Secret Key Extraction from Wireless Signal Strength in Real Environments

Sriram N. Premnath, Suman Jana, Jessica Croft, Prarthana L. Gowda, Mike Clark, Sneha K. Kasera, Neal Patwari, Srikanth V. Krishnamurthy

Abstract—We evaluate the effectiveness of secret key extraction, for private communication between two wireless devices, from the received signal strength (RSS) variations on the wireless channel between the two devices. We use real world measurements of RSS in a variety of environments and settings. The results from our experiments with 802.11 based laptops show that (i) in certain environments, due to lack of variations in the wireless channel, the extracted bits have very low entropy making these bits unsuitable for a secret key, (ii) an adversary can cause predictable key generation in these static environments, and (iii) in dynamic scenarios where the two devices are mobile, and/or where there is a significant movement in the environment, high entropy bits are obtained fairly quickly. Building on the strengths of existing secret key extraction approaches, we develop an environment adaptive secret key generation scheme that uses an adaptive lossy quantizer in conjunction with Cascade-based information reconciliation [9] and privacy amplification [15]. Our measurements show that our scheme, in comparison to the existing ones that we evaluate, performs the best in terms of generating high entropy bits at a high bit rate. The secret key bit streams generated by our scheme also pass the randomness tests of the NIST test suite [1] that we conduct. We also build and evaluate the performance of secret key extraction using small, low-power, hand-held devices - Google Nexus One phones - that are equipped 802.11 wireless network cards. Last, we evaluate secret key extraction in a multiple input multiple output (MIMO)-like sensor network testbed that we create using multiple TelosB sensor nodes. We find that our MIMO-like sensor environment produces prohibitively high bit mismatch, which we address using an iterative distillation stage that we add to the key extraction process. Ultimately, we show that the secret key generation rate is increased when multiple sensors are involved in the key extraction process.

Index Terms—wireless networks, multipath fading, physical layer, cryptography, key generation

1 Introduction

Secret key establishment is a fundamental requirement for private communication between two entities. Currently, the most common method for establishing a secret key is by using public key cryptography. However, public key cryptography consumes significant amount of computing resources and power which might not be available in certain scenarios (e.g., sensor networks). More importantly, concerns about the security of public keys in the future have spawned research on methods that do not use public keys. Quantum cryptography [7], [26] is a good example of an innovation that does not use public keys. It uses the laws of Quantum theory, specifically Heisenberg's uncertainty principle, for sharing a secret between two end points. Although quantum cryptography applications have started to appear recently [12], they are still very rare and expensive.

A less expensive and more flexible solution to the problem of sharing secret keys between wireless nodes (say Alice and Bob) is to extract secret bits from the inherently random *spatial and temporal variations* of the *reciprocal wireless channel* between them [6], [20], [18], [5], [24]. Essentially, the radio channel is a time and space-varying filter, that at any point in time has the identical filter response for signals sent from Alice to Bob as for signals sent from Bob to Alice.

Received signal strength (RSS) is a popular statistic of the radio channel and can be used as the source of secret information shared between a transmitter and receiver. We use RSS as a channel statistic, primarily because of the fact that most of the current of-the-shelf wireless cards, without any modification, can measure it on a per frame basis. The variation over time of the RSS, which is caused by motion and multipath fading, can be quantized and used for generating secret keys. The mean RSS value, a somewhat predictable function of distance, must be filtered out of the measured RSS signal to ensure that an attacker cannot use the knowledge of the distance between key establishing entities to guess some portions of the key. These RSS temporal variations,

- *This article extends our conference paper [17].*
- *S.N. Premnath, P.L. Gowda and S.K. Kasera are with the School of Computing, University of Utah, Salt Lake City. E-mail: {nandha, gowda, kasera}@cs.utah.edu*
- *S. Jana is with the Department of Computer Science, University of Texas, Austin. This work was completed while S. Jana was with the School of Computing, University of Utah, Salt Lake City. E-mail: suman@cs.utexas.edu*
- *M. Clark is with the Air Force Research Laboratory. This work was completed while M. Clark was with the School of Computing, University of Utah, Salt Lake City. E-mail: michael.clark2@wpafb.af.mil*
- *J. Croft and N. Patwari are with the Dept. of Electrical and Computer Engineering, University of Utah, Salt Lake City. E-mail: jessica.croft@utah.edu, npatwari@ece.utah.edu*
- *S.V. Krishnamurthy is with the Dept. of Computer Science and Engineering, University of California, Riverside. E-mail: krish@cs.ucr.edu*

Digital Object Indentifier 10.1109/TMC.2012.63

as measured by Alice and Bob, cannot be measured by an eavesdropper (say Eve) from another location unless she is physically very close to Alice or Bob. However, due to non-ideal conditions, including limited capabilities of the wireless hardware, Alice and Bob are unable to obtain identical measurements of the channel. This asymmetry in measurements brings up the challenge of how to make Alice and Bob agree upon the same bits without giving out too much information on the channel that can be used by Eve to recreate secret bits between Alice and Bob.

Azimi-Sadjadi et al. [6] suggested using two well-known techniques from quantum cryptography - *information reconciliation* and *privacy amplification*, to tackle the challenge caused by RSS measurement asymmetry. Information reconciliation techniques (e.g., Cascade [9]) leak out minimal information to correct those bits that do not match at Alice and Bob. Privacy amplification [15] reduces the amount of information the attacker can have about the derived key. This is achieved by letting both Alice and Bob use universal hash functions, chosen at random from a publicly known set of such functions, to transform the reconciled bit stream into a nearly perfect random bit stream.

Most of the previous research work on RSS-based secret key extraction, including that of Azimi-Sadjadi et al. [6], is based on either simulations or theoretical analysis. Other than the recent work by Mathur et al. [20] that was performed in a specific indoor environment, there is very little research on evaluating how effective RSS-based key extraction is in real environments under real settings. We address this important limitation of the existing research in this paper with the help of wide-scale real life measurements in both static and dynamic environments. In order to perform our measurements and subsequent evaluations, we implement different RSS quantization techniques in conjunction with information reconciliation and privacy amplification.

We first collect measurements under different environments to generically evaluate the effectiveness of secret key generation. We find that under certain environments due to lack of variations in the channel, the extracted key bits have very low entropy making these bits unsuitable for a secret key. Interestingly, we also find that an adversary can cause predictable key generation in these static environments. However, in scenarios where Alice and Bob are mobile, and/or where there is a significant movement in the environment, we find that high entropy bits are obtained fairly quickly. Next, building on the strengths of the existing schemes, we develop an environment adaptive secret key generation scheme that uses an adaptive lossy quantizer in conjunction with Cascade-based information reconciliation and privacy amplification. Our measurements show that our scheme performs the best in terms of generating high entropy bits at a high bit rate in comparison to the existing ones that we evaluate. The secret key bit streams generated by our scheme also pass the randomness tests of the NIST test suite [1] that we conduct. We also build and evaluate the performance of secret key extraction using small, low-power, hand-held devices - Google Nexus One phones - that are equipped 802.11 wireless network cards. Finally, we also evaluate secret key extraction in a multiple input multiple output (MIMO)-like sensor network testbed that we create using multiple TelosB sensor nodes. We find that our MIMO-like sensor environment produces prohibitively high bit mismatch, which we address using an iterative distillation stage that we add to the key extraction process. Ultimately, we show that the secret key generation rate is increased when multiple sensors are involved in the key extraction process.

2 Adversary Model

In our adversary model we assume that the adversary Eve can listen to all the communication between Alice and Bob. Eve can also measure both the channels between herself and Alice and Bob at the same time when Alice and Bob measure the channel between themselves for key extraction. We also assume that Eve knows the key extraction algorithm and the values of the parameters used in the algorithm. However, we assume that Eve cannot be very close (less than a few multiples of the wavelength of the radio waves being used [20]) to either Alice or Bob while they are extracting their shared key. This will ensure that Eve measures a different, uncorrelated radio channel [11]. We assume that Eve can neither jam the communication channel between Alice and Bob nor can she modify any messages exchanged between Alice and Bob. Essentially, Eve is not interested in disrupting the key establishment between Alice and Bob. However, in our model Eve is free to move intermediate objects between Alice and Bob and affect their communication channel although we assume that Eve is unable to restrict other movements in the channel and thus will not be able to significantly increase the coherence time of the channel. We also assume that Eve cannot cause a person-in-the-middle attack, i.e., our methodology does not authenticate Alice or Bob. In other words, our proposed scheme works against passive adversaries. Even without an authentication mechanism, the Diffie-Hellman secret key establishment scheme has found widespread use in network security protocols and standards (e.g., for providing Perfect Forward Secrecy, Strong password protocols, etc.). We expect that our scheme will provide a strong alternative to the Diffie-Hellman scheme in wireless networks. There is a growing amount of work in authenticating wireless devices based on their physical and radiometric properties (e.g., [10], [16]). These and future authentication mechanisms can be used in conjunction with our secret key establishment scheme.

3 Methodology

In this section, we first describe the three components of our wireless RSS-based secret key extraction. Next,

we briefly describe two classes of existing quantization approaches. Last, we develop a new approach by combining the advantages of the existing approaches.

