On the Effectiveness of Secret Key Extraction from Wireless Signal Strength in Real Environments [*]

Suman Jana[⋆] Sriram Nandha Premnath[⋆] Mike Clark[⋆]
Sneha K. Kasera[⋆] Neal Patwari[†] Srikanth V. Krishnamurthy[‡]

[⋆]School of Computing
University of Utah
{suman, nandha,
mikec, kasera}
@cs.utah.edu

[†]Dept. of Electrical &
Computer Engineering
University of Utah
npatwari@ece.utah.edu

[‡]Dept. of Computer Science &
Engineering
University of California
Riverside
krish@cs.ucr.edu

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. Our experimental results 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 [7] and privacy amplification [14]. 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 [21] that we conduct.

Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—Security and protection

General Terms

Security, Experimentation, Measurement, Performance

Keywords

PHY, Radio Channel, Multipath, RSSI

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 [6, 25] 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 [11], 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 use the inherent randomness in the wireless channel between them as the source for extracting bits of the secret key between these nodes [5, 18, 16, 4, 24]. Central to the secret bit extraction are three properties of transmission and reception of radio signals:

- Reciprocity of radio wave propagation: The multipath properties of the radio channel (gains, phase shifts, and delays) at any point in time are identical on both directions of a link. The reciprocity of radio wave propagation should not be confused with measured received signal strength (RSS) which may be asymmetric.

- Temporal variations in the radio channel: Over time, the multipath channel changes due to movement of either end of the link, and any motion of people and objects in the environment near the link. An application may specifically request a user to move or shake her device in order to generate more temporal variations.

- Spatial variations: The properties of the radio channel are unique to the locations of the two endpoints of the link. An

[*]This research was supported in part by ONR/ARL MURI grant #W911NF-07-1-0318, NSF Career Award #0748206, NSF Cyber Trust Award #0831490, and a seed grant from the University of Utah Research Foundation.

MobiCom'09, September 20–25, 2009, Beijing, China.
Copyright 2009 ACM 978-1-60558-702-8/09/09 ...$5.00.

> eavesdropper at a third location more than a few wavelengths from either endpoint will measure a different, uncorrelated radio channel [10].

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[1]. 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, 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. [5] 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 [7]) leak out minimal information to correct those bits that do not match at Alice and Bob. Privacy amplification [14] 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. [5], is based on either simulations or theoretical analysis. Other than the recent work by Mathur et al. [18] 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 use our implementation to perform 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 [21] that we conduct.

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 [18]) to either Alice or Bob while they are extracting their shared key. This will ensure that Eve measures a different, uncorrelated radio channel [10]. 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., [8, 15]). 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 broadly classify and describe 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, Alice and Bob both should measure the RSS values at the same time. However, typical commercial wireless transceivers are half duplex, i.e., each node can operate their transmitter or receiver, but not both 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 rate of change of the channel, they will have simi-

[1] In this paper, we do not consider any channel impulse response based key extraction [18, 26], and note that our conclusions might not apply to such systems.

lar 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 [4, 5, 18, 24]. The difference in these quantizers mainly results from their different choices of thresholds and the different number of thresholds that they use.

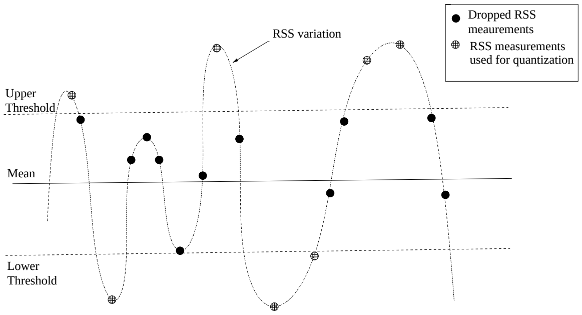
● Dropped RSS meaurements
⊕ RSS measurements used for quantization
RSS variation
Upper Threshold
Mean
Lower Threshold

Figure 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. The differences in the two bit streams arise primarily due to the following factors - 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.

The asymmetry in the bit streams 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 the adversary Eve to recreate secret bits between Alice and Bob. Existing information reconciliation techniques either use error correcting codes [9], or use some interactive information reconciliation protocol. In this paper, we use Cascade [7] which is an iterative interactive information reconciliation protocol. When using Cascade, one party (say Alice) permutes the bit stream randomly, divides it into small blocks and sends permutation and parity information of each block to the other party (say Bob). Bob permutes his bit stream in the same way, divides it into small blocks, calculates and checks if the parity of the blocks are same or not. For each block whose parity does not match, Bob performs a binary search to find if a small number of bits in the block can be changed to make the block match the parity information. These steps are iterated multiple times until the probability of success becomes higher than a desired threshold. As information reconciliation is a probabilistic technique, it might fail occasionally. In those cases the bit streams are discarded and the key extraction process is restarted by measuring RSS values again. However, low failure probability is achieved by suitably choosing the number of passes and the block size in each pass.

