Exploiting Multiple-Antenna Diversity for Shared Secret Key Generation in Wireless Networks

Kai Zeng, Daniel Wu, An (Jack) Chan, Prasant Mohapatra
Department of Computer Science, University of California, Davis, CA 95616
Email: {kaizeng,danwu,anch,pmohapatra}@ucdavis.edu

***Abstract*—Generating a secret key between two parties by extracting the shared randomness in the wireless fading channel is an emerging area of research. Previous works focus mainly on single-antenna systems. Multiple-antenna devices have the potential to provide more randomness for key generation than single-antenna ones. However, the performance of key generation using multiple-antenna devices in a real environment remains unknown. Different from the previous theoretical work on multiple-antenna key generation, we propose and implement a shared secret key generation protocol, Multiple-Antenna KEy generator (MAKE) using off-the-shelf 802.11n multiple-antenna devices. We also conduct extensive experiments and analysis in real indoor and outdoor mobile environments. Using the shared randomness extracted from measured Received Signal Strength Indicator (RSSI) to generate keys, our experimental results show that using laptops with three antennas, MAKE can increase the bit generation rate by more than four times over single-antenna systems. Our experiments validate the effectiveness of using multi-level quantization when there is enough mutual information in the channel. Our results also show the trade-off between bit generation rate and bit agreement ratio when using multi-level quantization. We further find that even if an eavesdropper has multiple antennas, she cannot gain much more information about the legitimate channel.**

I. Introduction

Traditional security mechanisms rely on cryptographic keys to support various security services, including authentication, confidentiality, and integrity. With the increasing popularity of wireless communications, key establishment in wireless networks becomes more challenging. For example, in a dynamic environment, mobile parties need to form their associations on-the-fly. A certificate authority or a key management center may not be available in such scenario. Thus, it is necessary to have alternative methods for key establishment between wireless entities without relying on a fixed infrastructure.

Recently, there is an increasing interest in generating a shared secret key between wireless devices by exploiting reciprocal and location-specific properties of a wireless fading channel [1], [2]. Based on the reciprocity, the bidirectional channel states should be identical between two transceivers at a given instant of time. In a multipath or mobile environment, the channel states randomly fluctuate due to fading. Therefore, two legitimate parties can take advantage of this natural correlated random process to generate a shared key. Furthermore, the channel state observed at an eavesdropper is uncorrelated with the legitimate channel if the eavesdropper is more than half a wavelength away from legitimate parties [3].

Generating shared secret keys via wireless channels has advantages over traditional mechanisms, e.g., Diffie-Hellman key exchange. It can eliminate the requirement of an authenticated communication channel and does not rely on the intractability of certain computational problems such as factoring large integers [2], [4]. Actually, integers could be factored in polynomial time using Shor's quantum factoring algorithm on quantum computers [5]. Although practical quantum computers may not be built in years, it is worthwhile to research on other key establishment mechanisms that do not rely on the computational intractability.

Previous experimental work shows two wireless devices can generate a shared key at approximately 1bit/sec by using off-the-shelf 802.11a hardware [2]. Under this secret bit generation rate, Alice and Bob may not be able to generate a long enough key in a mobile environment where the connectivity may be intermittent. For example, Advanced Encryption Standard (AES) requires a key length with at least 128 bits, then it takes about two minutes to generate a key. Therefore, it is necessary to increase the bit generation rate for real-world usage.

Intuitively, multiple-antenna devices have the potential to provide more randomness for key generation by exploiting spatial diversity. This potential, however, has not been well explored in the literature. Although a recent work studies the theoretic limits of multiple-antenna key generation [6], the feasibility and performance of key generation using off-the-shelf multiple-antenna devices in a real environment remains unknown. Furthermore, the binary quantization method proposed previously [2] may not fully make use of the randomness in the channel. Multi-level quantization can be applied to increase the bit generation rate when there is enough mutual information in the channel.

In this paper, we propose and implement a shared secret key generation protocol, *Multiple-Antenna KEy generator* (MAKE), that exploits spatial diversity in a real system with off-the-shelf 802.11n multiple-antenna devices. We also implement a practical multi-level quantization mechanism to increase the bit generation rate. We conduct extensive experiments and analysis in both indoor and outdoor environments. To the best of our knowledge, this is the first work on studying the shared key generation problem in a *real* multiple-antenna wireless system.

Experimental results show that using laptops with three antennas, MAKE can increase the bit generation rate by more

This research was supported in part by the National Science Foundation through the grant CNS-0709264 and the Army Research Office through the MURI grant W911NF-07-1-0318.

than four times over single-antenna systems. We also show that the wireless channel has enough mutual information for using multi-level quantization, which achieves higher bit generation rates than binary quantization. However, there is a trade-off between bit generation rate and bit agreement ratio when using multi-level quantization. We also find that the information obtained by a passive eavesdropper is negligible even if the eavesdropper has multiple antennas.

We summarize our main contributions as follows:

- We propose and implement a multiple-antenna key generation protocol (MAKE) in a real wireless system by using off-the-shelf 802.11n multiple-antenna devices.
- We investigate the capability of multiple-antenna systems on increasing the performance of the shared secret key generation over single-antenna systems in both real indoor and outdoor environments. Our experimental results show that using laptops with three antennas, MAKE can increase the bit generation rate by more than four times over single-antenna systems.

The rest of this paper is organized as follows. In Section II, we discuss the related work. Section III introduces the system model. We detail the design and implementation of MAKE in Section IV. In Section V, we present experimental setup and in Section VI, we analyze the experimental results. We discuss the robustness of our protocol against various attacks in Section VII. We conclude this paper and discuss the future work in Section VIII.

II. Related Work

Recently, there has been an increasing interest in exploiting the wireless channel randomness and principle of reciprocity to generate shared secret keys between wireless parties [1], [2], [6]–[9].

Generating identical bit strings between two parties based on two correlated random processes has been studied in the information theory community. Assuming Alice and Bob have already shared an authenticated channel, it is possible to extract the same random bits for the two parties. Even if an adversary, Eve, eavesdrops on all the communication between Alice and Bob, she would not have sufficient information to figure out the shared key [10]–[12]. The mechanism for generating shared secret keys between Alice and Bob generally includes three phases: advantage distillation, information reconciliation, and privacy amplification [13]. Previous work assumed an authenticated channel for information reconciliation while generating shared secret keys [1], [8], [9]. One recent work removed this assumption and proposed a shared secret key generation algorithm using level-crossings and quantization to extract secret bits from an unauthenticated wireless channel [2]. Another work proposed a method for key generation based on phase reciprocity of frequency selective fading channels [7]. While all the previous work focused on single-antenna systems, a recent work studies the theoretical limits of key generation in multiple-antenna systems [6]. However, the feasibility and performance of key generation in real environments using off-the-shelf multiple antenna devices remains unknown. Furthermore, the existing scheme [2] using the ICMP PING packets to probe the channel cannot be directly applied to multiple antenna systems, because PING cannot modify the transmitting antenna on a per frame basis.