3.1 Components of RSS-Based Secret Key Extraction

To establish a shared secret key, Alice and Bob measure the variations of the wireless channel between them across time by sending probes to each other and measuring the RSS values of the probes. Ideally, both Alice and Bob should measure the RSS values at the same time. However, typical commercial wireless transceivers are half duplex, i.e., they cannot both transmit and receive the signals simultaneously. Thus, Alice and Bob must measure the radio channel in one direction at a time. However, as long as the time between two directional channel measurements is much smaller than the inverse of the rate of change of the channel, they will have similar RSS estimates.

Most of the existing literature on key extraction from RSS measurements either use some or all of the following three steps:

3.1.1 Quantization

As multiple packets are exchanged between Alice and Bob, each of them builds a time series of measured RSS. Then, each node quantizes its time series to generate an initial secret bit sequence. The quantization is done based on specified thresholds. Figure 1 shows a sample RSS quantizer with two thresholds. Different quantizers have been proposed in the existing literature [5], [6], [20], [24]. The difference in these quantizers mainly results from their different choices of thresholds and the different number of thresholds that they use.

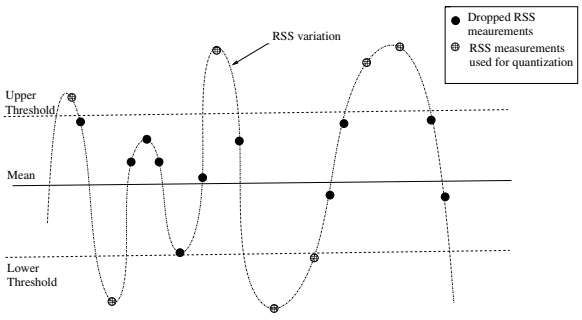

Fig. 1. A sample RSS quantizer. The values between the lower and upper threshold are dropped, the value greater than the upper threshold is encoded as 1 and the value less than the lower threshold is encoded as 0. In this example, the quantizer will output 1010011.

3.1.2 Information Reconciliation

Once both Alice and Bob extract the bit stream from the RSS measurements they collect using quantizers, to agree upon the same key, they must correct the bits where the two bit streams differ. Differences in their bit streams primarily arise due to the following - presence of noise and interference, hardware limitations, manufacturing variations, vendor-specific differences including differences in implementing automatic gain control, and the lack of sampling at the same time at Alice and Bob, primarily due to the half-duplex mode of communication in commercial transceivers.

Cascade [9] is an iterative, interactive information reconciliation protocol. In this protocol, Alice permutes the bitstream randomly, divides it into small blocks and sends permutation and parity information of each block to Bob. Bob permutes his bitstream in the same way, divides it into small blocks, computes parities and checks for parity mismatches. For each mismatch, Bob performs a binary search on the block to find if a few bits can be changed to make the block match the parity. These steps are iterated a number of times to ensure a high probability of success.

3.1.3 Privacy Amplification

When the probe packets are exchanged at a rate greater than the inverse of the coherence interval of the channel, there may be short-term correlation between subsequent quantized bits. Moreover, the information reconciliation stage reveals a certain fraction of information to correct the mismatching bits of Alice and Bob; the leaked portion needs to be removed so that an adversary cannot use this information to guess portions of the extracted key. Privacy amplification solves the above two problems by reducing the size of output bit stream. This is achieved by letting both Alice and Bob use universal hash functions, chosen at random from a publicly known set of such functions, to obtain fixed size smaller length output from longer input streams. Essentially, privacy amplification generates a shorter secret bit stream with a higher entropy rate from a longer secret bit stream with a lower entropy rate. Most of the popular methods used for privacy amplification are based on the *leftover hash lemma*, a well known technique to extract randomness from imperfect random sources [15]. We implement this technique in this paper.

3.2 Existing Approaches

We classify the existing approaches into the following two categories:

Lossy-quantization-based approach: In this approach, bits extracted from the RSS measurements are dropped probabilistically to maintain a high bit entropy. This approach does not use privacy amplification. The goal of this approach is to output a high entropy bit stream so that the output bit stream can be used directly as the shared secret key. This approach has a low output bit rate. Examples of this approach include quantization methods of Aono et al. [5], Tope et al. [24] and Mathur et al. [20].

Lossless-quantization-based approach: This approach does not drop any bits but uses privacy amplification

to increase the bit entropy. This approach produces a high rate output bit stream (e.g., Azimi-Sadjadi et al.'s method [6]).

Note that quantization is inherently lossy. However, in this paper lossless quantization corresponds to obtaining 1 bit or more per sample and lossy quantization corresponds to obtaining less than 1 bit per sample. Also note that we compare these different approaches for the quality of the bit streams they generate. This quality is quantified by three performance metrics -

1) **Entropy:** Entropy characterizes the uncertainty associated with a random variable. We estimate the entropy of a bit stream using NIST test suite's *approximate entropy test* [1].
2) **Bit mismatch rate:** We define the bit mismatch rate as the ratio of the number of bits that do not match between Alice and Bob to the number of bits extracted from RSS quantization.
3) **Secret bit rate:** We define secret bit rate as the average number of secret bits extracted per collected measurement. This rate is measured in terms of final output bits produced after taking care of bit losses due to information reconciliation and privacy amplification.

Note that the bit mismatch rate value we calculate is based on the bits we obtain immediately after the quantization step, and not after the privacy amplification step. In fact, the bit mismatch rate is expected to be zero after the information reconciliation step.

3.3 Adaptive Secret Bit Generation (ASBG)

Our experimental results in Section 5 suggest that some lossy quantizers like Aono et al.'s quantizer or Tope et al.'s quantizer that aim to achieve high bit rate can output bit streams with low entropy in certain settings, especially in those that have minimal movement. On the other hand, some other lossy quantizers like Mathur et al.'s quantizer, can output bit streams with reasonably high entropy but sacrifice the bit rate to achieve this or vice versa. The lossless quantizer described above also generates secret bits at a low rate. In summary, the existing approaches that use RSS measurements do not generate secret bits at a high rate and/or with high entropy. We develop a method, that we call Adaptive Secret Bit Generation (ASBG), that builds on the strengths of the existing approaches. In our method, we use a modified version of Mathur's quantizer [20] in conjunction with two well-known information reconciliation and privacy amplification techniques.

We first describe our quantizer and then identify the differences with Mathur's scheme. Our modified quantizer is described as follows. (i) Alice and Bob consider a block of consecutive measurements of size *block_size* which is a configurable parameter[1]. For each block, they calculate two adaptive thresholds $q+$ and $q-$ independently such that $q+ = mean + \alpha * std\_deviation$ and $q- = mean - \alpha * std\_deviation$, where $\alpha \geq 0$. (ii) Alice and Bob parse their RSS measurements and drop RSS estimates that lie between $q+$ and $q-$ and maintain a list of indices to track the RSS estimates that are dropped. They exchange their list of dropped RSS estimates and only keep the ones that they both decide not to drop. (iii) Alice and Bob generate their bit streams by extracting a 1 or a 0 for each RSS estimate if the estimate lies above $q+$ or below $q-$, respectively.

Our modified quantizer divides the RSS measurements into smaller blocks of size *block_size* and calculates the thresholds for each block separately. The adaptive thresholds allows our quantizer to adapt to slow shifts of RSS. Mathur et al. [20] subtract a running windowed average of RSS measurements before computing thresholds $q+$ and $q-$ to make their scheme adaptive to the slow variations of RSS. We also perform experiments to find the optimal block size. The results of these experiments are shown in Section 6. Unlike the Mathur quantizer that preserves only a single bit from m consecutive 1s or 0s and drops the other repeating $m-1$ bits, our modified quantizer extracts a bit out of each measurement that falls above the upper threshold or below the lower threshold but depends on the privacy amplification step to remove the effect of correlated bits.

Various single bit quantization methods drop a large amount of RSS samples that lie in between the upper and lower thresholds. These dropped samples constitute a loss of valuable information that can be used by Alice and Bob to generate secret bits and also result in an inefficient utilization of the wireless medium because more probes must be sent and received. Furthermore, privacy amplification also reduces the secret bit rate while increasing entropy. To increase the secret bit rate, we propose an adaptive scheme for extracting multiple bits from a single RSS measurement. Our multiple bit extraction scheme is described as follows.

Once Alice and Bob collect the RSS measurements, they perform the following steps - (i) determine the $Range$ of RSS measurements from the minimum and the maximum measured RSS values, (ii) find N, the number of bits that can be extracted per measurement, where $N \leq \lfloor \log_2 Range \rfloor$, (iii) divide the $Range$ into $M = 2^N$ equal sized intervals, (iv) choose an N bit assignment for each of the M intervals (for example use the Gray code sequence [30]), and (v) for each RSS measurement, extract N bits depending on the interval in which the RSS measurement lies. After completing the above steps, as in the single bit extraction case, Alice and Bob use information reconciliation to correct the mismatching bits, and finally, apply privacy amplification to the reconciled bit stream and extract a high entropy bit stream.