3.1.3 Privacy Amplification

In order to obtain independent channel measurements we must probe the channel only once during its coherence time period. Coherence time of a wireless channel is defined as the time during which the channel measurements remain predictable. In real scenarios, it is extremely difficult to estimate coherence time of a channel due to the presence of unpredictable movements caused by objects in the environment. Therefore, in the sampled RSS data, a bit and the subsequent bit may be correlated because of the two corresponding RSS measurements occurring within the coherence time. Thus a bit stream obtained from channel changes can exhibit short-term correlations between subsequent bits. We need a mechanism to minimize the correlation between the bits in a bit stream so that the keys extracted from the bit stream are strong. Moreover, we also need a mechanism to remove portions of the bit stream that are revealed during information reconciliation such that an adversary cannot use this information to guess portions of the extracted key. Privacy amplification solves the above two problems due to the correlation in bits and the revealed information 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. 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 [14]. 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.

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.

We describe specific examples of both these approaches in the following two subsections. *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. The entropy of a random variable X is defined as

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

where X is a set of n symbols $\{x_1, x_2, ...x_n\}$, and, $p(x_i)$ represents the probability of occurrence of the symbol x_i [23]. We estimate the entropy of a bit stream using NIST test suite's *approximate entropy test* [21].

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.

3.2.1 *Lossy-Quantization*

Much of the existing work on RSS-based secret key extraction uses this approach [18, 4, 24]. We describe three of these.

Aono et al.'s quantizer: The quantizer proposed by Aono et al. [4] uses the median value of the RSS measurements as a threshold and drops any measurements that are close to the median value. We run this scheme on our experimental data and find that the output bit streams have high bit mismatch rates and low entropy (as shown in Section 6). Therefore, this scheme is not very effective for secret key sharing.

Tope et al.'s quantizer: Tope et al. have proposed a quantizer that calculates differences in RSS values, denoted by $deltaRSS$, and uses two thresholds to remove those $deltaRSS$ values that are not likely to be similar in Alice's and Bob's measurements. Our experimental results, in Section 6, show that this scheme does not perform well in practice. The output bit streams do not have enough entropy to be useful for generating secret keys.

Mathur et al.'s quantizer: The scheme, proposed by Mathur et al. [18], uses two thresholds $q+$ and $q-$ such that $q+ = mean + \alpha * std\_deviation$ and $q- = mean - \alpha * std\_deviation$, where $0 < \alpha < 1$ and drops the samples with RSS values that are less than $q+$ and greater then $q-$. To reduce bit mismatch rate, this scheme only considers the bit positions which are in the middle of runs of equal length. Furthermore, in this scheme, m consecutive runs of sample values on one side of a threshold are replaced with a single 1 or a single 0. Mathur et al. also propose using random sub-sampling of the extracted key bits to increase the entropy at the expense of the secret bit rate.

3.2.2 *Lossless-Quantization (Azimi-Sadjadi et al.)*

Azimi-Sadjadi et al. have proposed an alternate approach to quantize the RSS values [5]. Their quantizer performs the following two steps. First, it finds the positions of deep fades in the RSS measurements. Next, it encodes the measurements into a bit stream by placing a 0 whenever the measurements are less than the deep fade threshold and a 1 otherwise.

This scheme produces low entropy bit streams at a faster rate. Azimi-Sadjadi et al. suggest using privacy amplification techniques to extract high entropy secret keys from the low entropy bit streams produced by the quantizer. Interestingly, our experiments show that during privacy amplification large portions of the bit stream need to be removed to extract a high entropy bit stream. Therefore, the final secret key generation rate using this scheme is relatively low.

3.3 Adaptive Secret Bit Generation (ASBG)

The results of our experiments, described later 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 [18] 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[2]. 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. [18] 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.

The various single bit quantization methods that we describe above drop a large amount of RSS samples. Specifically, the quantization methods using an upper and a lower threshold drop all the samples that lie in between these 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 [27]), 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

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

extraction in conjunction with information reconciliation and privacy amplification is able to achieve higher entropy in comparison to existing schemes, and 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 RSS. As we noted earlier, commercial 802.11 transceivers support only half-duplex mode of communication. Thus Alice and Bob must measure the RSS values in one direction at a time. To minimize the deviation between Alice's and Bob's measurements, the time difference between their measurement instances should be as small as possible. The measurement process should also overcome packet losses so that the RSS values collected at both ends of the channel can be properly paired. We implement a simple protocol to minimize the time differences between Alice's and Bob's measurements and for dealing with any packet loss during the measurements.

