Shape Matters, not the Size: a New Approach to Extract Secrets from Channel

Yue Qiao[1]
qiaoyu@cse.ohio-state.edu

Kannan Srinivasan[1]
kannan@cse.ohio-state.edu

Anish Arora[1 2]
anish@cse.ohio-state.edu

[1]Department of Computer Science and Engineering
The Ohio State University
Columbus, OH 43210, USA

[2]The Samraksh Company
5980 Venture Dr, Suite 1B
Dublin, OH 43017, USA

ABSTRACT

Existing secret key extraction techniques use quantization to map wireless channel amplitudes to secret bits. This paper shows that such techniques are highly prone to environment and local noise effects: They have very high mismatch rates between the two nodes that measure the channel between them. This paper advocates using the shape of the channel instead of the size (or amplitude) of the channel. It shows that this new paradigm shift is significantly robust against environmental and local noises. We refer to this shape-based technique as **Puzzle**. Implementation in a software-defined radio (SDR) platform demonstrates that Puzzle has a 63% reduction in bit mismatch rate than the state-of-art frequency domain approach (CSI-2bit). Experiments also show that unlike the state-of-the-art received signal strength (RSS)-based methods like ASBG, Puzzle is robust against an attack in which an eavesdropper can predict the secret bits using planned movements.

1. INTRODUCTION

For wireless communications, there has been a great interest in generating shared secrets from the physical layer as a complementary approach to the traditional methods of cryptography. The interest stems from the open nature of the wireless medium and the infrastructure constraints associated with key management in mobile scenarios. There are two main approaches for secret-sharing in wireless. One is based on information-theoretic principles of exploiting the secrecy capacity between Alice and Bob compared to Alice and Eve [1]. The main drawback of this approach is that secrecy is dependent on rather strong assumptions about eavesdropper's capability. Equally importantly, even a modest increase in the spatial density of eavesdroppers harms the secrecy rate of the approach dramatically [9].

The other approach is based on channel reciprocity. Channel reciprocity refers to the physical principle whereby near-simultaneous observations of the channel by two communicating parties are identical due to the channel paths between them being symmetrical. Fig. 1 in Section 3 shows this reciprocity in our testbed. The time for which the wireless channel remains correlated is called the *coherence time*. By extracting channel state information (CSI) from the observed signals, Alice and Bob can share bits by transmitting signals to each other within the coherence time. Furthermore, extensive theoretical analysis and experimentation have shown that observations of the wireless channel over distances larger than half-the-wavelength of the carrier frequency are uncorrelated [10]. In a 2.4GHz ISM band, for instance, at any location farther than 6cm away from Bob, Eve will observe Alice's signal through an uncorrelated channel. Channel reciprocity and spatial decorrelation together make the wireless channel an excellent random source for generating shared secret keys.

There is significant prior work that exploits channel reciprocity for secret extraction. One set of techniques use the received signal strength (RSS) as the secret source [2, 4, 7, 8, 14]. These techniques measure the RSS over different coherent intervals to generate a sequence of RSS. They choose a threshold and transform the signal strength sequence into 1s (if above that threshold) and 0s (if below the threshold). The largest drawback with these techniques is that large variations can be easily introduced by an attacker by blocking transmission every now and then. These variations make the secret predictable since the attacker knows the exact moments at which the signal-to-noise ratio (SNR) will drop or increase. Section 4 presents this attack and shows this vulnerability.

Another set of techniques use the fine-grained temporal [12, 6, 13] or frequency [5] components contained in received signals as the secret source. The temporal techniques use ultra-wideband transmissions ($\approx$ GHz bandwidth) to capture this fine-grained temporal information. Therefore, these techniques are not applicable for narrowband systems such as Wi-Fi (with only 20MHz bandwidth). Furthermore, another challenge in temporal techniques is that temporal information is sensitive to sampling offset which leads to a high rate of secret disagreement. In contrast, the frequency technique of Liu et al [5] is applicable to narrowband systems and is not sensitive to sampling offset. The authors quantize the frequency response in each subcarrier in OFDM and map them to secret bits. In Section 4, we dispute the authors' claim of high secrecy rate and show that the secrets generated from their method is very limited.