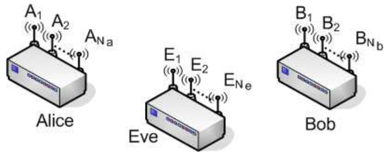


Fig. 1. Alice and Bob generate a shared key using the multiple antennas through the wireless channel. Eve can eavesdrop on the communication between them.

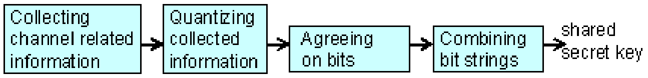

Collecting channel related information
Quantizing collected information
Agreeing on bits
Combining bit strings
shared secret key

Fig. 2. Steps of generating shared secret keys in multiple-antenna systems using the channel related information.

Our work generates shared secret keys in a real multiple-antenna system. Different from all the previous work, we build an experimental multiple-antenna testbed using off-the-shelf IEEE 802.11n equipment. Our protocol allows the sender to change the transmitting antenna on a per frame basis, which yields a controllable and efficient way to harvest the multiple-antenna diversity for key generation. We further validate the potential of multi-level quantization in increasing the key generation rate when there is enough mutual information in the channel. To the best of our knowledge, this is the first work on studying key generation in real multiple-antenna systems.

III. Shared Secret Key Generation in Multiple Antenna Systems

Figure 1 illustrates our multiple-antenna system model. Two legitimate parties, Alice and Bob, want to generate a shared secret key using the channel related information (e.g. signal strength). They are equipped with N_a and N_b antennas, respectively. There is an adversary, Eve, who tries to compromise the generated key by eavesdropping on the communication between Alice and Bob. Eve is also equipped with multiple (N_e) antennas. In this paper, we assume a passive attacker model and mainly focus on the key generation between Alice and Bob. We will give a more detailed discussion on the robustness of the key generation protocol against various attacks in Section VII.

To generate shared secret keys in the multiple antenna system, Alice and Bob perform the steps shown in Figure 2. We will detail these steps in this section.

A. Collecting Channel Related Information

For shared key generation, a variety of channel related information can be used. They include channel impulse response [11], signal envelopes [1], signal phases [7], and received signal strength indicator (RSSI) [2], [8]. We use RSSI as the channel related information in this work due to the ease of extracting RSSI from an off-the-shelf wireless card. We would like to emphasize that the methodology and protocol presented in this paper is applicable to any other channel related information when they are available.

For collecting channel related information, Alice and Bob have to transmit probing frames to each other and record the measurement on both sides for every antenna pair, $A_i - B_j$, for $1 \leq i \leq N_a$ and $1 \leq j \leq N_b$. Suppose two sequences, $\mathbf{h}_{ij} = [h_{ij}(t_1), h_{ij}(t_2), ..., h_{ij}(t_n)]$ and $\mathbf{h}_{ji} = [h_{ji}(t'_1), h_{ji}(t'_2), ..., h_{ji}(t'_n)]$ are measured on antenna B_j and A_i, respectively. $h_{ij}(t_k)$ is the channel related information (a random variable) estimated from the probing frame sent from antenna A_i received by antenna B_j at time t_k. In practice, although the estimates $h_{ij}(t_k)$ and $h_{ji}(t'_k)$ may not be exactly the same due to measurement error or channel variation, they would be highly correlated if Alice and Bob probe the channel at a fast enough rate (i.e., $(t'_k - t_k)$) that is shorter than the channel coherence time. Within the channel coherence time, the channel is considered stable and predictable, so $h_{ij}(t_k) \approx h_{ij}(t'_k)$. Under the principle of reciprocity, $h_{ij}(t'_k) \approx h_{ji}(t'_k)$. Thus, $h_{ij}(t_k) \approx h_{ji}(t'_k)$.

B. Quantizing Collected Information

After gathering enough measurements of channel related information, Alice and Bob will quantize each of their measurement into a bit string based on the randomness of the measurements. First, they have to extract the randomness in the measurements.

1) Extracting Randomness: The raw collected information consists of deterministic component which is determined by the distance (or path loss) between Alice and Bob. For example, the RSSI will be larger if Alice and Bob are closer. To deal with this issue, we need to cancel out the large scale deterministic component in the measurement and extract the small scale randomness (fading) in it. We apply a moving window average method to serve this purpose. As described in Eq. (1), we convert the original measurements $h(t_k)$ to "small-scale" $\widetilde{h}(t_k)$ by subtracting the mean of the measurements within a window with size w centered by $h(t_k)$. The window size w should be chosen such that the large scale component does not change much in the window and the small scale fluctuation is remained after converting.

$$\widetilde{h}(t_k) = h(t_k) - \frac{\sum_{i=k-\lfloor \frac{w-1}{2} \rfloor}^{k+\lfloor \frac{w}{2} \rfloor} h(t_i)}{w} \qquad (1)$$

2) Deciding Quantization Levels: We perform quantization on the small scale measurements. The more shared randomness there is between Alice and Bob, the more levels we can split the measurements into. If the channel provides enough mutual information, instead of performing binary quantization, we can apply multi-level quantization [2]. Theoretically, if we want the bit agreement ratio to approach to 1.0, the bit length of the resulting quantization should be bounded by the mutual information between Alice and Bob [9].

In practice, Alice does not know the mutual information between Bob and herself. But she can compute the estimated entropy of the measurements. As long as the reciprocity holds, the estimated entropy should be close to (but no less than) the mutual information. So she can use the estimated entropy to infer the mutual information. The *estimated entropy* is calculated as $\mathcal{E} = -\sum_{\widetilde{h}} p(\widetilde{h}) log_2 p(\widetilde{h})$, where $p(\widetilde{h})$ represents the frequency occurrence of measurement $\widetilde{h}$ in the collected channel related information.

Since the estimated entropy is an upper bound of the mutual information between Alice and Bob, we should not use the quantization level higher than the estimated entropy if we want a high bit agreement ratio. Therefore, the maximum quantization level, v, is bounded by $v \leq 2^{\mathcal{E}}$.