In our protocol, one key establishing party takes the role of an initiator and the other takes the role of the responder. The initiator sends frames to the responder, for which the responder responds with a reply packet. On receiving a packet from the initiator, the responder measures and records the RSS value. Immediately, it sends a reply packet and the initiator measures and records the RSS value. These two measurements collected by the initiator and responder form a pair if there are no packet losses. Packet losses are handled by the initiator. In order to deal with the packet loss and retransmissions, each packet from the initiator carries a sequence number; and the reply packet from the responder has the same sequence number so that the RSS measurements at either end can be paired together. When the responder receives two frames with the same sequence number, it removes the last RSS measurement and records the new RSS measurement.

For implementing our protocol, we choose to use specially crafted 802.11 management frames for communication between the initiator and the responder. 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* [1], 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. We use ipwraw at both the initiator and the responder to implement our protocol. 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 [2].

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 all the quantization schemes described in Section 3 as pluggable modules to our key extraction scheme. For privacy amplification, we use the 2-universal hash family consisting of all the functions $h\colon \{1..M\} \rightarrow \{0,1\}^m$ of the form

$$g_{a,b}(x) = (ax+b) \mod p_M \tag{1}$$

$$h_{a,b}(x) = g_{a,b}(x) \mod m \tag{2}$$

for every $a \in \{1, ..., p_M - 1\}$ and $b \in \{0, ..., p_M - 1\}$. The integer p_M is a prime number with $p_M > M$. For implementation of this hash function family, we use the *BigNumber* routines from the OpenSSL library. The initiator node randomly selects values of a and b and sends them to the responder. The value of p_M is fixed and known to both the initiator and the responder. For our implementation, we use $M = 2^{256}$ and choose m based on the entropy estimate of the bit stream and the value of the allowable deviation of the output from the desired uniform distribution. We divide the input bit stream into blocks of size 32 bytes (*i.e.* 256 bits) and convert these bit streams of 32 bytes into 256 bit long numbers using the *BigNumber* routines. We choose p_M to be a prime number larger than 2^{256}. When the responder receives the values of a and b, both the responder and the initiator calculate the final secret key bits using (1) and (2).

For information reconciliation, we implement the well-known interactive Cascade [7] 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 [3]. Each of our RSS measurements is quantized to produce one or more bits depending on the quantization scheme used, and forms the basis for key extraction.

We conduct our experiments in a wide variety of environmental settings and under different scenarios (with and without mobility of endpoints/intermediate objects, etc.). The environments considered include an underground concrete tunnel, a typical office building, and different outdoor environments. The goal of these experiments is to find answers to the following two questions. First, which settings are better suited for secret bit extraction? Second, which of the approaches described in Section 3 yields better performance by producing bit streams with high entropy, minimal number of mismatched bits between Alice and Bob, and at a fast rate?

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.

In our first three experiments in static environments (Experiments A-C) there is no line of sight between Alice and Bob. In all the other experiments in dynamic environments (Experiments D-H), with several intermediate objects, or the endpoints themselves moving around, the presence or the absence of line of sight changes with time. Except for the predictable channel attack experiments in Section 5.4, our experiments A-H in Sections 5.1 - 5.3 do not include Eve, the adversary.

5.1 Stationary Endpoints and Intermediate Objects

5.1.1 *Underground concrete tunnel (Experiment A)*

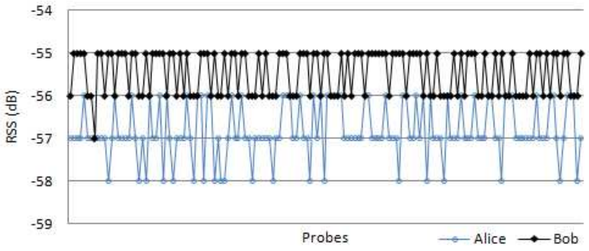

Figure 2: Underground Concrete Tunnel Measurements

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.