HotWireless'14, September 11, 2014, Maui, Hawaii, USA.

ACM 978-1-4503-3076-3/14/09 ...$15.00.
http://dx.doi.org/10.1145/2643614.2643618 .

Overall, this paper takes the stand that the amplitude (the **size**) of a signal –in time or frequency– is prone to perturbations from the environment as well as hardware imperfections. This leads to quantization errors at nodes and high mismatch in secrecy bits generated by wireless nodes. Instead, this paper proposes to use the ***shape*** of a signal to deduce secrecy bits. Specifically, we make the following contributions.

- We propose and implement a shape-based secret extracting algorithm called Puzzle that we show to be robust to noise and device imperfections. In particular, no online or offline device calibration is required.
- We prove that the power spectrum density (PSD) of random data can be used to extract CSI. This implies that no modification is needed for the higher layers of the wireless communication, such as transmitting special training data.
- Our experiments show that Puzzle produces a 5-bit secret per packet and has a 63% improvement in bit mismatch rate than the frequency domain approach mentioned above.

2. SYSTEM MODEL

Consider two wireless nodes, Alice and Bob, that wish to create a shared secret S within a coherence time, during which the channel is stable. An adversary, Eve eavesdrops the communication between Alice and Bob. Our goal is to develop a secret extraction algorithm that introduces as little communication and computation overhead as possible and ensures that Eve obtains little information about S.

Jamming is also a primary threat to wireless protocols. It is true that external interference can alter the shape of the CFR in a subtle way that Puzzle fails to produce an agreed secret while the received signal is still decodable. But in reality, the more probable case is that both of them fail. So jamming is more of a DoS threat to the underlying data communication which puzzle is built upon. Therefore the treatment of this threat is out of scope of our paper.

2.1 Physical Layer Model

2.1.1 Channel model

Assume Alice and Bob operate in a Time-Division Duplexing (TDD) system. If they talk to each other in coherence time, the observed signals of Alice and Bob are represented by

$$y_A(t) \;=\; (h * x_A)(t) + n_A(t) \tag{1}$$

$$y_B(t) \;=\; (h * x_B)(t) + n_B(t) \tag{2}$$

where $h(t)$ is the channel impulse response, which is identical in both directions by virtue of channel reciprocity, x_A and x_B are the signals transmitted by Alice and Bob respectively, $n_A(t)$ and n_B are additive white Gaussian noise with the same variance N, and "$*$" indicates convolution. In the frequency domain, the equations above are rewritten as

$$Y_A(f) \;=\; H(f)\cdot X_A(f) + N_A(f),\ \frac{-W}{2} + f_c < f < \frac{W}{2} + f_c \tag{3}$$

$$Y_B(f) \;=\; H(f)\cdot X_B(f) + N_B(f),\ \frac{-W}{2} + f_c < f < \frac{W}{2} + f_c \tag{4}$$

where W is the transmission bandwidth, f_c is the center frequency, and $H(f)$ is the channel frequency response.

2.1.2 Channel Frequency Response

In this section, we propose two ways to extract the channel frequency response $H(f)$.

Direct calculation: By using pre-defined training signals or decoding the received signals, Alice and Bob know the frequency components $X_A(f)$ and $X_B(f)$ of the transmitted signals. Therefore, they can calculate $H(f)$ easily, assuming that noise can be ignored.