3) Deciding Quantization Intervals: After deciding quantization levels, we have to decide the quantization interval. For comparison purposes in this paper, we examine binary quantization and multi-level quantization techniques. Binary quantization is where a measurement is converted to bit '1' if it is larger than q_+, and '0' if it is less than q_- [2]. q_+ and q_- are the mean of the measurements plus and minus a scaled standard deviation, respectively.

For multi-level quantization, more steps must be taken. In order to increase the bit agreement ratio, we insert guard bands, g_i, between two consecutive quantization levels q_{i-1} and q_i. Assuming the measurement, $\widetilde{h}$, follow a certain probability distribution, $f_{\widetilde{h}}$, we seek a quantization scheme such that all outputs are equiprobable. We use α to denote the guard band to data ratio which is the excluded measurements in all the guard bands over the total measurements. We assume each guard band excludes the same portion of measurements. Suppose we use m quantization levels (from level 0 up to $m-1$), we have quantization intervals $I_0 = (q_0, q_1 - g_1]$, $I_1 = (q_1, q_2 - g_2]$, ... , $I_{m-1} = (q_{m-1}, q_m)$, where q_0 and q_m is the minimum and maximum value of $\widetilde{h}$, respectively. The value of q_i $(1 \leq i \leq m-1)$ is determined by:

$$\int_{q_{i-1}}^{q_i - g_i} f_{\widetilde{h}} d\widetilde{h} = \frac{1-\alpha}{m}, \quad \int_{q_i - g_i}^{q_i} f_{\widetilde{h}} d\widetilde{h} = \frac{\alpha}{m-1} \qquad (2)$$

Using the quantization intervals solved in Eq. (2), we can quantize each measurement to a certain level if it falls into the corresponding interval. If m levels are used, each level is represented by an n-bit string ($n = log_2 m$) whose decimal value is equal to the level index. We call the multi-level quantization described in this section as m-ary quantization. Note that our 2-ary quantization is different from the binary quantization used in [2].

C. Agreeing on Bits

For each quantized measurement corresponding to each antenna pair, Alice records the start positions of excursions with consecutive s measurements quantized to the same level. Excursions are counted only once (if there are more than s consecutive same level measurements, the next excursion starts at $s+1$). She sends Bob a message containing the positions of these excursions. Bob then checks the excursion in his own measurements at the positions specified by Alice. Due to measurement error or channel variation, Bob may not observe an excursion at all the same positions. He only records the positions where he also observes excursions. These positions are a subset of the positions Alice sends to him. Then Bob sends that positions back to Alice. Both Alice and Bob concatenate the bit string quantized from the measurements on the positions to generate a bit string.

Here is an illustrative example. Suppose Alice and Bob each has eight measurements. After binary quantization, Alice obtained "00101111" and Bob obtained "00111011". Assume excursion size is 2, then Alice finds three excursions "00", "11", and "11" starting at positions 1, 5, and 7, respectively. She sends 1,5,7 to Bob. Bob observes these positions in his list and finds excursions starting at positions 1 and 7. He sends

1,7 back to Alice. Then they use the bits at positions 1 and 7 to generate the final shared key as “01”.

In practice, it could still happen Alice and Bob come up with different bits. By increasing the excursion size, we can decrease the chance of disagreement. To make sure Alice and Bob generate the same key, they can also apply existing information reconciliation and error correction mechanisms, such as low-density parity-check (LDPC) codes [14].

D. Combining Bit Strings

After Alice and Bob agree on a bit string on each antenna pair, they combine the bit strings to generate a final shared secret key. Simply concatenating the bit strings may not yield a random secret key because correlation between antenna pairs may cause correlation between the bit strings [15]. One technique to combine multiple bit strings and de-skew the correlation between them is by using the bit-wise XOR function [16]. We interleave the bits from different bit strings in time sequence, and XOR a certain number of bits together to enhance the randomness of the final key. In this way, the randomness of the combined bit string is not compromised. Other privacy amplification technique such as universal hash can also be applied on the concatenated bit string to improve its randomness [17].

In the next section, we discuss the design and implementation of our key generation protocol.

IV. Protocol Design and Implementation

Now we discuss the detailed design and implementation of our multiple-antenna key generation protocol, MAKE. For Alice and Bob to generate a shared key, our protocol contains two stages: channel related information collection and key generation. The channel related information collection stage corresponds to the first step in Figure 2, and key generation stage includes all the remaining steps. For a practical usage of our protocol with the existing off-the-shelf 802.11n hardware, we use RSSI as the channel related information. Previous work used ICMP PING packets to collect the RSSI for single-antenna systems [2]. However, it is not applicable for our multiple-antenna system because PING cannot modify the transmitting antenna on a per frame basis. In order to harvest the multiple-antenna diversity gain, we then propose the synchronous channel probing in MAKE as follows.

A. Channel Related Information Collection

One way to exploit the multiple-antenna diversity is to measure the RSSI between each antenna pair in a round-robin way. In our implementation, both Alice and Bob have three antennas which makes nine antenna pairs. Suppose we probe the sub-channels periodically in the order of $\langle A_1 - B_1, A_3 - B_3, A_2 - B_1, A_1 - B_3, A_3 - B_2, A_1 - B_2, A_3 - B_1, A_2 - B_3, A_2 - B_2\rangle$ shown in Figure 3, we will get nine RSSI sequences corresponding to each sub-channel respectively at both Alice and Bob sides.

The motivation for this probing method comes from two facts: First, each sub-channel has a limited amount of dynamics, which is constrained by the channel coherence time [1]. It then becomes unnecessary to use a very high probing rate to extract the mutual information in a single channel. Second, a single bidirectional probing can be done much faster than the channel coherence time. This allows us to probe multiple sub-channels within the channel coherence time. So there is enough room to exploit multiple-antenna diversity by probing different sub-channels in such a round-robin way.