5.1.2 *Gallery in the Engineering building (Experiment B)*

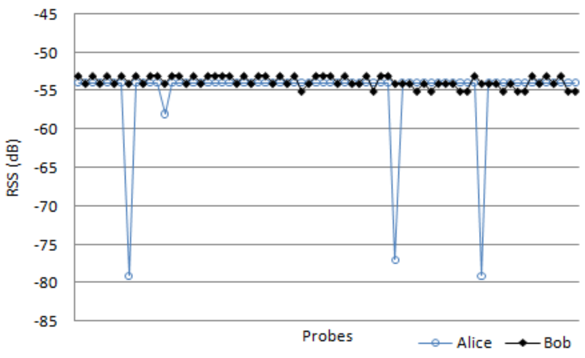

Figure 3: Engineering Building Gallery Measurements

Next, we perform RSS measurements in an indoor setting in one of the Engineering buildings. This experiment is done on a holiday evening to ensure that the gallery is mostly empty and there is minimal external movement. Please note that unlike our experiment in the previous subsection, this setting has normal interference effects caused by other wireless devices operating in the vicinity. This setting allows us to study the channel variations with laptops separated by larger distances (~ 30 feet), in a relatively calm indoor environment. Figure 3 shows the variations in RSS measurements made by Alice and Bob. We find that like our tunnel experiment, Alice's and Bob's measurements are significantly different indicating a very low channel reciprocity. The non reciprocity of the channel is primarily due to the large distance between the laptops. When the distance between Alice and Bob becomes large, the channel measurements are dominated by random thermal noise and different interference sources affecting each laptop differentially. Like the tunnel scenario, it is not possible to extract secret bits at a fast rate in this type of setting as well.

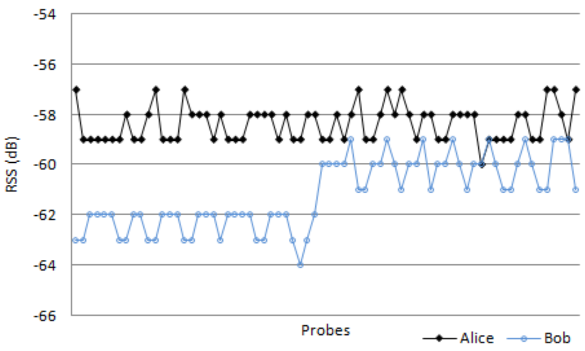

Figure 4: Measurements in the lawn between the Cafeteria and Library

5.1.3 Lawn in between the Cafeteria and Library (Experiment C)

We perform this experiment on a calm, windless day with minimal external movement on a lawn under the trees in between the Cafeteria and the Library. The distance between the laptops is about 10 feet. Figure 4 shows the RSS measurement variations as seen by Alice and Bob, respectively. In this figure, due to the stationary settings, we only find infrequent, small scale variations in the channel measurements. This experiment shows that low-reciprocity is not just a characteristic of the indoor environments, it can occur even in typical stationary outdoor environments. Similar to the first two experiments, this type of setting is also not conducive to fast secret bit extraction and several minutes would be needed to generate a secret key of any reasonable size.

5.2 Mobile Endpoints

5.2.1 Walk inside an Engineering Building (Experiment D)

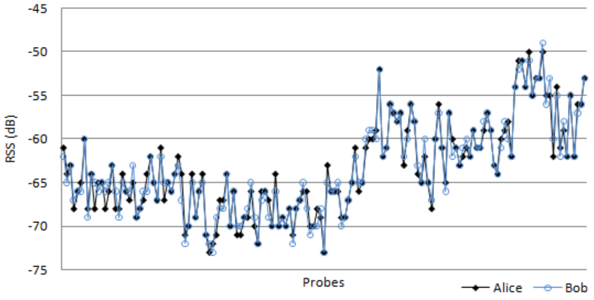

Figure 5: Measurements while walking inside an Engineering Building

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 perform 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[3]. Figure 5 depicts the variations in RSS values measured by Alice and Bob. As we can clearly observe, unlike previous experiments, the channel varies often with a wide variation window (-49 dB to -73 dB) and with a high degree of reciprocity. This experiment shows that mobility in indoor settings helps achieve fast secret key extraction from RSS measurements by increasing the inherent entropy of the measurements and by improving the reciprocity of the channel.

5.2.2 Walk from an Engineering Building to the Cafeteria (Experiment E)

We perform an experiment by carrying two laptops while walking at a normal speed from an Engineering Building to the Cafeteria along two parallel streets. For most part of the experiment, the laptops are separated by a distance of about 20-25 feet. The results of this experiment are shown in Figure 6. As we can see, the measurements show a wide range of variation. The channel variation window is from -49 dB to -76 dB. We also note that like

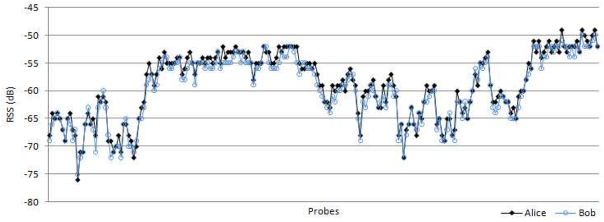

Figure 6: Measurements while walking from an Engineering Building to the Cafeteria

the measurements while walking inside the engineering building, the RSS measurements in this experiment also shows a high degree of reciprocity. This shows that the outdoor environment combined with mobility causes a significant increase in the variation of the channel and improves its reciprocity. Correspondingly, there is a significant increase in the secret bit rate compared to the stationary experiments.