PSD-based method: Let $\{\ x_0, x_1, ..., x_{N-1}\ \}$ be a complex sample sequence. Since the sequence is stationary and random, the auto-correlation of the sequence is

$$R(t_1, t_2) \;=\; \frac{P}{N} \times \delta(t_2 - t_1) \tag{5}$$

where P is the power contained by the signal sequence. Then, the PSD of the sequence is

$$F[R(\tau)] \;=\; \int_{-\infty}^{+\infty} \frac{P}{N} \times \delta(\tau) e^{-j\omega\tau} d\tau = \frac{P}{N} \tag{6}$$

From Equation 6, we know that

$$X_A(f) \;=\; \frac{P_A}{W}, \quad X_B(f) \;=\; \frac{P_B}{W} \tag{7}$$

Combining Equations 3 through 7 we get

$$Y_A(f) \;\approx\; \frac{H(f)\cdot P_A}{W} + N,\ Y_B(f) \;\approx\; \frac{H(f)\cdot P_B}{W} + N \tag{8}$$

According to the above equations, we conclude that the PSD of $y_A(t)$ is the same as that of $y_B(t)$ as long as $P_A = P_B$. It is worth noting that even if $P_A \neq P_B$, the shape of Alice's and of Bob's PSD are still similar. This property is remarkable because it can be extended to the case in which Alice and Bob experience different levels of transmission power, noise or cross-band interference. Even in such cases, the shapes still don't change significantly.

2.2 Threat Model

Eve is motivated to derive the shared secret generated by Alice and Bob. There are two main ways of achieving this.

2.2.1 Eavesdropping

Eve can attempt to derive Ch_{AB} from Ch_{AE} or Ch_{BE}, where Ch_{AB}, Ch_{AE}, and Ch_{BE} denote the channel from Alice to Bob, Alice to Eve, and Bob to Eve, respectively. This may be possible if Eve has full knowledge of the environment. In general, however, full knowledge of the environment is a rather unrealistic assumption, even with multiple eavesdroppers involved, so we do not regard it as the main threat to our system. Instead, we focus on the threat of spatial correlation of the secrets produced by our algorithm. We assume that Eve cannot stalk Alice or Bob to being within half of a wave length of either of them. This assumption is reasonable since close eavesdroppers suffer from a high exposure risk. Recall that theory [10] supports that channels decorrelate beyond half a wavelength.

2.2.2 Planned movement

Eve can move in between Alice and Bob to block and unblock their transmissions. Planned movements can thus introduce predictable increase or decrease of RSS at Alice and Bob. Note that while this attack is harmful to RSS-based methods, without the full knowledge about the environment, Eve cannot, however, predict the impact of the planned move on the frequency response of the channel.

3. SECRET GENERATION

Note from Fig. 1 that although channel reciprocity is clearly apparent for the naked eye, the frequency response curves are more or less shifted or zoomed versions at corresponding frequencies. Moreover, distinct local fluctuations exist. These discrepancies are unavoidable because they spontaneously result from the hardware imperfections and environment interferences. This shows that direct quantization and mapping of the frequency response can lead to high mismatch rates. We, therefore, develop a shape-based approach to solve the encoding problem.

Algorithm 1: CurveCoding

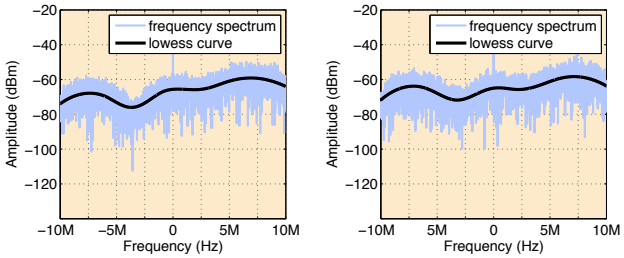
Input:
complex samples $a[0, \cdots, n]$;
number of segments m;
Output:
code $[C_1, C_2, \cdots, C_m]$
Initialization
divide $a[0, \cdots, n]$ *into* m *segments* $b_1, b_2, \cdots, b_m$;
$peak =$ { `Max1`$(a[0, \cdots, n])$ - `Min1`$(a[0, \cdots, n])$}
`PatternGeneration`$(\lfloor n/m \rfloor, m,$ *peak*$)$:
  generate 3 patterns of size $\lfloor n/m \rfloor$: p_1, p_2, p_3;
for $i \leftarrow 1$ **to** m **do**
  $temp = \infty$;
  for $j = 1 \rightarrow 3$ **do**
    $dis =$ Fréchet(b_i, p_j);
    if $temp > dis$ **then**
      $temp = dis$;
      $C_i = j$;
    end
  end
end