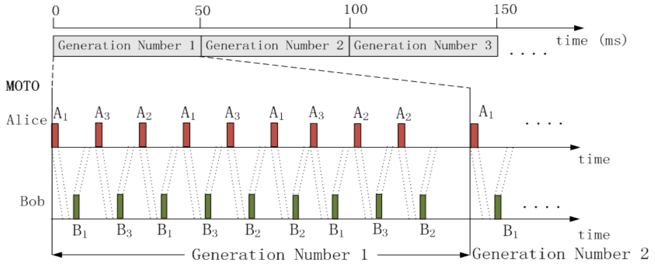
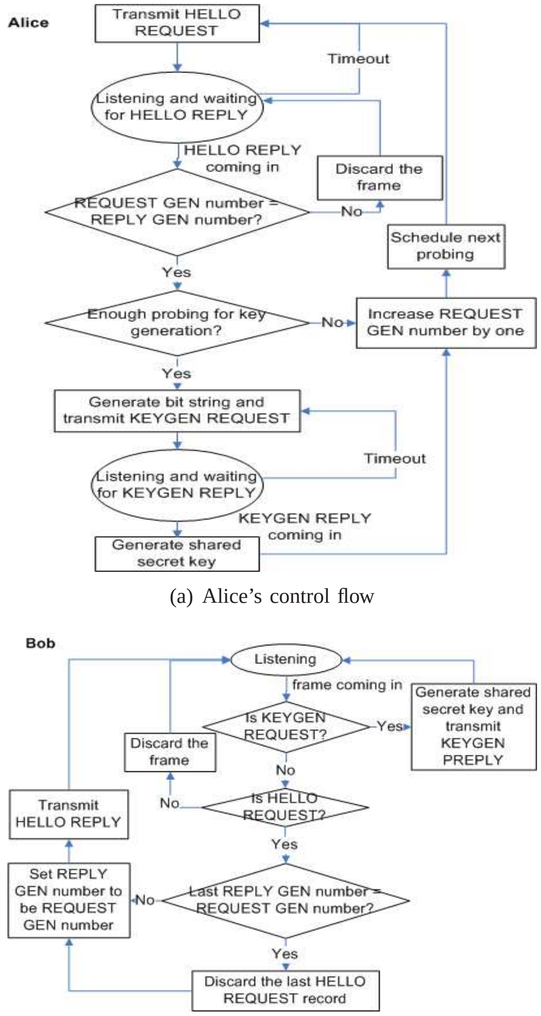


Fig. 3. An example of channel probing in multiple antenna systems where both Alice and Bob have three antennas.

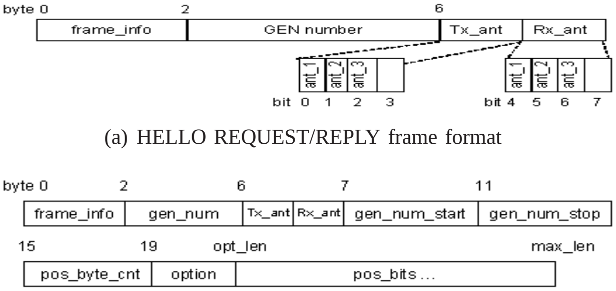

Alice
Transmit HELLO REQUEST
Timeout
Listening and waiting for HELLO REPLY
HELLO REPLY coming in
Discard the frame
REQUEST GEN number = REPLY GEN number?
No
Schedule next probing
Yes
Enough probing for key generation?
No
Increase REQUEST GEN number by one
Yes
Generate bit string and transmit KEYGEN REQUEST
Timeout
Listening and waiting for KEYGEN REPLY
KEYGEN REPLY coming in
Generate shared secret key
Bob
Listening
frame coming in
Generate shared secret key and transmit KEYGEN PREPLY
Is KEYGEN REQUEST?
Yes
Discard the frame
No
No
Is HELLO REQUEST?
Transmit HELLO REPLY
Yes
Set REPLY GEN number to be REQUEST GEN number
No
Last REPLY GEN number = REQUEST GEN number?
Yes
Discard the last HELLO REQUEST record

(a) Alice’s control flow

(b) Bob’s control flow

Fig. 4. Control flows for Alice and Bob.

In our protocol, Alice is the initiator of the channel probing. For each antenna-pair, the control flows at Alice and Bob sides are shown in Figure 4. Using Figure 3 as illustration, Alice begins with $A_1 - B_1$ where she transmits a HELLO REQUEST (shown in Figure 5(a)) by using antenna A_1, and sets the Tx_ant and Rx_ant fields in the frame to indicate the antenna pair being probed. After receiving the HELLO REQUEST,



(a) HELLO REQUEST/REPLY frame format

(b) KEYGEN REQUEST/REPLY frame format

Fig. 5. HELLO REQUEST/REPLY frame format

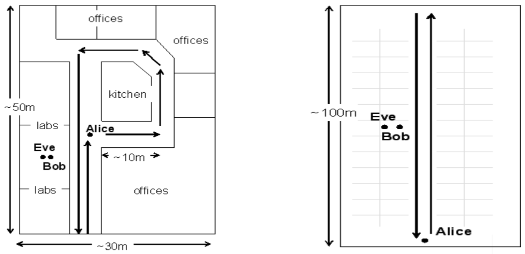
Bob can collect three RSSI values on his three antennas, but he only records the RSSI on the indicated receiving antenna B_1. He then instantly echoes a HELLO REPLY using transmitting antenna B_1. Alice will record the RSSI value on A_1 when she receives the reply. When the time for probing channel $A_3 - B_3$ comes, Alice transmits a HELLO REQUEST through antenna A_3, and Bob will reply it through antenna B_3. The probing continues according to the probing sequence, and continues in a round-robin fashion.

Due to interference or severe channel fading, a HELLO REPLY can be corrupted, Alice will resend the non-replied HELLO REQUEST when a small timeout expires. For each antenna pair, a generation number (GEN number) is used to keep track of the probings. The generation number is increased when a new HELLO REQUEST is generated for the corresponding antenna pair. It will not change when retransmitting a HELLO REQUEST. If the generation number of the received HELLO REPLY is not equal to that of the HELLO REQUEST just sent, Alice will discard the frame and wait for the expected one. Similarly on Bob side, he will check if the newly received HELLO REQUEST generation number is the same as the previous generation number he used for sending HELLO REPLY. If a duplicated HELLO REQUEST is received, Bob will discard the previous HELLO REQUEST record, and use this new one as the record and send a reply.

When a certain antenna pair has collected enough RSSI values, Alice and Bob will start the key generation step.