5.2.3 Bike Ride on City streets (Experiment F)

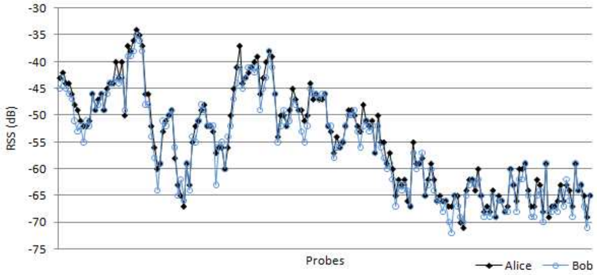

Figure 7: Measurements from slow bike ride on city streets

To evaluate the effect of nodes moving faster than normal walking speeds on the channel variation, we perform an outdoor experiment while we go on a bike tour on city streets. With one bike trailing another, a distance of 10 feet or more is maintained for the most part of the bike ride. As expected, this outdoor experiment exhibits the widest variations (-35 dB to -70 dB) in the channel as shown in Figure 7. The bikes moving at a higher speed compared to walking create an even faster changing channel. As in the previous two cases, this environment also results in a highly reciprocal channel. These two factors together help in achieving a higher secret bit generation rate.

5.3 Mobile Intermediate Objects

5.3.1 Crowded Cafeteria (Experiment G)

As we find in our previous experiments that mobile nodes result in a variable and highly reciprocal channel, we expect to observe similar effects if we have moving intermediate objects in the environment between the nodes 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

[3] Except for the very initial phase of our experiments, and/or when there is intervening traffic in our paths during the experiment, the specified distance is maintained. Also, we do not show the trajectories for any of our experiments due to space limitations.

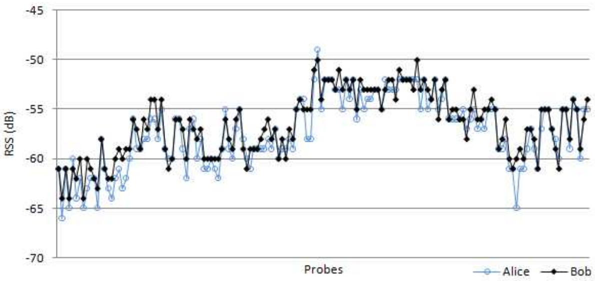

Figure 8: Crowded Cafeteria Measurements

and Bob are shown in Figure 8. As expected, even though the laptops are stationary, the random movements of people in between causes channel variations comparable to the last three experiments with mobile endpoints.

5.3.2 Across a busy road (Experiment H)

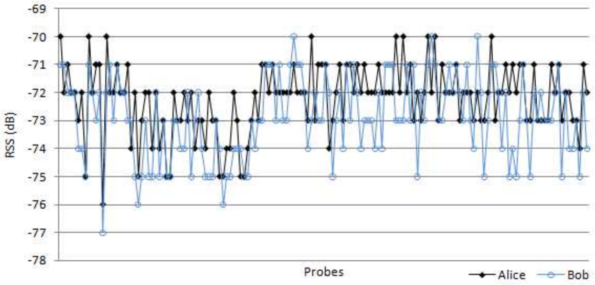

Figure 9: Measurements across a busy road

We perform another experiment to examine the effect of fast moving intermediate objects between two stationary nodes on the RSS measurements. We conduct this experiment across a busy road adjacent to the Engineering building. In this experiment, the vehicles on the road move at high speeds ($\sim$ 30-40 mph). Our laptops are stationary and are separated by a distance of about 25 feet across the road. This environment causes the nodes to experience the highest packet loss rate compared to all the previous experiments. We expect the channel variations to be larger than the previous measurement as the intermediate objects are moving at a faster rate in this case. However, Figure 9 shows that the channel variation window is smaller (-70 dB to -77 dB) than the cafeteria case (Experiment G). Notice that the channel variation and reciprocity in Experiment H are still high compared to the pure stationary environment with a similar distance between the two laptops (Experiment B) and hence will result in secret key extraction at a faster rate.

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 conduct two experiments to 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. The first experiment is conducted 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.