Our results, as presented in Section 6, show that our single bit extraction in conjunction with information reconciliation and privacy amplification is able to achieve higher entropy in comparison to existing schemes, and

1. The Cascade block size is not related to the *block_size* we use for determining the quantization thresholds.

our multiple bit enhancement (evaluated in Section 7) allows us to significantly increase the secret bit rate as well.

4 Implementation

We implement the key extraction scheme consisting of three components, namely quantization, information reconciliation, and privacy amplification, on two laptops (Alice and Bob) equipped with in-built Intel PRO/Wireless 3945ABG wireless network cards, operating in the 802.11g mode. Both laptops run the Ubuntu Linux operating system. In order to establish a secret key, Alice and Bob exchange probe packets periodically and use these probe packets to measure the RSS values.

In our implementation, we use specially crafted 802.11 management frames as probe packets. We prefer to use management frames as a communication mechanism over standard data frames because in the case of data frames, acknowledgement frames are sent by the receiving wireless card. On the other hand, in the case of management frames, no acknowledgement frame is sent by the receiving wireless card. Moreover, management frames are prioritized over data frames and are queued separately. These facts motivate us to design our own acknowledgement scheme using management frames instead of data frames to better control the probing rate. In our implementation, among the different management frames, we choose to use the *beacon* frames for the communication between the initiator and the responder. The sequence number field of beacon packet is used as our protocol's sequence number to handle packet loss and retransmissions. We use raw packet injection in the *monitor* mode to send these specially crafted beacon frames. We utilize *ipwraw* [2], a wireless card driver for Intel 3945 cards, for raw packet injection. We also use the monitor mode to receive the beacon frames. In any other mode (e.g., the AP, or STA mode), the wireless device driver does not forward these frames to any upper layer applications. In our implementation, the endpoints exchange beacon frames at a rate of approximately 20 frames per second, and measure the RSS values on a per-frame basis. The RSS measurements we collect are reported by ipwraw driver in the radio tap header of each received frame [3].

We implement our key extraction scheme in a modular way so that different methods of performing quantization, information reconciliation or privacy amplification can be put together to build different schemes using the same basic framework. To compare the performance of different quantizers, we implement them as pluggable modules to our key extraction scheme. For privacy amplification, we use the 2-universal hash family of functions. Alice and Bob use these hash functions to generate the output secret bits. For implementing these hash functions, we use the *BigNumber* routines from the OpenSSL library. These routines allow us to treat chunks of 32 bytes from the input bit stream as very large 256 bit numbers. We describe our implementation in a greater detail in our earlier work [17].

For information reconciliation, we implement the well-known interactive Cascade [9] protocol. In Cascade, the information leakage depends on the block size used in each pass. For optimal information leakage the probability of mismatch should be known a priori as the suitable block size can be determined based on the mismatch probability. However, in our case the mismatch probability is variable and unknown. If the selected block size is too small, a large amount of information will be leaked. On the other hand if the block size is too big, very few bit mismatches will be corrected. We address this problem by using two thresholds (one upper and one lower) and choose random block sizes within those thresholds. We find that the amount of leaked information by Cascade when using random block sizes between 50 and 400 is quite close to the optimal information leakage by Cascade when the probability of mismatch is known a priori.

We also use an Atheros based card to evaluate the effect of heterogeneous hardware on the key extraction process. We present the results that we obtain using the Atheros card in Section 5.5.

5 Measurements

In this work, we use the variation of the wireless channel by measuring RSS on a per frame basis. An RSS measurement represents the average of the energy arriving during the preamble sequence. The wireless card drivers report the RSS values as integers, and the calculation of RSS is vendor dependent. For example, Atheros devices report RSS values from -35 dB to -95 dB, Symbol devices report RSS values from -50 dB to -100 dB, in 10 dB steps, and Cisco devices report RSS values in the range -10 dB to -113 dB [4]. Each of our RSS measurements is quantized into one or more bits for secret key extraction.

We conduct eight experiments in a variety of environments that are classified into three categories - (i) stationary endpoints and stationary intermediate objects, (ii) mobile endpoints, and (iii) stationary endpoints and mobile intermediate objects. We refer to these three categories henceforth as *stationary*, *mobile* and *intermediate* settings respectively. Due to space limitations, we describe only one representative experiment under each category in this section; a more thorough description of all our experiments is available in [17].

We expect that with increased mobility of either the endpoints or of the objects in the environment, the channel variations become more pronounced. As we will see in Section 6, mobile environments offer higher bit rates, higher entropy and fewer bit mismatch rates across all the quantization schemes. We show that secret key extraction can work with reasonable efficiency even when Alice and Bob use wireless cards from two different vendors, despite the differences in the manner in

which the RSS values are calculated by each vendor. Very interestingly, we also show that static environments can be exploited by an adversary to cause predictable key generation.

5.1 Stationary Endpoints and Intermediate Objects

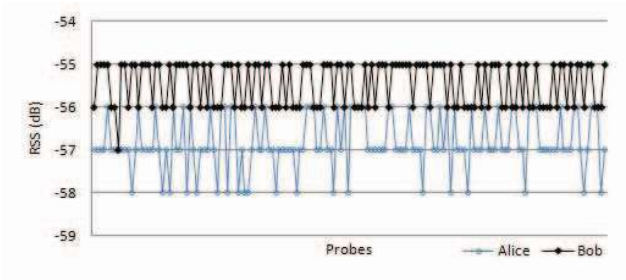

Fig. 2. Underground Concrete Tunnel Measurements

We perform our first experiment inside an underground concrete tunnel that runs between two Engineering buildings inside the University of Utah campus. The concrete tunnel provides an environment that is free from most of the external interference sources, and the effects of mobility of any objects in the environment. Therefore, even though this is an atypical environment, it provides us the opportunity to study the amount of channel variation observed in a completely stationary environment. The two laptops are separated by a distance of about 10 feet during the experiment. Figure 2 shows the variations in RSS measurements collected by Alice and Bob. As expected, there are not much noticeable variations in the channel - at each instant the RSS values vary only as much as 2 dB from the mean. We also note that the curves for Alice and Bob do not follow each other indicating a channel with low reciprocity. This happens because the variations in a static channel are primarily generated by hardware imperfections and thermal effects which are non-reciprocal. RSS measurements in this type of environment contain very low inherent entropy. Therefore, it is not possible to extract secret bits at a fast rate in this type of setting. In fact, using our measurements, we find that it would take 7-8 minutes to generate a 256 bit secret key in this environment.

5.2 Mobile Endpoints

To examine the effect of mobility of nodes in indoor environments, we carry around two laptops at normal walking speed on the third floor of an Engineering Building and record the RSS measurements. The laptops are carried along the corridors in the third floor in such a way that one trails the other and are separated by a distance of $10-15$ feet for the most part. Figure 3 depicts the variations in RSS values measured by Alice and Bob. As we can clearly observe, the channel varies often with a wide variation window (-49 dB to -73 dB) and with

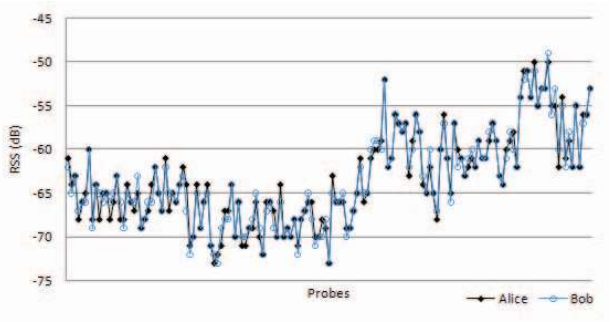

Fig. 3. Measurements while walking inside an Engineering Building

a high degree of reciprocity. This experiment shows that mobility in indoor settings can help achieve fast secret key extraction from RSS measurements by increasing the inherent entropy of the measurements and by improving the reciprocity of the channel.

5.3 Mobile Intermediate Objects

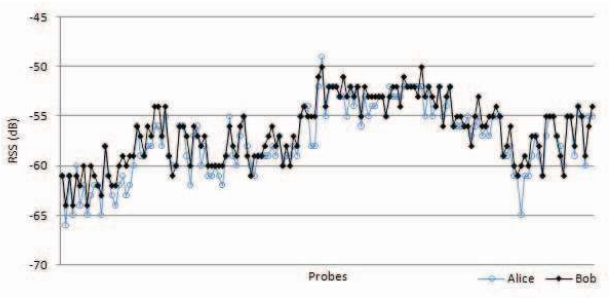

Fig. 4. Crowded Cafeteria Measurements

As we find from our other experiments that mobile nodes result in a variable and highly reciprocal channel, we expect to observe similar effects if we have mobile intermediate objects in the environment instead of the nodes moving themselves. To verify this, we first perform an experiment where we study the effects of randomly moving intermediate objects at low speed. We conduct this experiment during a busy lunch hour in a crowded cafeteria. We keep our laptops stationary on two tables separated by a distance of 10 feet across the main entrance of the cafeteria. In this setting, we see many people frequently walk between these two tables. The channel variations measured by Alice and Bob are shown in Figure 4. As expected, even though the laptops are stationary, the random movements of people produces enough reciprocal channel variations. However, the range of measurements is smaller in comparison to settings with mobile endpoints, but much larger in comparison to stationary settings.