B. Key Generation

Alice initiates the key generation process. She decides quantization levels and performs the quantization on her RSSI list as described in Section III-B. She then sends a KEYGEN REQUEST (shown in Figure 5(b)) to Bob. In the KEYGEN REQUEST frame, she indicates which antenna-pair measurements are used for key generation by setting the Tx_ant and Rx_ant. She also tells Bob which portion of the RSSI list is used by using gen_num_start and gen_num_stop fields. She indicates the start positions of excursions using the pos_bits field. The field pos_byte_cnt indicates the actual size of pos_bits. Alice adds other information such as the quantization levels to $option$ field. After receiving the KEYGEN REQUEST, Bob will quantize his lists using the same quantization levels (but may use different intervals according to his own measurements). Bob finds a subset of the positions where he also finds excursions, and sends a KEYGEN REPLY to Alice indicating those positions in the pos_bits field. Both Alice and Bob generate the key in the same way based on those positions. Further reconciliation mechanisms [14], [18] can be applied by exchanging more KEYGEN REQUEST and REPLY frames if the key is not agreed.



(a) Indoor environment

(b) Outdoor environment

Fig. 6. Experimental settings.

C. Implementation

We implemented MAKE on Dell e5400 laptops which run a modified Fedora Linux kernel version 2.6.29-rc5-wl based on the wireless-testing tree. Each laptop is integrated with three antennas, and equipped with an 802.11n Intel WiFi Link 5300 wireless card. We made modifications to the Linux wireless device driver (iwlagn), the 802.11 stack (mac80211) and the kernel-to-userspace communication library (radiotap) for instrumentation purposes. The modifications allow the nodes to control the transmitter antenna and modulation rate from userspace on a per-frame basis. It also allows the recording of all three antenna RSSI values per frame on frame reception. The RSSI provided by the driver is an integer value in the range $[-95, -20]$. We use reserved bits "0110" in *frame control* field of MAC header to indicate the frame for MAKE. This program is written in C using threads and raw sockets to communicate with a wireless monitor interface.

V. EXPERIMENTAL SETUP

To study the feasibility and efficiency of MAKE, we conducted extensive experiments in both real indoor and outdoor environments. We use three Dell e5400 laptops acted as Alice, Bob and Eve, respectively. To communicate with each other, Alice and Bob use channel one in the 2.4GHz frequency, 12Mbps modulation rate, and 15dBm transmission power. Alice and Bob run the MAKE protocol to probe the RSSI on both sides for each antenna pair and generate shared keys as described in Section IV. The communication duration is ten minutes for each run. Eve eavesdrops on all the communications between Alice and Bob, and records the RSSI on her three antennas for each frame she overheard. We perform 30 runs under different environments and configurations.

A. Experimental Environment

Experiments were conducted under indoor and outdoor environments. In both environments, Alice is walking at a speed about $1m/s$, while Bob and Eve are stationary and placed close to each other (0.5m apart). For indoor, Alice is walking in the hallway of the second floor in the Watershed Sciences building at UC Davis, while Bob and Eve stay in a room (Figure 6(a)). For outdoor, Alice is walking back and forth in a parking lot outside of Watershed Sciences, while Bob and Eve stay on the floor (Figure 6(b)).

The effectiveness of MAKE depends on how much mutual information the wireless channel possesses and how much mutual information MAKE can extract from it. Indoor environment introduces more fading, which has more randomness and mutual information than outdoor environment. By using the indoor and outdoor environments, we can have a comprehensive understanding of how MAKE performs in different fading scenarios. Interference exists in both environments due to nearby campus 802.11 access points operating on the same channel.

B. Experimental Parameters

Besides varying the experimental environment, we set different protocol parameters for MAKE.

When multiple antennas are used in MAKE, we call it Multiple One antenna To One antenna (MOTO) mode. Since there are 9 antenna pairs, we have $9!$ sequences. It is impossible to test them all. We tested two probing sequences: S_1 shown in Figure 3, and $S_2 = \langle A_1-B_1, A_2-B_1, A_3-B_1, A_3-B_2, A_2-B_2, A_1-B_2, A_1-B_3, A_2-B_3, A_3-B_3\rangle$. We choose these two sequences because they present different antenna correlations between two consecutive probings. The antenna correlation may yield bit correlation between the generated bits in each antenna pair.

When we set the probing sequence to contain only one antenna pair, MAKE degenerates to the single antenna case. We call it Single One antenna To One antenna (SOTO) mode. In order to compare the performance of multiple antenna systems with single antenna systems, we tested $A_1 - B_1$, $A_2 - B_2$, and $A_3 - B_3$ cases.

The time intervals between two consecutive probing frames (HELLO REQUEST) for each antenna pair will have effect on how much randomness we can capture or sample from the channel. We set it to 50ms, 25ms, or as short as the device driver allows (about 2.5ms).

C. Evaluation Metrics

For a shared key generation protocol, the three most important evaluation metrics are:

1) **Shared bit generation rate**: it evaluates how fast Alice and Bob can generate agreed shared secret bits using as a key. It is calculated as the number of agreed bits between Alice and Bob over the communication duration.
2) **Bit agreement ratio**: it measures how many bits are agreed between Alice and Bob. It is the ratio of the number of agreed bits to the total number of bits in the generated strings from Alice and Bob. This metric evaluates the potential of Alice and Bob agreeing on the same bit string.
3) **Randomness**: we use the approximate entropy [19] as an indicator of the randomness of the bit string. With log base 2, the approximate entropy scales from 0 to 1. Larger approximate entropy indicates more randomness of the bit string.

VI. Experimental Results and Performance Evaluation

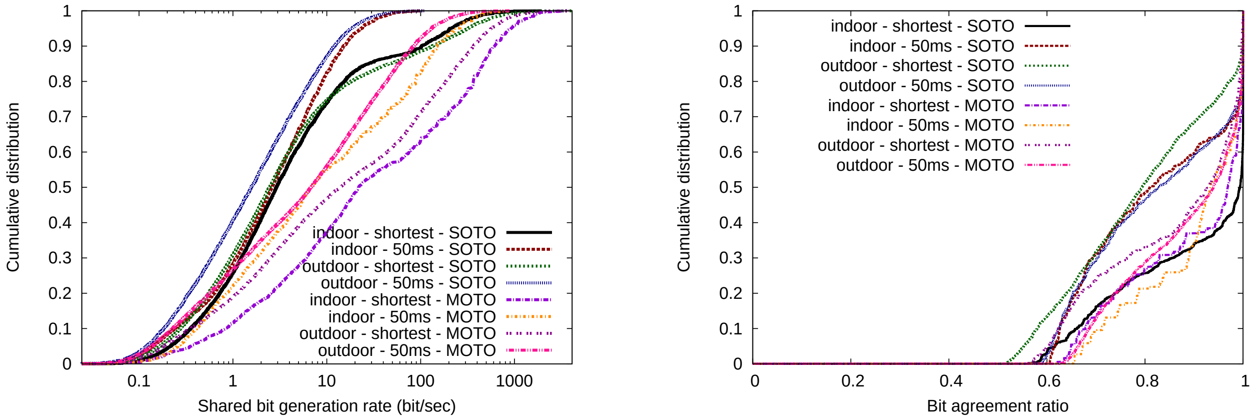
To evaluate the performance of MAKE, we carried out the experiments with the combination of different environments and parameters described in the previous section. We also carried out off-line analysis on the logged data to get more insight about how key generation parameters would affect the performance. We vary the excursion size (s) from 1 to 15, quantization levels m from 2 to 8, guard band to data ratio α from 0 to 0.6, and XOR count from 0 (no XOR applied) to 8. For all the average values we graphed, we indicate the 95% confidence interval. We present and analyze our results as follows.