Algorithm 2: PatternGeneration

Input:
k, m, *peak*;
Output:
3 patterns $p_1[1, \cdots, k]$, $p_2[1, \cdots, k]$, $p_3[1, \cdots, k]$
for $i \leftarrow 1$ **to** k **do**
  $p_1[i] = \frac{peak \times i}{k \times m}$;
  $p_2[i] = -\frac{peak \times i}{k \times m}$;
  $p_3[i] = \frac{peak}{m \times 2}$;
end

3.1 Curve Smoothing

As mentioned above, even though local details of a power spectral density pair are significantly different, channel reciprocity manifests itself by the similarity of the overall shapes between the pair. By plotting smoothed points, conformal information about the overall shape is extracted despite the local variations. In our algorithm, we adopt Locally Weighted Scatter Plot (Lowess) smoothing [3], a curve fitting method that calculates the smoothed value by applying locally weighted regression over a span. Fig. 1 depicts two PSD curves obtained by two communicating wireless nodes and their corresponding curves after applying Lowess smoothing with a span of 0.4. From Fig. 1, we can see that the Lowess curves coincide with each other almost exactly and the overall shapes are preserved, even though the original ones differ from each other in most of the locations.



(a) Lowess curve by Alice (b) Lowess curve by Bob

Figure 1: Lowess curves derived by Alice and Bob. Lowess curves are much more similar to each other than the original PSD curves as local variations are removed.

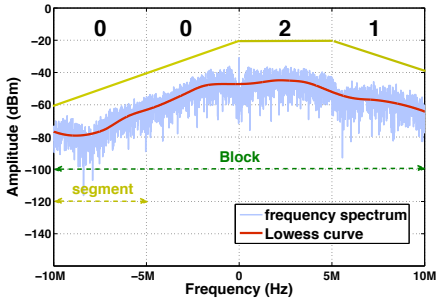


Figure 2: An example of curve encoding.

3.2 Curve Encoding

By using curve smoothing, we obtain two highly similar curves. To solve the encoding problem, let us first briefly consider several alternative methods: 1) encode in accordance with an approximation function that describes the curve; 2) encode in accordance with the statistical properties of the curve; 3) encode by describing the shape of the response. We adopt the third one for the following reason. As mentioned in Section 3.1, channel reciprocity is readily seen by the similarity of the overall shapes between curves. Hence, encoding by describing the shape should preserve most of the information shared by the two ends. By way of contrast, extracting secrets from the statistical properties definitely suffers from losing much of the mutual information. And the approximation function does not tolerate even small deviations, but measurement error and interference make such deviations quite common. Fig. 2 gives an example of curve coding. The curve obtained in a certain band is treated as a block, which can be divided into varying number of segments of equal length, and then the segments are mapped to one of three curve patterns which are of the same length, as shown in Fig. 2. These three patterns are indexed as 0, 1, and 2. The three "predetermined" patterns describe the ascending, descending and steady trend of the curves respectively. By "predetermined", we mean that the indices and the shapes of the patterns are well known to all wireless nodes. The gradient of the ascending and descending lines, however, is decided by each node according to the maximum and minimum values of the smoothed curve, and the length of the segment. We have designed that pattern generation thus to tolerate measurement errors and different device settings. For example, two communicating nodes may

wish to use different tx/rx gains that would amplify the signals differently. Since each pattern is related to the locally received signals, it describes the shape correctly without the need to negotiate with the other node. We set the gradient of the ascending pattern to be relative to $\frac{max-min}{\#\text{ of segments}}$, and likewise for the descending pattern is relative to $-\frac{max-min}{\#\text{ of segments}}$. The segment is then mapped to the most similar of the three patterns by measuring the discrete Fréchet distance [11] δ_{dF} between the segment and the patterns, which measures the similarity of two polygonal curves while taking the location and ordering of the points along the curves into consideration. The smaller the distance, the more is the similarity the two curves share. The complete algorithm is presented in Algorithm 1 and Algorithm 2.