5.4 Predictable Channel Attack

As mentioned earlier, stationary environments cannot support fast secret key extraction. However, another

significant drawback of stationary environments is that an adversary can use planned movements in such environments causing desired and predictable changes in the channel between the actual sender and receiver nodes.

We show that the adversary can, in fact, cause desired changes in the channel between the sender and receiver by controlling the movements of some intermediate object or of the actual radios. We conduct an experiment in a student lab in one of the Engineering buildings with two laptops; the separation between the two laptops is about 10 feet and the intermediate object is moved at about the halfway point in between the laptops.

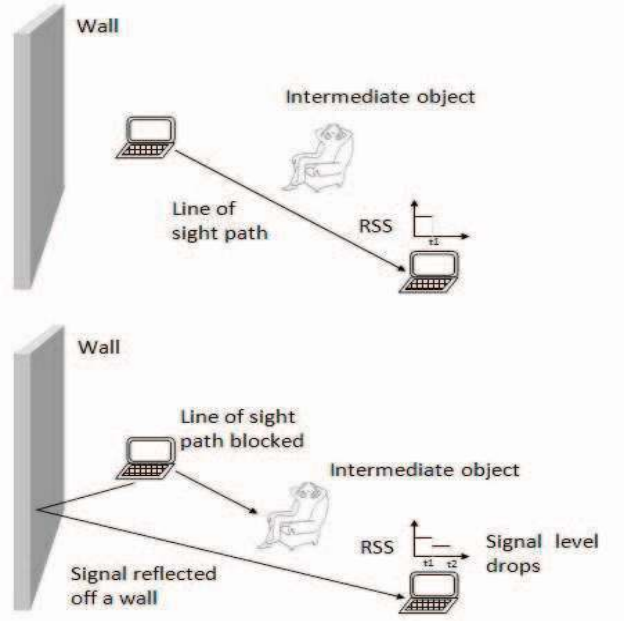

Fig. 5. Schematic of the attack. In the top portion of this figure, there is a line of sight path. In the bottom portion, the attacker intermittently blocks the line of sight path causing a predictable drop in the RSS values.

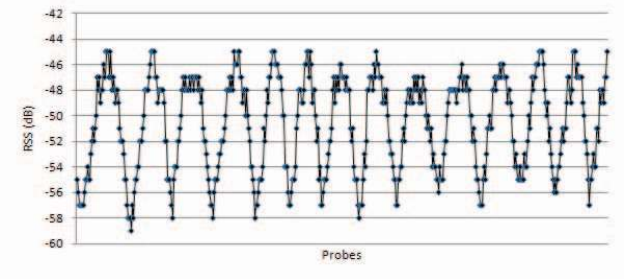

Fig. 6. Predictable variations of the RSS values when an adversary repeatedly blocks and unblocks the line of sight path using an intermediate object.

The schematic of the experiment is shown in Figure 5. One of the authors (say X), sitting on a chair and intermittently leaning backward and forward, takes the role of the intermediate object. Sitting on the chair, whenever X leans backward obstructing the line of sight path, the RSS drops and whenever X leans forward so that there is no obstruction along the line of sight path, the RSS regains its original value. Figure 6 shows the variations of the RSS values and the pattern of variation follows the movements of X. Under these circumstances, when any key extraction scheme is used on such a data set, it produces a predictable pattern of secret bits.

For the RSS values shown in Figure 6, our quantization scheme, actually generates an alternating sequence of multiple 0s and 1s, e.g., 0000111100001111 . . .. Alice and Bob could possibly use random sub-sampling of the bit sequence, as in [20], or use privacy amplification, to ensure that the resulting bit pattern is random. However, if an adversary is able to completely control the bit sequence coming out of the quantization process, then no post-processing technique will be able to ensure the security of the resulting bit sequence. Consequently, it is important to weigh the relationship between the adversary's ability to control the environment and the block size used in sub-sampling or privacy amplification.

It is very important to note that we obtain the above results even with coarse movements, without the use of any precision machinery to create the movements. Thus, our experiments demonstrate that it is quite easy for an adversary to launch a "predictable channel" attack in a stationary environment and cause desired changes in the channel between the sender and receiver making them extract a predictable sequence of secret key bits. One of the possible ways to avoid this attack is to use the RSS measurement based secret extraction scheme only in places where multiple moving objects are present so that the attacker's movement alone will not be able to change the channel predictably. The effectiveness of the predictable channel attack on key extraction methods using other channel characteristics (e.g., channel impulse response) will be explored in the future.

5.5 Heterogeneous Devices

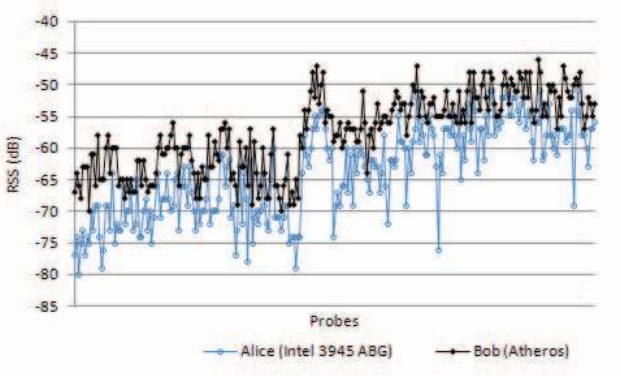

Fig. 7. Measurements from heterogeneous devices while walking inside an Engineering Building

The experiments described so far use identical hardware for both transmitter and receiver. However in reality, different users could have different hardware. To

investigate the effects of using heterogeneous devices, we perform an experiment in a setting similar to that of Section 5.2 (walk inside an Engineering Building). For this experiment, Alice is equipped with an Intel 3945 ABG card and Bob with an Atheros chipset based card. Figure 7 depicts the variations in RSS values measured by Alice and Bob. We can clearly see that even with heterogeneous endpoints, the channel measurements exhibit a very high degree of reciprocity. Alice's RSS values range from -80 dB to -51 dB while Bob's RSS values range from -70 dB to -46 dB. We find that with heterogeneous hardware, when using our quantization method, the mismatch fraction between Alice's and Bob's bit streams is about 11%. In our implementation, information reconciliation can handle this mismatch rate. Therefore, even though heterogeneous hardware introduces higher bit mismatch rates than using homogeneous ones, we can still perform secret key extraction with reasonable efficiency.

6 Comparison of Key Extraction Approaches

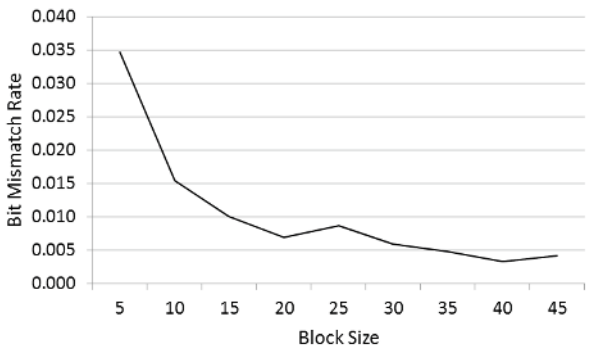

Fig. 8. Variation of Mismatch rate against Block size for ASBG method.

In this section, we compare the performance of ASBG with other existing schemes in terms of entropy, secret bit rate and bit mismatch rate. Although ASBG is capable of multiple bit extraction, we evaluate only single bit extraction in this section. We show that ASBG not only outputs a secret bit stream with the highest entropy but also the secret bit rate and bit mismatch fraction of ASBG are comparable, if not better than all the existing methods.

Various key extraction approaches that we compare in this work use one or more configurable parameters. We choose the parameters for all these quantization schemes such that they help strike a balance between the entropy and the secret bit rate. For the results shown in this section, we use the following configurable parameters. In Aono et al.'s scheme, the configurable parameter β is chosen such that at most 15% of the RSS measurements are deleted from the data set. Tope et al.'s method uses two thresholds - γ_l and γ_h. We choose

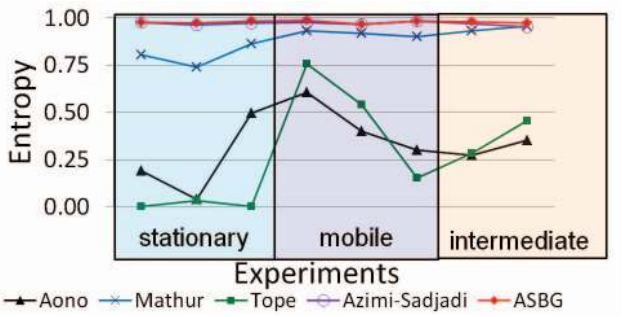

Fig. 9. Entropy comparison between existing quantization schemes and ASBG under various settings.