A. Shared Randomness between Alice and Bob

The key generation performance is fundamentally contained by the shared randomness between Alice and Bob. To quantify this shared randomness, we applied the method proposed in [20] to compute the mutual information of the recorded RSSI between Alice and Bob. The larger the mutual information between two random processes, the more information they share. Using $A_1 - B_1$ as an example, Table I shows the mutual information between Alice and Bob is larger than 2 bits in all the tested real environments and under different probing intervals. This observation validates there is enough shared randomness between Alice and Bob to allow multi-level quantization. We also computed the mutual information about $A_2 - B_2$ and $A_3 - B_3$. $A_2 - B_2$ has more than 2 bits mutual information. While $A_3 - B_3$ presents lower mutual information which is about 1.6 and 1.4 for indoor and outdoor, respectively. This observation indicates that different channels may contain different shared randomness for key generation. A more sophisticated channel probing and key generation protocol which opportunistically take advantage of channels with high randomness will be our future work.

B. Eve's Inability to Gain Correct Channel Information

One interesting question is "when Eve has multiple antennas, can she obtain more information about the channel between Alice and Bob?" To evaluate this, we calculate the mutual information between Eve and Alice (Bob), shown in Table I. We first look at Eve's observation at each antenna individually. For instance, $B_1 - E_1$ (0.2bit) has about two bits lower mutual information compared to $A_1 - B_1$. The mutual information between Alice and Eve is even smaller, due to the fact that Alice and Eve are very far apart during the experiments. Assuming Eve uses all her three antenna RSSI information. The $A_1A_1A_1 - E_1E_2E_3$ and $B_1B_1B_1 - E_1E_2E_3$ columns in Table I shows the mutual information in concatenating the RSSI information from all three of Eve's antennas and comparing it against Alice's and Bob's RSSI. Even with extra information from all three antennas, Eve still cannot gain very much information about Alice's (or Bob's) channel information. The $B_1B_1B_1 - E_1E_2E_3$ column has much lower mutual information than $B_1 - E_1$ due to the fact that E_2 and E_3 also have low mutual information with B_1, which dilutes the mutual information of the combined lists. To summarize, confidentiality between Alice and Bob during the bit generation is achieved since Eve does not share the same channel information as Alice-Bob channel even Eve has multiple antennas.



(a) CDF of bit generation rate.

(b) CDF of bit agreement ratio.

Fig. 7. Results for SOTO vs. MOTO with approximate entropy ≥ 0.9 for 2-ary and 4-ary quantization levels.

TABLE I
MUTUAL INFORMATION FOR SOTO ANTENNA-PAIRS BETWEEN ALICE AND BOB, ALICE AND EVE, AND BOB AND EVE

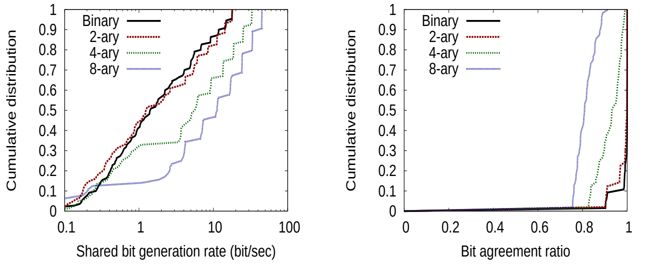
Scenario		Mutual Information (bits)				
		$A_1 - B_1$	$A_1 - E_1$	$B_1 - E_1$	$A_1 A_1 A_1 - E_1 E_2 E_3$	$B_1 B_1 B_1 - E_1 E_2 E_3$
Indoor	shortest	2.1691	0.0061	0.3072	0.0045	0.023
	25ms	2.1633	0.0198	0.2146	0.0015	0.010
	50ms	2.1764	0.0081	0.3489	0.0060	0.045
Outdoor	shortest	2.2792	0.0482	0.2979	0.0468	0.0460
	25ms	2.1891	0.0411	0.1512	0.0364	0.0248
	50ms	2.2722	0.0478	0.2643	0.0181	0.1260

C. Improvement of MOTO over SOTO

Figure 7(a) is a CDF of the shared bit generation rates corresponding to the keys made by 2-ary and 4-ary quantization levels (filtered by approximate entropy ≥ 0.9) under different environments, probing intervals and operating modes. We can see that, for all scenarios, the median bit generation rate of MOTO (corresponding to multiple antenna systems) is at least 4.5 times of that achieved by SOTO (corresponding to single antenna systems). The indoor environment provides higher variation (entropy), so all the indoor cases outperform the corresponding outdoor cases. When we probe the channel faster (with shorter interval), we get higher bit generation rate because we can catch more shared randomness from the channel. However, the bit generation rate is fundamentally constrained by the time-variation of the channel. If the channel itself does not change much in a short interval, even if we can probe the channel at a very high rate, we cannot extract more randomness.

Probing the channel with the shortest possible interval provides an upper bound of the extractable shared randomness. Therefore, the bit generation rate achieved by the shortest SOTO case should be the upper bound of the rate it can achieve. We can clearly see that both MOTO 50ms and MOTO shortest cases achieve much higher bit generation rate than SOTO shortest one. The improvement comes from the multiple-antenna diversity.

Figure 7(b) shows the corresponding bit agreement ratios of the keys. For the MOTO case, the median bit agreement ratio is around 0.9, which is higher than the corresponding SOTO case except for the indoor shortest SOTO one. The agreement ratio improvement of MOTO over SOTO comes from the fact that MOTO uses more sub-channels to generate the key. There exist some sub-channel(s) which provide more shared randomness than the SOTO one. Then for a high quantization level, these sub-channels can achieve higher agreement ratio. Therefore, MOTO, which combines all the sub-channels, can have higher agreement ratio than each SOTO channel. The indoor SOTO with the shortest probing interval achieves the highest median bit agreement ratio because it can extract the most shared randomness from the single channel.