4. EXPERIMENTAL VALIDATION

In this section we study four important metrics to measure the performance of Puzzle.

- **Entropy**: Entropy measures the unpredictability of a random variable X. It is defined as
$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$
where $x_1, \cdots, x_n$ are possible values of X.
- **Bit Mismatch Rate**: Bit mismatch rate is defined as the ratio of the number of bits between Alice and Bob that do not match and the number of bits extracted from the shape of the spectrum.
- **Correlation**: Correlation $\rho_{x,y}$ is defined as
$$\rho_{x,y} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2 \sum_{i=1}^{n}(y_i-\bar{y})^2}}$$
We use correlation to measure the dependence of codes generated by Puzzle relative to different distance between Bob and Eve.
- **Leakage**: Letting p_{mis} be the mismatch rate between Alice and Eve, we define the leakage between them as
$$leakage = \begin{cases} 1-\frac{p_{mis}}{0.5} & \text{if } p_{mis} < 0.5 \\ 0 & \text{otherwise} \end{cases}$$

4.1 Environment and System

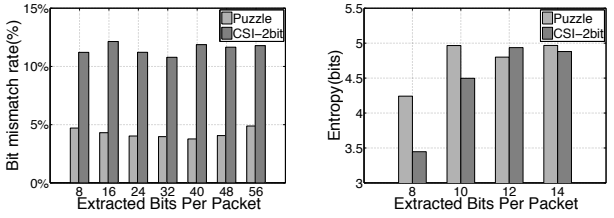
The measurement environment is a lab where there are 6 cubicles. Data were collected during daytime (from 7:00 am to 6:00 pm). Human activities introduced a certain level of interference in the channel, but generally speaking, the environment is quite stable. We conducted the experiment in such a stable environment because we wanted to see clearly the performance comparisons without risking mismatches caused by the changes of the channel itself. In theory, further implementation in mobile environment would give both higher mismatch rate and higher secret bit extraction rate.

The communication system consists of three SDRs. Each of their RF chains contains an XCVR2450 (RF front end), an NI-5781 (data converter module) and an NI PXIe-7965R (a Xilinx Virtex-5 FPGA). Two of the three transceivers transmit at 2.45 GHz with 20MHz bandwidth. We call these two transceivers Alice and Bob. The third transceiver, Eve, overhears the communication. During reception, each transceiver records the I and Q samples at a sampling rate of 100 MHz and down converts to the baseband. The received samples are then sent to the NI PXIe-8133, an RTOS-based controller. Except for the experiment done in Section 4.2.1, all the results of Puzzle are obtained based on the PSD of 10240 received samples with QPSK modulation.

4.2 Performance Evaluation

4.2.1 Entropy and mismatch rate

We first compare Puzzle with the frequency domain secret key generation method with 2-bit quantization [5], which in the rest of this paper is referred as the CSI-2bit. We choose CSI-2bit as the basis for bit mismatch rate and entropy comparison because, to the best of our knowledge, it achieves the highest bit generation rate along with a low mismatch rate. We conducted an experiment where packets were transmitted over coherence time using OFDM in a 20MHz band, with each OFDM symbol consisting of 72 subcarriers. A channel frequency response is extracted from each OFDM subcarrier. The same CFR was used in both Puzzle (to construct curves) and CSI-2bit (to quantize the response). By dividing the curve composed of the 72 channel frequency responses into a certain number of segments of even length for Puzzle, and by selecting a certain number of frequency responses evenly from all the 72 subcarriers for CSI-2bit, we extracted the respective secrets from each packet for the two methods, thus obtaining secrets of different lengths. Note that in puzzle certain segment is matched to one of three cases. However, in CSI-2bit each segment is encoded into four states. We calculate the number of extracted bits based on the exact number of states. For example, in Puzzle, a 5-segment band will produce a secret of $5\times\log(3) \approx 8$bits. We compare this secret with a 8-bit secret generated by CSI-2bit. Entropy of the agreed secrets is calculated in accordance with the distribution of those secrets produced by two ends.