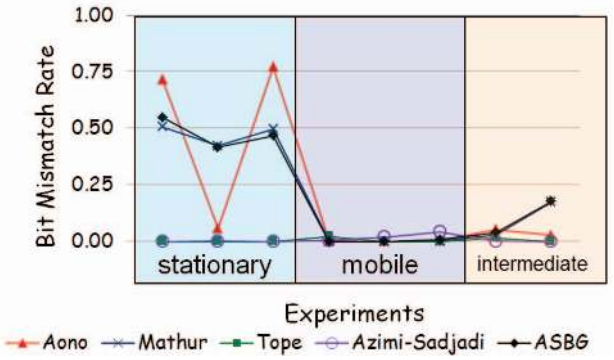

Fig. 10. Bit Mismatch rate comparison

$\gamma_l = avg\_of\_delta\_values + 0.4 * std\_deviation$, and $\gamma_h = avg\_of\_delta\_values + std\_deviation$. In Mathur et al.'s scheme, two thresholds $q+$, $q-$ and m, the minimum number of measurements on an excursion above or below the thresholds, are used such that $q+ = mean + \alpha * standard\_deviation$ and $q- = mean - \alpha * standard\_deviation$. In order to remove the affects of slowly moving average signal power, as suggested in [20], we subtract a windowed average from each RSS measurement. We choose $\alpha = 0.2$ and $m = 2$ to ensure that a large fraction of measurements is considered for bit extraction. We do not implement the random subsampling step because although this step improves the entropy of the extracted bit stream, it negatively impacts the secret bit rate. In Azimi et al.'s scheme, a threshold value of 10 is used to determine the deep fades. When extracting one bit per measurement, ASBG uses two thresholds $q+$, $q-$ with $\alpha = 0.8$ and $block\_size = 25$. Figure 8 shows the variation of the bit mismatch rate with block size for our ASBG scheme. We observe that the mismatch rate gradually falls and becomes very small after a certain block size threshold and stays small even when the block size is increased beyond the threshold. We pick a block size ($= 25$) where the mismatch rate is low.

The performance of the different secret key extraction schemes is shown in Figures 9, 10 and 11. Aono et al.'s scheme has the highest secret bit rate. However, their scheme produces bit streams with very low entropy.

TABLE 1
P-values from NIST statistical test suite results. Experiments $\{A, B, C\} \in$ stationary category, $\{D, E, F\} \in$ mobile category and $\{G, H\} \in$ intermediate category.

Test	A	B	C	D	E	F	G	H
Frequency	0.35	0.03	0.51	0.14	0.51	0.37	0.98	0.95
Block Frequency	0.52	0.57	0.82	0.66	0.38	0.94	0.63	0.03
Cumulative sums(Fwd)	0.46	0.05	0.78	0.19	0.34	0.68	0.55	0.18
Cumulative sums (Rev)	0.27	0.03	0.46	0.09	0.89	0.39	0.52	0.21
Runs	0.21	0.54	0.74	0.41	0.74	0.38	0.55	0.07
longest run of ones	0.08	0.1	0.49	0.65	0.76	0.4	0.78	0.96
FFT	0.71	0.74	0.28	0.59	0.51	0.52	0.23	0.65
Approx. Entropy	0.06	0.34	0.56	0.67	0.65	0.21	0.55	0.25
Serial	0.84, 0.50	0.40, 0.23	0.84, 0.64	0.50, 0.59	0.50, 0.64	0.43, 0.59	0.60, 0.36	0.16, 0.50

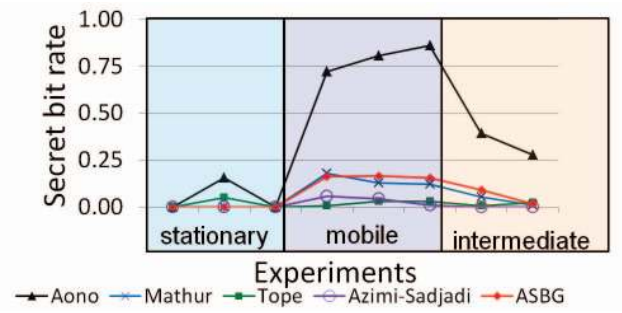

Fig. 11. Secret bit rate comparison between existing quantization schemes and ASBG under various settings.

On the other hand Mathur et al.'s scheme generates bit streams with relatively high entropy at a moderate rate. Note that when random sampling step is employed in Mathur et al.'s scheme, the secret bit rate will be correspondingly lower than what we report in Figure 11. Azimi-Sadjadi et al.'s scheme results in bit streams with highest entropy. However, the bit rate of their scheme is very low. ASBG produces bit streams with highest entropy like Azimi-Sadjadi's scheme while still maintaining the bit rate as high as Mathur et al.'s scheme. In Figure 9, the plots corresponding to Azimi-Sadjadi et al.'s scheme and ASBG are one behind the other.

To ensure the randomness of the bit streams generated by ASBG, we also run randomness tests available in the NIST test suite [1]. There are a total of 16 different statistical tests in the NIST test suite. Of these 16 tests, we run only 8 tests. The bit streams that we obtain from our experiments, meet the input size recommendation [1] of the 8 NIST tests only. We find that the ASBG generated bit streams pass all the 8 tests. The results of these test are shown in Table 1. The remaining 8 tests require a very large input bit stream (specifically, 6 of the 8 remaining tests require $\approx 10^6$ bits). We plan to collect large traces in the future to run these remaining tests.

We briefly describe the purpose of these statistical tests in the NIST test suite [1] as follows. The frequency test determines whether the number of ones and zeros in a sequence are approximately the same. The block frequency test checks whether the frequency of ones in a given M-bit block is approximately $M/2$. Using numeric values -1 and $+1$ in place of bits 0 and 1, the cumulative sums test determines whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of the cumulative sum for random sequences. The runs test verifies whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. The purpose of the longest run of ones (LRO) test is to determine whether the length of the LRO within the tested sequence is consistent with the length of the LRO that would be expected in a random sequence. The FFT test checks for periodic features that would indicate a deviation from the assumption of randomness. The approximate entropy test compares the frequency of overlapping blocks of two consecutive lengths against the expected result for a random sequence. The serial test determines whether the number of occurrences of the 2^m m-bit overlapping patterns is approximately the same as would be expected for a random sequence.

Each of these statistical tests outputs a P-value; the P-value summarizes the strength of the evidence against the null hypothesis, which corresponds to the sequence being tested is random. P-value denotes the probability that a perfect random number generator would have produced a sequence less random than the input sequence that is tested. For a P-value ≥ 0.01, the sequence is considered as random with a confidence of 99%. Note that all the P-values shown in Table 1 are at least 0.01, which demonstrates that the secret bit streams are in fact random with a very high degree of confidence.

7 Multiple Bit Extraction

In this section, we evaluate the performance of extracting multiple bits from a single RSS sample. The goal here is to find whether or not the extraction of multiple bits from a single RSS sample increases the secret bit rate in comparison to single bit extraction.

In Section 5, we have shown that the measurements from static settings exhibit a very narrow RSS range (for example, only 2 dB variation in the experiment of Section 5.1). Extracting even 2 bits from an RSS sample requires a range of at least 4 dB when RSS is reported in 1 dB steps. Further, in Section 6 we have shown that the mismatch rate in the static settings is as high

as 50%. Attempting to extract multiple bits will cause the mismatch rate to increase further. Therefore, we apply our multiple bit extraction method only to mobile settings that do not suffer from these problems of narrow range and very high mismatch rates.

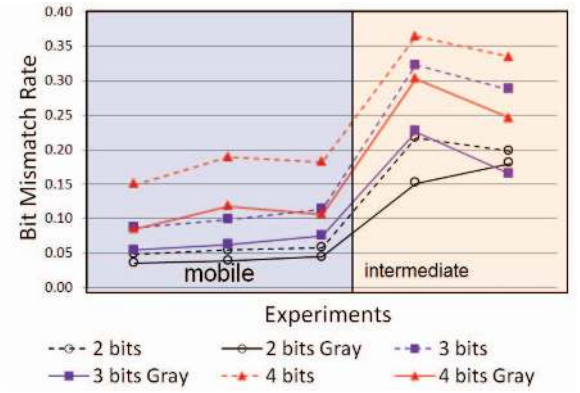

Fig. 12. Bit Mismatch rate comparison

Recall from Section 3 that N is the number of bits extracted per RSS measurement, and M $(= 2^N)$ is the number of equi-sized intervals the RSS range is divided into. Figure 12 shows the mismatch rates for extracting $N = 2-4$ bits respectively from each RSS measurement. Observe that the mismatch fraction increases with N, the number of bits extracted per measurement. Further, the way in which the N bits are assigned to each of the M intervals also affects the mismatch fraction. For example, the use of Gray codes results in a substantially lower mismatch fraction compared to the use of a regular binary sequence as shown in Figure 12. Due to non-perfect channel reciprocity, if an RSS measurement of Alice and that of Bob belong to adjacent intervals, use of Gray codes ensures that the N bits extracted by Alice and Bob differ by at most one bit, where as using a regular binary sequence, causes the bits extracted by Alice and Bob to potentially differ in all the N bits. This accounts for a lower mismatch rate and subsequently higher secret bit rate when using a Gray code sequence.