(a) CDF of bit generation rate.

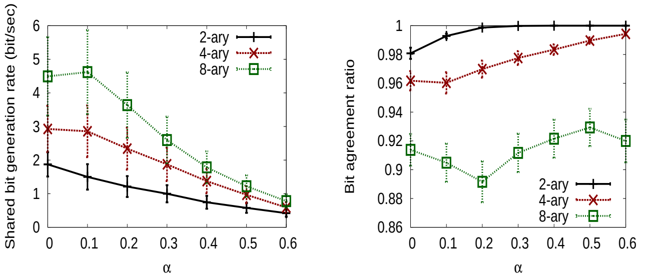
(b) CDF of bit agreement ratio.

Fig. 8. Results for m-ary vs. binary quantization, Indoor - 50ms scenario, SOTO $A_1 - B_1$ with approximate entropy ≥ 0.9.

For different probing sequences used in MOTO, we found that S_2 achieves higher shared bit generation rate but lower bit agreement ratio than S_1 in the outdoor-shortest scenario. But in other scenarios, they present similar performance. It demonstrates that, different probing sequences do affect the key generation performance due to different antenna correlation/diversity among sub-channel probings. Finding the optimal probing sequence according to different scenarios is a challenging issue, and will be our future work.

D. Improvement of Multi-Level Quantization over Binary Quantization

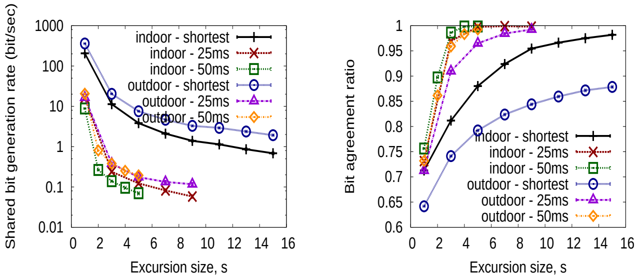
Since m-ary quantization can be used to generate keys with more bits, we compare its performance with the binary quantization method. For evaluation purposes, we tried different quantization levels from 2 to 8. Figure 8(a) is a CDF of bit generation rate under different quantization levels. The “Binary” indicates the method proposed in [2], and the others indicate our method proposed in Section III-B3. They



(a) Impact of guard band on bit generation rate.

(b) Impact of guard band on bit agreement ratio.

Fig. 9. Effects of the guard parameter on key generation for the m-ary, SOTO $A_1 - B_1$, 50ms.



(a) Impact of excursion size on bit generation rate.

(b) Impact of excursion size on bit agreement ratio.

Fig. 10. Effects of excursion size on key generation over all keys generated from our experiments.

are drawn from all the keys with approximate entropy no less than 0.9 generated from $A_1 - B_1$ for indoor with 50ms probing interval. Other antenna pairs show the similar results. Compared with the binary quantization, the higher level m-ary quantization can significantly increase the bit generation rate.

Figure 8(b) is the CDF of the bit agreement ratios between Alice and Bob corresponding to the keys of Figure 8(a). Different from bit generation rate, the bit agreement ratio decreases when higher quantization levels are used. When the quantization level is larger than 4, the maximum bit agreement ratio is 90%. This observation validates the theory that if a nearly 100% bit agreement ratio is desired, we have to keep the quantization level lower than which the mutual information can provide. In this case, the quantization level should be kept below $2^{2.2}$ (see Table I), if we want a 100% bit agreement ratio. We also tried quantization levels larger than 8, which yields even lower bit agreement ratio. So it is non-sense to use high level quantization if the mutual information between Alice and Bob cannot support it.

An interesting observation in Figure 8(a) is that each line shows a step pattern. This jump comes from the different excursion sizes being used. When the excursion size is decreased, more bits would be generated from the whole RSSI records, thus yielding a higher bit generation rate. We will show more detail results about the impact of this factor on the key generation performance in Section VI-E2.

E. Impact of Design Parameters on Key Generation

1) Impact of the Guard Band on Key Generation: Figure 9(a) shows that the bit generation rate decreases as the guard band to data ratio increases. Larger guard band to data ratio means more RSSI records are discarded. This decreases the length of the bit string, and decreases the bit generation rate. On the other hand, as shown in Figure 9(b), the bit agreement ratio increases when guard band to data ratio increases. Because some boundary cases are excluded. However, for 8-ary, as they exceed the theoretical quantization levels ($2^{2.2}$), their bit agreement ratio is low even with the increase of the guard band.

2) Impact of Excursion Size on Key Generation: Recall that excursion size is the number of consecutive processed RSSI records quantized to the same level in order to be counted as a valid bit (or bits in the m-ary case). When the excursion size increases, more consecutive records are needed to generate one bit(s) in the final key, so the bit generation rate decreases (Figure 10(a)). By counting the consecutive records in this way, the final key will be shorter by a factor of the excursion size.

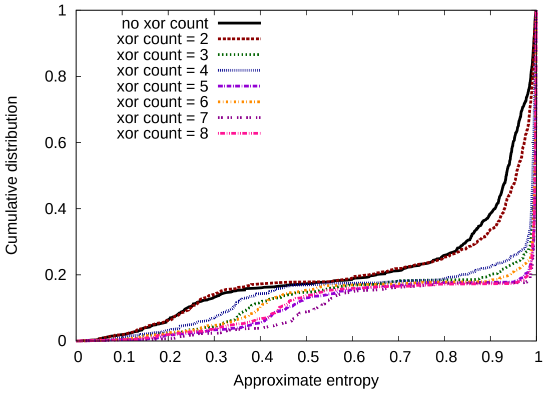


Fig. 11. CDF of the approximate entropy for different XOR counts for the MOTO mode.

The usefulness of the excursion size is shown in Figure 10(b). By increasing the excursion size, the agreement ratio between Alice and Bob increases as well. This increase is due to the fact that given Alice and Bob both find an excursion, the probability for them falling into different quantization levels decreases.

There is a significant increase in bit agreement ratio when the excursion size is two or three for all the different scenarios. These two values are good choices in practice since higher values will give diminishing returns and lower the bit generation rate by a high factor.

Besides guard band to data ratio and excursion size, we also examined different moving window sizes when extracting small scale measurements. We found that a moving window covering a time duration around one second is proper to cancel out large scale component in the measurements. Moving window size has not obvious impact on the performance.