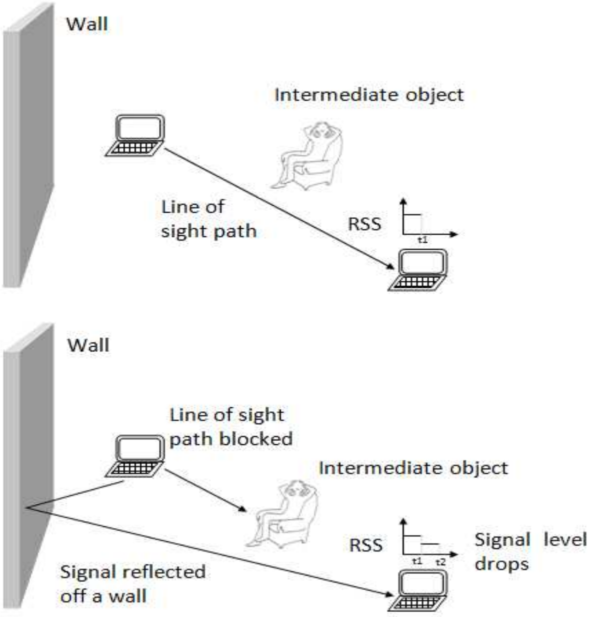

Figure 10: Schematic of the attack. In the top portion of this figure, there is a line of sight path. In the bottom portion, the attacker intermittently blocks the line of sight path causing a predictable drop in the RSS values.

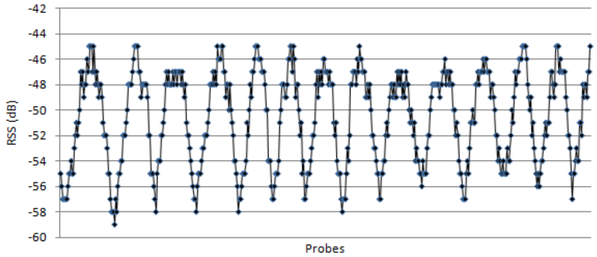

Figure 11: Predictable variations of the RSS values when an adversary repeatedly blocks and unblocks the line of sight path using an intermediate object.

The schematic of the first experiment is shown in Figure 10. 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 11 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 11, our quantization scheme, actually generates an alternating sequence of multiple 0s and 1s, e.g., $0000111100001111\ldots$. Alice and Bob could possibly use random sub-sampling of the bit sequence, as in [18], 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.

In the second experiment, we use a laptop (receiver) and a wireless router (sender) such that they are separated by about 5 feet. The wireless router periodically sends beacon packets that are received by the laptop. While resting the hinges of the laptop on a flat table, we move the laptop back and forth so that the leading edge of its base goes up and down. Again, as in first experiment, the RSS values follow a pattern similar to Figure 11.

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

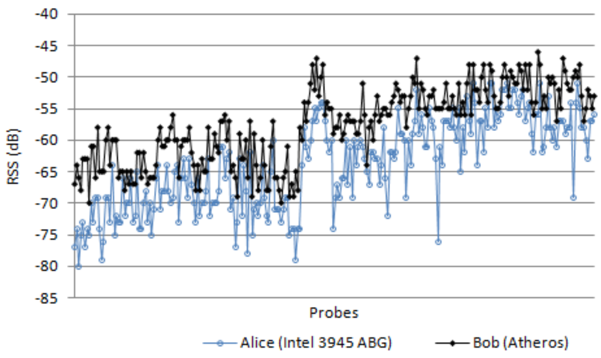

Figure 12: Measurements from heterogeneous devices while walking inside an Engineering Building

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 Experiment D (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 12 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.

5.6 Summary of Measurements

In summary, the environments with stationary endpoints and stationary intermediate objects exhibit small scale variations in the wireless channel. Comparatively, environments with mobile endpoints exhibit a much wider variation in the channel. The small scale variations (for example, -55 dB to -57 dB in Experiment A) in static settings are mainly due to variations in the hardware and random noise. On the other hand, the large scale variations in the mobile settings (for example, -35 dB to -70 dB in Experiment F) are primarily caused by actual changes in the channel. Random noise due to the hardware are also present in the measurements taken in the mobile settings but its effects are not large enough to affect the reciprocity of the channel. Therefore, stationary environmental settings yield much higher bit mismatch rates compared to mobile settings. Further, due to lack of enough variations, static settings also produce bit streams with very low secret bit rates. In short, mobility improves both secret bit rate and bit mismatch rate and hence mobile environments are better suited for the RSS measurement based key extraction schemes.

An adversary can potentially guess the secret key established between the sender and receiver if the adversary, by some means, can affect the channel in a predictable way. Before applying the key extraction methods based on wireless channel characteristics, care must be taken to ensure that there is enough randomness in the environment so that an adversary cannot cause such attacks. One way to ensure this is to force Alice and/or Bob move in a somewhat unpredictable manner while extracting secret keys. Environments including outdoor busy streets and crowded cafeterias, are characterized by unpredictable relative motion between the sender, receiver, and the objects in the environment. These environments are most suitable for key extraction based on reciprocal and dynamic wireless channels.