Figure 13 shows a comparison of secret bit rates for our single and multiple bit extraction methods under various mobile settings. Notice that for the mobile settings, the secret bit rate for single bit extraction is about 16%, whereas for two bits extraction ($N = 2$) using gray coding, the secret bit rate is about 67%. Notably, the secret bit rate of the multiple bit extraction method is at least four times higher than that of the single bit extraction method even when only 2 bits are extracted from each measurement. This substantial improvement accounts for the fact that the single bit extraction method drops all the RSS measurements that lie within the upper and lower thresholds, while the multiple bit extraction method utilizes most of the measurements. Furthermore, similar to our single bit extraction method, the extracted bit streams have an entropy value close to 1 due to privacy amplification. To summarize, the multiple bit quantization scheme substantially improves the secret bit rate in environments with mobile devices.

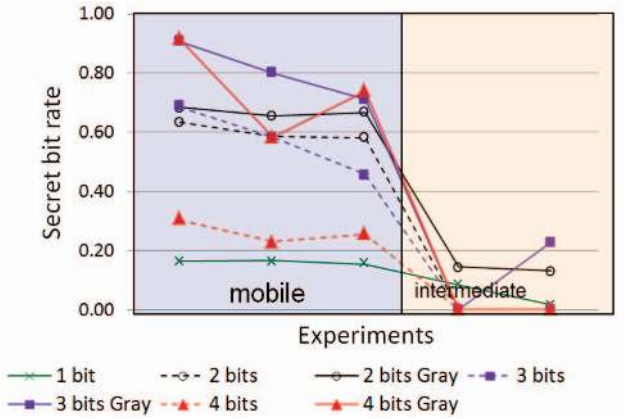

Fig. 13. Secret bit rate comparison when extracting different number of bits under various settings.

8 Secret Key Extraction using Handheld Devices

Given the widespread prevalence of inexpensive and low-power mobile devices, in this section, we evaluate our secret key extraction using two mobile devices, Google Nexus One smartphones, that are equipped with Broadcom BCM 4329 chipset based 802.11 wireless network cards. We first perform experiments similar to the ones described in the previous section in two different environments. Although not shown here, we obtain high entropy secret bits fairly quickly when using these smartphones and our secret bit streams also pass the NISTs approximate entropy test, achieving an entropy value close to the ideal value of one. In the rest of this section, we examine the impact of distances between two smartphones, Alice and Bob, on secret key extraction in two different environments while they transmit at a very low power.

8.1 Experimental Setup

We conduct a number of experiments in the University of Utah campus under two different environments that are changing with time. In each environment, we perform four *walk-experiments* where the phones representing Alice and Bob are carried at normal walking speeds. The average distance (d) in feet between Alice and Bob is varied with each experiment and $d \in \{25, 50, 75, 100\}$.

This first environment is a hallway on the third floor of the Merrill Engineering building. In the experiments conducted in this environment, our phones use the lowest transmit power of 4 dBm.

We conduct a second set of experiments in an outdoor environment across varying terrain, with many trees and bushes in the path between Alice and Bob. Because of the terrain and obstructions in this environment, the path losses are higher. Due to greater path loss in this environment, we use a higher transmit power of 8 dBm.

TABLE 2
Bit Mismatch Rate as a function of Distance.

Distance (feet)	Bit Mismatch Rate (Hallway)	Bit Mismatch Rate (Trees)
25	0.29%	2.46%
50	1.67%	3.60%
75	1.81%	3.63%
100	1.91%	5.29%

TABLE 3
Packet Loss Rate as a function of Distance.

Distance (feet)	Packet Loss Rate (Hallway)	Packet Loss Rate (Trees)
25	1%	1%
50	4%	3%
75	4%	18%
100	9%	27%

8.2 Results

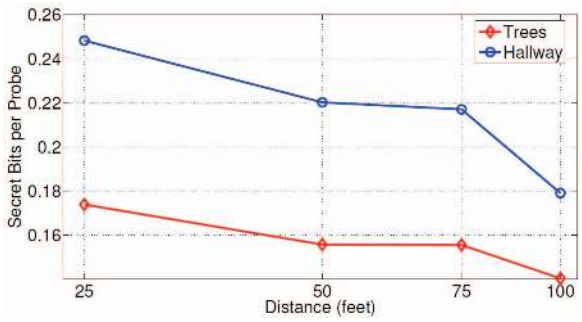

Fig. 14. Secret bits per probe as a function of distance.

In this subsection, we evaluate secret key extraction as a function of distance between Alice and Bob. Our results show that in the hallway environment, even with the lowest transmit power, Alice and Bob can extract about 0.25 secret bits per probe when they are separated by about 25 feet. Figure 14 shows a plot of secret bits per probe as a function of the distance between Alice and Bob. Though we use a lower transmit power in the hallway-environment, in comparison to the trees-environment, the hallway-environment achieves a higher performance due to lower signal attenuation – from our measurements, we find that for a given distance, the average received powers are about $2-7$ dB higher in the hallway environment in comparison to the obstructed outdoor environment. As we show in Figure 14, secret bits per probe decreases with increase in distance, which is attributed to the following reason: As the distance increases the signal-to-noise ratio (SNR) decreases, which consequently increases both the bit mismatch rate (Table 2) and the packet drop probabilities (Table 3); the increase in packet drop further contributes to an increase in the time duration between channel measurements. Nevertheless, on the whole, a comparison of our results in Figure 14 and Figure 11 shows that secret keys can be established efficiently even with low-powered, mobile devices.

9 Secret Key Extraction in MIMO-like Sensor Networks

In the previous sections, we have investigated the effectiveness of secret key generation in various environments using single antenna, single input and single output (SISO) radios available in laptops/smartphones. In order to understand how key extraction applies to sensor nodes, and in a multi-antenna, multiple input multiple output (MIMO) system, we first create a simple, yet flexible, MIMO-like testbed with the help of multiple sensor nodes. Next, we use this testbed to measure RSS, and extract secret keys from RSS variations.

Wallace et al. [25] have recently proposed the use of multiple-input and multiple-output (MIMO) for enhancing secret key extraction. However, their work is an analytical study, presenting only the simulation results. Further, they assume that multiple antennas belong to the same node. However, due to size and power limitations, sensor nodes do not typically have multiple antennas. In this work, we propose to obtain the multi-antenna capability using multiple sensors.

We find that our MIMO-like sensor environment has a much higher bit mismatch rate in comparison to our SISO setup using laptops. To solve this problem, we introduce a *distillation* stage[2] in our key extraction methodology comprising the quantization, information reconciliation, and privacy amplification stages. The distillation stage, introduced between the quantization and the information reconciliation stages, iteratively improves the output from the quantizer by eliminating measurements that are likely to cause mismatching bits at Alice and Bob. This stage ensures that the percentage of mismatching bits is low enough to be handled by information reconciliation without compromising security. In fact, without the distillation stage, the information reconciliation stage by itself is unable to reconcile the bit mismatch.

9.1 Experimental Setup

In our experiments, we use $802.15.4-$compliant Crossbow TelosB wireless sensors for the experiments. As Wilson et al. [27][3] describe, the sensors, which form a token ring, take turn in exchanging probe packets and collecting RSS measurements. We use two sets of five sensors each representing Alice and Bob respectively.

2. The distillation stage as described in this work does not involve any exchange of parity information, and is different from the advantage distillation in quantum cryptography.

3. We thank Joey Wilson for sharing his tinyos program for recording the RSS measurements.

This sensor network platform allows us to readily explore the impact of using multiple antennas on secret key extraction.

For our implementation, we could have possibly used devices equipped with 802.11n wireless cards based on the MIMO technology. However, these off-the-shelf wireless cards typically have $2-3$ pre-installed antennas. In comparison, our MIMO-like configuration allows us to experiment with $1-5$ antennas using the same setup in a flexible manner. Additionally, our platform also allows us to examine RSS-based key extraction in sensor networks.

We conduct our experiment in a student lab. Nodes representing Alice remain stationary in one corner of the lab while the other set of nodes (Bob) is carried around at normal walking speed. The distance between Alice and Bob is maintained between 2 m - 8 m. Nodes of Alice and Bob are arranged in two parallel rows, with each sensor separated from its neighbor by a distance of about 12 cm, which is greater than the de-correlation distance of 6.25 cm for signals transmitted in the 2.4 GHz band. This ensures that the measurements collected at neighboring nodes are mutually uncorrelated. Therefore, we use the secret bit extraction process (shown in Figure 16) separately for each one of the N^2 channels, where N represents the number of nodes at Alice/Bob. We extract two bits from each RSS measurement that we collect in this setup.

9.2 Prohibitively High Bit Mismatch

When using multiple sensors, we find that the bit mismatch rate is significantly higher in comparison to our earlier experiments that use 802.11 single antenna systems. Note that for a mismatch rate of about 22%, the information reconciliation protocol essentially reveals all the bits. So, the collected measurements that exhibit very high bit mismatch are not useful in establishing a secret key.

