security for next generation wireless networks: Theories, technologies, and challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 347–376, Firstquarter 2017.

[50] X. Zhou, L. Song, and Y. Zhang, *Physical Layer Security in Wireless Communications*. CRC Press, 2014.

[51] X. Wang, L. Jin, K. Huang, M. Li, and Y. Ming, "Physical layer secret key capacity using correlated wireless channel samples," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[52] K. Zhang, *Wireless Communications: Principles, Theory and Methodology*. Wiley, 2015. [Online]. Available: https://books.google.co.uk/books?id=GGXKCQAAQBAJ

[53] P. Medina, J. R. Gallardo, J. Sanchez, and F. Ramarez-Mireles, "Impact of Delay Spread on IEEE 802.15.4a Networks with Energy Detection Receivers," *Journal of applied research and technology*, vol. 8, pp. 352 – 362, 12 2010.

[54] D. Gay, P. Levis, R. Von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesc language: A holistic approach to networked embedded systems," in *Acm*

Sigplan Notices, vol. 38, no. 5. ACM, 2003, pp. 1–11.

[55] X. Yin and X. Cheng, *Propagation Channel Characterization, Parameter Estimation, and Modeling for Wireless Communications*, ser. Wiley - IEEE. Wiley, 2016. [Online]. Available: https://books.google.co.uk/books?id=IRWuDQAAQBAJ

[56] F. Gravetter and L. Wallnau, *Statistics for The Behavioral Sciences*. Cengage Learning, 2016. [Online]. Available: https://books.google.co.uk/books?id=ZCNTCwAAQBAJ

[57] H. Anderson, *Fixed Broadband Wireless System Design*. Wiley, 2003. [Online]. Available: https://books.google.co.uk/books?id=M8NOGnp2IRwC

[58] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Booz-Allen and Hamilton Inc Mclean Va, Tech. Rep., 2001.

[59] M. Z. n. Zamalloa and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Trans. Sen. Netw.*, vol. 3, no. 2, Jun. 2007. [Online]. Available: http://doi.acm.org/10.1145/1240226.1240227

[60] C. Phillips, D. Sicker, and D. Grunwald, "Bounding the error of path loss models," in *2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2011, pp. 71–82.

[61] W. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge University Press, 2009. [Online]. Available: https://books.google.co.uk/books?id=0gwqxBU_t-QC

[62] T. Instruments and I. Slas, "MSP430F15x, MSP430F16x, MSP430F161x MIXED SIGNAL MICROCONTROLLER," p. 77, 2011. [Online]. Available: https://web.archive.org/web/20181222222602/https://www.ti.com/lit/ds/symlink/msp430f1611.pdf