6. COMPARISON OF KEY EXTRACTION APPROACHES

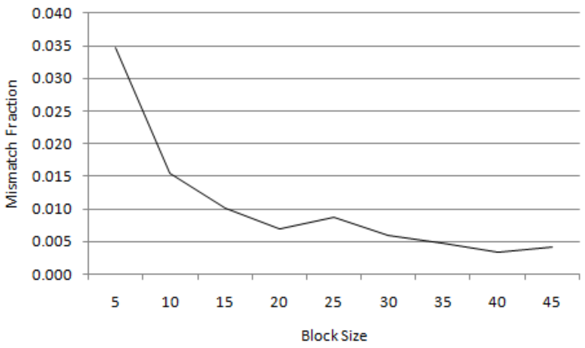

Figure 13: Variation of Mismatch rate against Block size for ASBG method.

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,

Table 1: NIST statistical test suite results. The p value from each test is listed below. To pass a test, the p value for that test must be greater than 0.01.

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

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.

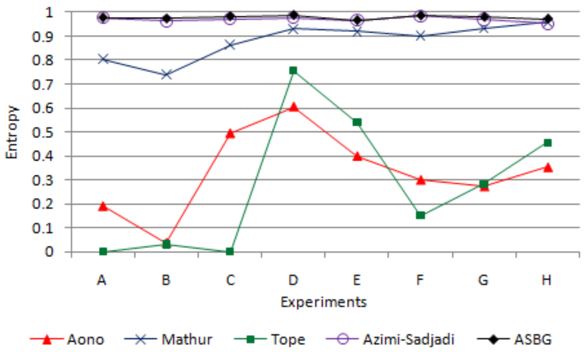

Figure 14: Entropy comparison between existing quantization schemes and ASBG under various settings.

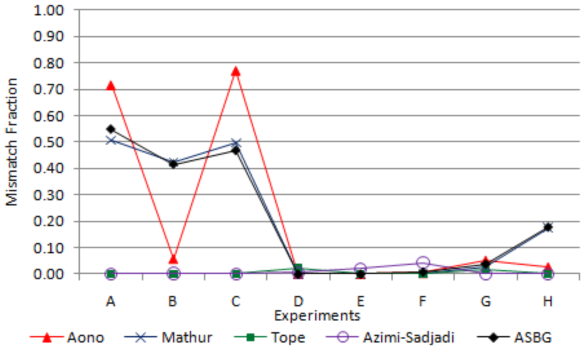

Figure 15: Bit Mismatch rate comparison

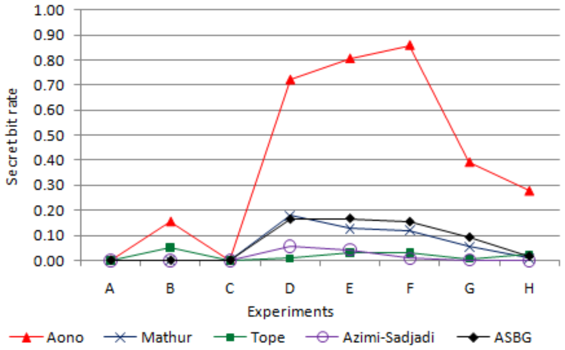

Figure 16: Secret bit rate comparison between existing quantization schemes and ASBG under various settings.

The key extraction approaches, especially the quantization approaches, described in Section 3, 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, in each experiment (A to H) 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 $\gamma_l = avg\_of\_delta\_values + 0.4 * std\_deviation$, 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 [18], 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 sub-sampling 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 13 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 14, 15 and 16. Aono et al.'s scheme has the highest secret bit rate. However, their scheme produces bit streams with very low entropy. 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 16. 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 14, 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 [21]. 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 [21] 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.

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 Experiment A). 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.

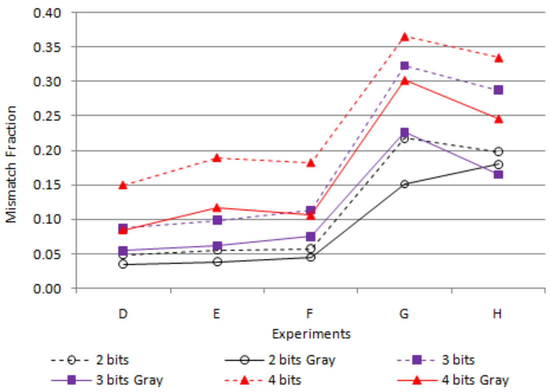

Figure 17: Bit Mismatch rate comparison

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 17 shows the mismatch rates for extracting $N = 1 - 4$ 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 17. 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.