F. Enhancing Key Randomness with XOR

The XOR function takes as an input the bit string and the number of consecutive bits (XOR count) we should XOR together as one bit. To know how the XOR helps improve the randomness of the combined key for MOTO case, we collect all the keys with different XOR count and draw their approximate entropy in Figure 11. Before XOR, the median approximate entropy is 0.95; after XOR it is increased to 1, except the case where xor count equal to two. So there do exist correlations between the bit string generated on each channel. When we combine the bit strings together we need to decorrelate the correlations between the bits. A drawback of using XOR count is that it decreases the final bit string length by a factor of the XOR count. So the shared key generation rate shown for MOTO case in this paper, can be considered as an achievable lower bound.

VII. Robustness Against Attacks

The natural decorrelative properties of a fading channel provides our key generation protocol security against passive eavesdropping attacks as verified in Section VI. However, an active attacker can impersonate Alice or Bob in either the channel related information collection stage or key generation stage of our protocol and inject false frames. We can prevent these spoofing attacks by using similar techniques mentioned in previous works [2], [21]. For example, they can check if the RSSI variations between subsequent frames from the other vary too much. They can also improve the detection accuracy by using all RSSI readings from multiple antennas. To prevent man-in-the-middle attacks against MAKE, Alice and Bob will have to authenticate each other's identity before hand [2]. Another concern is how interference would affect the performance. We conducted our experiments in real indoor and outdoor environments with interference coming from campus 802.11 access points operating on the same channel. It is safe to conclude that MAKE works in such noisy environments where other interference source still follows the 802.11 CSMA/CA medium access control rule. In a more severe scenario, if there is an attacker trying to jam the channel without following the medium access control rule, Alice and Bob may run into trouble in generating shared keys. However, they can detect such attack with high probability if they find they cannot hear each other well in the communication. They can also try to switch to other channels or apply channel hopping to alleviate the effect of the jamming attack. Jamming-resistant key generation will be an interesting topic to explore.

VIII. Conclusion and Future Work

We proposed and implemented a shared secret key generation protocol, MAKE, over multiple-antenna systems. MAKE allows the sender to dynamically change the transmit antenna on-the-fly on a per-frame basis. We carried out extensive experiments in various scenarios to evaluate the performance of MAKE. By exploiting multiple-antenna diversity and by incorporating a practical multi-level quantization mechanism, MAKE demonstrates superior effectiveness in performance. For example, with each laptop having three antennas, MAKE can achieve more than four times higher bit generation rate than single-antenna protocols. We further examined the impact of different design parameters on the performance of key generation. Increasing excursion size and guard band to data ratio can improve the bit agreement ratio, but will decreases shared bit generation rate. We also demonstrate the usefulness of XOR as a randomness enhancement technique when combing keys generated from different antenna pairs.

While the two legitimate users enjoy the high quality key generation, an adversary cannot gain much information about the legitimate channel even if she has multiple antennas. Such findings in the experiments help us rule out the vulnerability of MAKE to passive attacks. In the future, we plan to study opportunistic channel probing to take advantage of higher random sub-channels to maximize the shared bit generation rate. Applying other privacy amplification techniques (such as universal hash) to improve the key combination efficiency is also an interesting topic.

References

[1] 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*. ACM, 2007, pp. 401–410.

[2] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008, pp. 128–139.

[3] T. S. Rappaport, *Wireless Communications: Principles and Practice*. New Jersey: Prentice Hall, 2001.

[4] U. Maurer and S. Wolf, "Secret-key agreement over unauthenticated public channels .I. definitions and a completeness result," *Information Theory, IEEE Transactions on*, vol. 49, no. 4, pp. 822–831, April 2003.

[5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, pp. 1484–1509, 1997.

[6] J. W. Wallace, C. Chen, and M. A. Jensen, "Key generation exploiting mimo channel evolution: algorithm and theoretical limits," in *Proceedings of 3rd European Conference on Antennas and Propagation*. European Association on Antennas and Propagation, 2009.

[7] H. Koorapaty, A. Hassan, and S. Chennakeshu, "Secure information transmission for mobile radio," *IEEE Communications Letters*, vol. 4, no. 2, pp. 52–55, Feburary 2000.

[8] T. Aono, K. Higuchi, M. Taromaru, T. Ohira, and H. Sasaoka, "Wireless secret key generation exploiting the reactance-domain scalar response of multipath fading channels : RSSI interleaving scheme," *Wireless Technology, 2005. The European Conference on*, pp. 173–176, October 2005.

[9] R. Wilson, D. Tse, and R. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *Ultra-Wideband, 2007. ICUWB 2007. IEEE International Conference on*, pp. 270–275, September 2007.

[10] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*. Springer-Verlag New York Inc., 1994, pp. 410–423.

[11] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[12] C. Ye, A. Reznik, and Y. Shah, "Extracting secrecy from jointly gaussian random variables," *Information Theory, 2006 IEEE International Symposium on*, pp. 2593–2597, July 2006.

[13] C. Cachin and U. M. Maurer, "Linking information reconciliation and privacy amplification," *Journal of Cryptology*, vol. 10, pp. 97–110, 1997.

[14] M. Bloch, J. Barros, M. Rodrigues, and S. McLaughlin, "Wireless information-theoretic security," *Information Theory, IEEE Transactions on*, vol. 54, no. 6, pp. 2515–2534, June 2008.

[15] J. Kermoal, L. Schumacher, K. Pedersen, P. Mogensen, and F. Frederiksen, "A stochastic mimo radio channel model with experimental validation," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 6, pp. 1211–1226, August 2002.

[16] D. E. Eastlake, S. D. Crocker, and J. I. Schiller, "Randomness recommendations for security," in *RFC 1750*. IETF, 1994.

[17] C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy amplification by public discussion," *SIAM J. Comput.*, vol. 17, no. 2, pp. 210–229, 1988.

[18] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): design and construction," *Information Theory, IEEE Transactions on*, vol. 49, no. 3, pp. 626–643, Mar 2003.

[19] S. M. Pincus, "Approximate entropy as a measure of system complexity," in *Proceedings of the National Academy of Sciences of the USA*. National Academy of Sciences of the USA, March 1991, pp. 2297–2301.

[20] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E*, vol. 69, no. 6, p. 066138, Jun 2004.

[21] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Fingerprints in the ether: Using the physical layer for wireless authentication," in *IEEE ICC'07*, 2007, pp. 4646–4651.