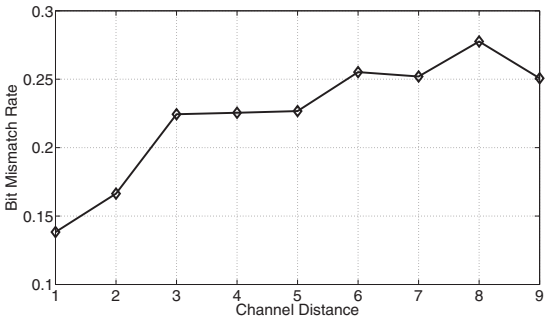

Fig. 15. Bit Mismatch Rate vs Channel Distance

We identify the following reasons for such high mismatch rates. First, when multiple nodes take turn in exchanging probe packets, it increases the average time-gap between any pair of measurements taken in each direction of a channel, and also reduces the probing rate on each channel. Both these factors contribute in increasing the bit mismatch rate. This is also verified in a plot of bit mismatch rate vs channel distance, where channel distance is the absolute difference between the node ids (as defined by the token ring order) of the transmitting and receiving sensors. Figure 15 clearly shows the general increase in mismatch rate with channel distance. Time gap between each unidirectional measurement pairs is proportional to the channel distance. So, mismatch rate increases with channel distance/multiple antennas.

Second, channels in $802.15.4$ are much narrower in comparison to 802.11. A non-reciprocal deep fade (perhaps due to strong interference only at Alice) occurring on a narrow channel significantly reduces the average RSS computed at Alice while not affecting much at Bob. This results in a greater likelihood of asymmetry in measurements, and therefore higher bit mismatch when using narrow channel measurements.

9.3 Distillation

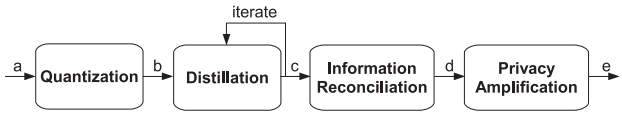

Fig. 16. Secret Bit Extraction Process. a - RSS measurements, b - quantization interval labels, c - distilled bits, d - reconciled bits, e - secret bits.

To address the problem of very high bit mismatch rates, we augment the secret key extraction process with the distillation stage. Distillation ensures that the percentage of mismatching bits is low enough for information reconciliation to correct the differences without revealing all the extracted bits. Figure 16 shows the distillation stage in relation to the other stages of the key extraction process.

Plotting the measurements from channels with large channel distances, we find that a large fraction of consecutive measurements exhibit abrupt transitions from one quantization level to another resulting in asymmetry. The distillation stage seeks to iteratively eliminate such measurements causing abrupt transitions. If the mismatch is still too high even after one round of eliminations, it is necessary to eliminate further; in which case, the next best elimination candidates are those that follow the previously eliminated measurements. When this process is iterated over a number of times, it is guaranteed to improve the bit mismatch rate. Note that the number of iterations required depends on the *current expected mismatch rate* of the channel, which can be determined based on the *history of mismatch rate of the channel*. Algorithm 1 succinctly expresses the steps taken in each iteration. Essentially, in a given block of at least i quantization labels, which are identical (e.g., consecutive a's), iteration number i removes the prefix of length i

Algorithm 1 Distill Input

```
while there is input do
  if current_label = previous_label then
    Output current_label
  else
    Output exclude_label
    previous_label ← current_label
  end if
end while
```

from that block; in case the block length is less than i, it removes the entire block.

Algorithm 1 assumes that the quantizer outputs the labels (e.g., a, b, c, d) of each quantization interval instead of the actual bit pattern assigned to each interval. $exclude\_label$ is a special label indicating an eliminated measurement. In each iteration, the distiller processes the input as shown in Algorithm 1. For the first iteration, the distiller gets its input from the quantizer; and for the successive iterations, the distiller's output becomes the input for the next iteration. In the last iteration, the distiller outputs the bit patterns corresponding to each quantization interval. The following example shows two iterations of distillation; the _ symbol represents the $exclude\_label$.

Distiller Input: $aaaaabbaaaabbbbbaaaa\cdots$
Iteration 1 output: $\_aaaa\_b\_aaa\_bbbb\_aaa\cdots$
Iteration 2 output: $\_\_aaa\_\_\_\_aa\_\_bbb\_\_aa\cdots$

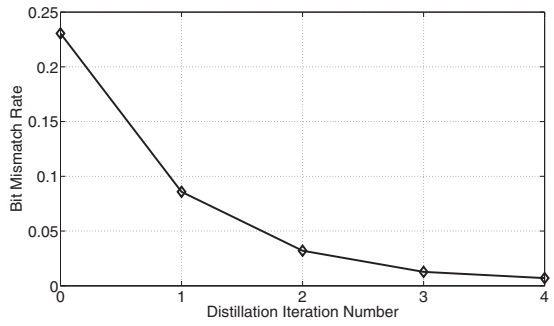

Fig. 17. Effectiveness of distillation in drastically reducing the bit mismatch rate

Figure 17 shows the improvement in bit mismatch rate with each iteration for the 5×5 configuration. Without distillation, the average mismatch rate is about 23%, in which case information reconciliation leaks out all the bits. But two iterations of distillation reduces the mismatch rate to a sufficiently small value ($< 5\%$) for efficient information reconciliation. Thus, despite the simplicity of the distillation approach, these results show that it can reduce the bit mismatch rate very effectively.

9.4 Gain in Secret Bit Rate

Figure 18 shows a plot of the secret bit rate as a function of number of nodes at Alice/Bob. It can be clearly seen

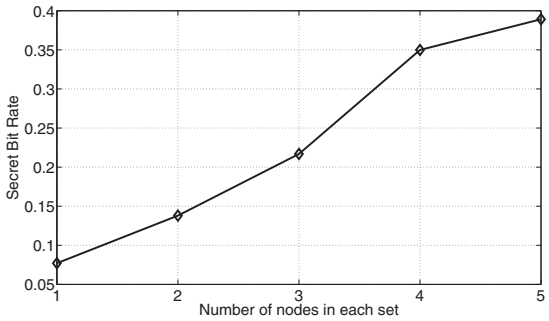

Fig. 18. Secret bit rate vs Number of nodes

that the secret bit rate increases linearly with the number of nodes. We also measure the randomness of the extracted bit streams using NIST's approximate entropy test. We find that the entropy values for the extracted secret bit streams from all the $N \times N$ configurations ($1 \leq N \leq 5$) are close to 1, the ideal value.

10 Related Work

This paper advances the research area [14], [13], [23], [20], [19], [22], [21], [31], [28] of generation of shared secret keys from the observation and processing of radio channel parameters. Amplitude or channel gain is the most common reciprocal channel feature used for secret generation in the literature [6], [18], [30], [5], [24], [20]. Amplitude can be measured more easily than time delay or phase on most existing hardware, and thus is more readily applicable to common wireless networks. In this paper, we similarly use measurements of amplitude, based on their universal availability in wireless networks.

In [28], several bi-directional UWB measurements are made and used to compute the number of secret bits which could be generated. In [18], an implementation using the universal software radio peripheral (USRP) and GNU software radio generates and receives the required multi-carrier signal and evaluates the secret bit rate of the system. In [5], researchers use a steerable directional antenna in combination with Zigbee radio hardware to generate a secret between two nodes and test what an eavesdropper would have received. In [20], Mathur et al. implement two different systems, one using channel impulse response and another using amplitude measurements, to generate secret keys and test how an eavesdropper's measurements differ from the original measurements. Our work differs from Mathur's in the following significant ways. First, we perform extensive real world measurements in a variety of environments and settings to determine the effectiveness of RSS-based secret key extraction. Second, we propose an adaptive secret key extraction scheme that instead of dropping mismatched bits uses information reconciliation to reduce the mismatched bits and also uses privacy amplification. Third, we expose the problem of a predictable channel

attack. Last, we further increase the secret bit rate by extracting multiple bits from each RSS measurement.

Bloch et al. [8] and Ye et al. [29] present an alternative multiple bit extraction scheme that is strongly tied to their use of low-density parity-check (LDPC) based error correction mechanism, which allows them to exploit the correlation between the bits of each sample for error correction. Our work differs from Bloch et al. [8] and Ye et al. [29] in the following ways. First, Bloch et al. conclude that the memory requirements and the complexity of such LDPC based schemes may be too high, especially for low-cost systems, while the cascade [9] based information reconciliation mechanism in our ASBG scheme has very low memory requirements and is much less complex than the LDPC based schemes. Second, these LDPC based schemes rely on *redundant/over-quantized* bits for error correction; they extract M bits from each sample, where M is *at least* $\log_2 K$, and K denotes the number of unique, discrete-valued measurements; in our multiple bit quantization, on the other hand, we extract *at most* $\lfloor \log_2 K \rfloor$ bits from each sample. Hence, in our scheme, we do not extract more bits per sample than what is indicated by the upper bound on the actual information content / entropy present in the measurements, which equals $\log_2 K$. Third, it is possible to calculate the fraction of information that is leaked with cascade for a given bit mismatch rate; and our privacy amplification stage appropriately reduces the output secret key size depending on this fraction of information leakage.