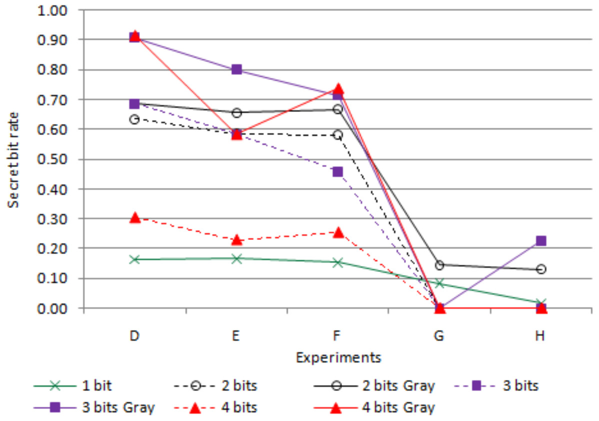

Figure 18: Secret bit rate comparison when extracting different number of bits under various settings.

Figure 18 shows a comparison of secret bit rates for our single and multiple bit extraction methods under various mobile settings (Experiments D-H). Notice that for the experiments D-F, 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.

8. RELATED WORK

This paper advances the research area [13, 12, 22, 18, 17, 20, 19, 28, 26] 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 [5, 16, 27, 4, 24, 18]. 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 [26], several bi-directional UWB measurements are made and used to compute the number of secret bits which could be generated. In [16], 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 [4], 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 [18], 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.

9. 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. 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. We also plan to implement our scheme on a variety of handhelds with different wireless cards.

10. REFERENCES

[1] http://homepages.tu-darmstadt.de/p̃_larbig/wlan/.

[2] http://www.radiotap.org.

[3] http://www.wildpackets.com/elements/whitepapers/Converting_Signal_Strength.pdf.

[4] 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.

[5] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener. Robust key generation from signal envelopes in wireless networks. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 401–410, Nov. 2007.

[6] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin. Experimental quantum cryptography. *J. Cryptol.*, 5(1):3–28, 1992.

[7] G. Brassard and L. Salvail. Secret key reconciliation by public discussion. *Lecture Notes in Computer Science*, 765:410–423, 1994.

[8] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. PARADIS: Wireless device identification with radiometric signatures. In *ACM MOBICOM Conference*, Sept. 2008.

[9] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT*, pages 523–540, 2004.

[10] G. D. Durgin. *Space-Time Wireless Channels*. Prentice Hall PTR, 2002.

[11] L. Greenemeier. Election Fix? Switzerland Tests Quantum Cryptography. *Scientific American*, October 2007.

[12] A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu. Cryptographic key agreement for mobile radio. *Digital Signal Processing*, 6:207–212, 1996.

[13] J. E. Hershey, A. A. Hassan, and R. Yarlagadda. Unconventional cryptographic keying variable management. *IEEE Trans. Commun.*, 43(1):3–6, Jan. 1995.

[14] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *STOC*, 1989.

[15] S. Jana and S. K. Kasera. On fast and accurate detection of unauthorized access points using clock skews. In *ACM MOBICOM Conference*, Sept. 2008.

[16] Z. Li, W. Xu, R. Miller, and W. Trappe. Securing wireless systems via lower layer enforcements. In *Proc. 5th ACM Workshop on Wireless Security (WiSe'06)*, pages 33–42, Sept. 2006.

[17] 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.

[18] 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 Conference*, Sept. 2008.

[19] U. M. Maurer. Secret key agreement by public discussion from common information. *IEEE Trans. Info. Theory*, 39(3):733–742, May 1993.

[20] U. M. Maurer and S. Wolf. Unconditionally secure key agreement and the intrinsic conditional information. *IEEE Trans. Info. Theory*, 45(2):499–514, 1999.

[21] NIST. A statistical test suite for random and pseudorandom number generators for cryptographic applications, 2001.

[22] A. Sayeed and A. Perrig. Secure wireless communications: Secret keys through multipath. In *IEEE Int. Conf. Acoustic, Speech & Signal Processing (ICASSP'08)*, pages 3013–3016, April 2008.

[23] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.

[24] M. A. Tope and J. C. McEachen. Unconditionally secure communications over fading channels. In *Military Communications Conference (MILCOM 2001)*, volume 1, pages 54–58, Oct. 2001.

[25] S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983.

[26] 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.

[27] C. Ye, A. Reznik, and Y. Shah. Extracting secrecy from jointly gaussian random variables. In *2006 IEEE International Symposium on Information Theory (ISIT'06)*, pages 2593–2597, July 2006.

[28] C. Ye, A. Reznik, G. Sternberg, and Y. Shah. On the secrecy capabilities of ITU channels. In *IEEE VTC'07-Fall*, pages 2030–2034, Oct. 2007.