11 Conclusions

We evaluated the effectiveness of secret key extraction from the received signal strength (RSS) variations in wireless channels using extensive real world measurements in a variety of environments and settings. Our experimental results showed that bits extracted in static environments are unsuitable for generating a secret key. We also found that an adversary can cause predictable key generation in static environments. However, bits extracted in dynamic environments showed a much higher secret bit rate. We developed an environment adaptive secret key generation scheme and our measurements showed that our scheme performed the best in terms of generating high entropy bits at a high bit rate in comparison to the existing ones that we evaluated. The secret key bit streams generated by our scheme also passed the randomness tests of the NIST test suite that we conducted. We were able to further enhance the rate of secret bit generation of our scheme by extracting multiple bits from each RSS measurement. We also evaluated secret key extraction in a MIMO-like sensor network testbed and showed that secret key generation rate can be improved by involving multiple sensors in the key extraction process. The conclusions drawn in this paper, specifically the predictable channel attack, are primarily for key extraction using RSS measurements, and these may not directly apply to key extraction using channel impulse response measurements. We would like to explore this in our future work.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. 0855261 and No. 0831490, and by the ONR/ARL MURI Grant W911NF-07-1-0318.

References

[1] NIST, A statistical test suite for random and pseudorandom number generators for cryptographic applications. http://csrc.nist.gov/publications/nistpubs/800-22/sp-800-22-051501.pdf. 2001.
[2] ipwraw, http://homepages.tu-darmstadt.de/ p_larbig/wlan/.
[3] radiotap, http://www.radiotap.org.
[4] Converting signal strength percentage to dbm values, http://www.wildpackets.com/elements/whitepapers/Converting_Signal_Strength.pdf.
[5] 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*, 53(11):3776-3784, Nov. 2005.
[6] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener. Robust key generation from signal envelopes in wireless networks. In *ACM CCS*, 2007.
[7] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin. Experimental quantum cryptography. *J. Cryptol.*, 5(1):3-28, 1992.
[8] M. Bloch, J. Barros, M. R. D. Rodrigues, and S. W. McLaughlin. Wireless information-theoretic security. *IEEE Transactions on Information Theory*, 54(6):2515-2534, 2008.
[9] G. Brassard and L. Salvail. Secret key reconciliation by public discussion. *Lecture Notes in Computer Science*, 765:410-423, 1994.
[10] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *MOBICOM*, 2008.
[11] G. D. Durgin. *Space-Time Wireless Channels*. Prentice Hall PTR, 2002.
[12] L. Greenemeier. Election fix? switzerland tests quantum cryptography. *Scientific American*, October 2007.
[13] A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu. Cryptographic key agreement for mobile radio. *Elsevier Digital Signal Processing*, 6:207–212, 1996.
[14] J. E. Hershey, A. A. Hassan, and R. Yarlagadda. Unconventional cryptographic keying variable management. *IEEE Trans. Commun.*, 43(1):3-6, Jan. 1995.
[15] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *STOC*, pages 12-24, 1989.
[16] S. Jana and S. K. Kasera. On fast and accurate detection of unauthorized access points using clock skews. In *MOBICOM*, 2008.
[17] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *ACM MOBICOM*, 2009.
[18] Z. Li, W. Xu, R. Miller, and W. Trappe. Securing wireless systems via lower layer enforcements. In *ACM WiSe*, 2006.
[19] M. G. Madiseh, M. L. McGuire, S. W. Neville, and A. A. B. Shirazi. Secret key extraction in ultra wideband channels for unsynchronized radios. In *CNSR*, May 2008.
[20] S. Mathur, W. Trappe, N. B. Mandayam, C. Ye, and A. Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *ACM MOBICOM*, 2008.
[21] U. M. Maurer. Secret key agreement by public discussion from common information. *IEEE Trans. Info. Theory*, 39(3):733-742, May 1993.
[22] U. M. Maurer and S. Wolf. Unconditionally secure key agreement and the intrinsic conditional information. *IEEE Trans. Info. Theory*, 45(2):499-514, 1999.
[23] A. Sayeed and A. Perrig. Secure wireless communications: Secret keys through multipath. In *ICASSP*, pages 3013-3016, April 2008.
[24] M. A. Tope and J. C. McEachen. Unconditionally secure communications over fading channels. In *MILCOM*, 2001.

[25] J. W. Wallace, C. Chen, and M. A. Jensen. Key generation exploiting mimo channel evolution: Algorithms and theoretical limits. In *EuCAP*, Mar. 2009.
[26] S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78-88, 1983.
[27] J. Wilson and N. Patwari. Radio tomographic imaging with wireless networks. *IEEE Transactions on Mobile Computing*, 2009.
[28] R. Wilson, D. Tse, and R. A. Scholtz. Channel identification: Secret sharing using reciprocity in UWB channels. *IEEE Transactions on Information Forensics and Security*, 2(3):364-375, Sept. 2007.
[29] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. B. Mandayam. Information-theoretically secret key generation for fading wireless channels. *IEEE Transactions on Information Forensics and Security*, 5(2):240-254, 2010.
[30] C. Ye, A. Reznik, and Y. Shah. Extracting secrecy from jointly gaussian random variables. In *IEEE ISIT*, 2006.
[31] C. Ye, A. Reznik, G. Sternberg, and Y. Shah. On the secrecy capabilities of ITU channels. In *IEEE VTC*, Oct. 2007.

Sriram Nandha Premnath received a B.E. in Computer Science and Engineering from College of Engineering, Guindy, Anna University, India in 2005. He is currently pursuing his Ph.D. in Computer Science from the School of Computing, University of Utah. He has also interned at Bell Labs, Murray Hill. His research interests include network security, wireless networks and Systems.

Suman Jana received the masters degree from the University of Utah. He is currently pursuing his PhD degree in the University of Texas at Austin. His primary research interest is in the field of computer systems security.

Jessica Croft received a B.A. in physics and a B.Music from the University of Montana, Missoula in 2004. She received a M.S.E.E. from Montana Tech, Butte in 2007. She has interned at the Idaho National Lab. She has been pursuing a Ph.D. in electrical engineering at the University of Utah and will graduate in December 2011. She works at Schweitzer Engineering Laboratories.

Prarthana Lakshmane Gowda received a B.E. in Information Science and Engineering from Visvesvaraya Technological University, India in 2009. She is currently pursuing her Masters degree at the School of Computing in the University of Utah. Her primary research interests are in the fields of network security and wireless networks.

Mike Clark received the B.S. degree from Brigham Young University in 2008 and the M.S. degree from the University of Utah in 2010. He is currently a researcher for the Air Force Research Laboratory and pursuing a PhD at the Air Force Institute of Technology. His research interests include cryptography and network security.

Sneha Kumar Kasera is an Associate Professor in the School of Computing at the University of Utah in Salt Lake City. From 1999-2003, he was a member of technical staff in the Mobile Networking Research Department of Bell Laboratories. Earlier, he received a Ph.D. in Computer Science from the University of Massachusetts Amherst, and a Master's degree in Electrical Communication Engineering from the Indian Institute of Science Bangalore. He has held research and development positions at Wipro Infotech and Center for Development of Advanced Computing at Bangalore, India. Dr. Kasera's research interests include computer networks and security. He is a recipient of the 2002 Bell Labs President's Gold Award for his contribution to wireless data research. He has served as the technical program committee chair of the IEEE ICNP and IEEE SECON conferences and as a member of technical program committees of several ACM and IEEE conferences. He serves as a member of the IEEE SECON steering committee and an associate editor for the IEEE/ACM Transactions on Networking and the IEEE Transactions on Mobile Computing. He has also served on the editorial boards of ACM MC2R, ACM/Springer WINET, and Elsevier COMNET journals.

Neal Patwari received the B.S. (1997) and M.S. (1999) degrees from Virginia Tech, and the Ph.D. from the University of Michigan, Ann Arbor (2005), all in Electrical Engineering. He was a research engineer in Motorola Labs, Florida, between 1999 and 2001. Since 2006, he has been at the University of Utah, where he is an Assistant Professor in the Department of Electrical and Computer Engineering, with an adjunct appointment in the School of Computing. He directs the Sensing and Processing Across Networks (SPAN) Lab, which performs research at the intersection of statistical signal processing and wireless networking. He received the NSF CAREER Award in 2008, the 2009 IEEE Signal Processing Society Best Magazine Paper Award, and the 2011 University of Utah Early Career Teaching Award. Neal has served on technical program committees for IEEE conferences SECON, ICDCS, DCOSS, ICC, RTAS, WoWMoM, ICCCN, and MILCOM. He is an associate editor of the *IEEE Transactions on Mobile Computing*.

Srikanth V. Krishnamurthy received his Ph.D degree in electrical and computer engineering from the University of California at San Diego in 1997. From 1998 to 2000, he was a Research Staff Scientist at the Information Sciences Laboratory, HRL Laboratories, LLC, Malibu, CA. Currently, he is a professor of Computer Science at the University of California, Riverside. His research interests are primarily in wireless networks and security. Dr. Krishnamurthy is the recipient of the NSF CAREER Award from ANI in 2003. He was the editor-in-chief for ACM MC2R from 2007 to 2009. He is a Fellow of the IEEE.