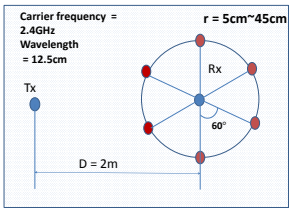


(a) Mismatch rate of different bit generation rates.

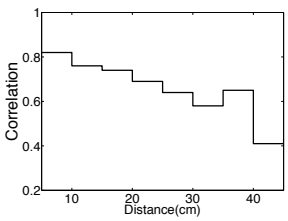
(b) Puzzle produces a comparable amount of entropy as CSI-2bit.

Figure 3: Bit Mismatch Rate and Entropy

Fig. 3a shows that Puzzle outperforms CSI-2bit in bit mismatch rate for bit generate rates from 8bit/pkt to 56bit/pkt. On average, Puzzle has a 63% lower bit mismatch rate than CSI-2bit. It is worth noting CSI-2bit has an option of online device calibration but that procedure requires the two communicating nodes collect CSI over hundreds of coherence intervals, therefore it has high overhead and is not practical for fast secret sharing. Fig. 3b shows that in both methods, the entropy of the generated bits does not increase linearly with the number of bits used to encode them. This is caused by the fact that neighboring subcarriers are correlated. For example, for a 14-bit code generated by Puzzle or CSI-2bit, the real secret contained in it, is not longer than 5-bit. Also



(a) Deployment of correlation experiment.



(b) The correlation of two codes generated by Puzzle relative to distance.

Figure 4: Deployment and result of correlation experiment

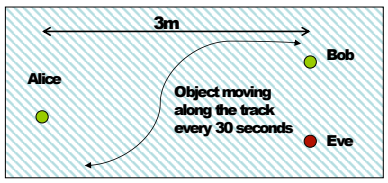
the entropy is saturating as the bit generation rate increases. Therefore, the claim made by Liu et al [5] that, CSI-2bit can generate 60-bit secret per packet by using 30 subcarriers in a 20MHz band, is not correct. Furthermore we can see that Puzzle produces a comparable amount of entropy as CSI-2bit. This implies that Puzzle does not produce more correlated bits than CSI-2bit.

4.2.2 Correlation of codes relative to distance

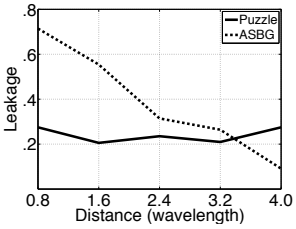
To evaluate the resistance to eavesdropping, we establish the correlation of bits generated by two receivers at different distances. We performed an experiment where we fixed the distance between one transmitter and one receiver, and then placed another receiver at a certain distance away from the first receiver along 6 orientations as shown in Fig. 4a. Each frequency response curve is segmented into 4 pieces. We measured the correlation between the codes produced by the two receivers at distances ranging from 5cm to 45cm. From Fig. 4b, we can see that the correlation decreases rapidly as the distance between two receivers increases. In practice, it is reasonable to assume that eavesdroppers are beyond one meter away, otherwise they suffer from high risk of exposures. Therefore Puzzle is robust against eavesdropping.

4.2.3 Leakage

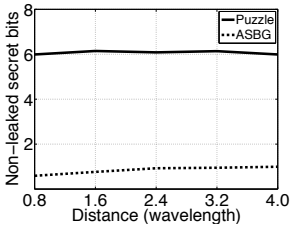
Towards validating the resistance to the planned movement attacker (cf. Section 2), we compared the leakage performance of the state-of-the-art RSS-based method ASBG and Puzzle by moving an object across the transmission path between Alice and Bob, while placing an eavesdropper near Bob, as shown in Fig. 5a. Since ASBG like many other RSS-based methods asks the two communicating ends to drop some RSS values based on certain thresholds and to exchange the indices of those values, Eve knows exactly which RSS probe is used by Bob but dropped by herself. In this case, we assume that Eve makes a random guess as to the quantization result with a success rate of 50%. We calculate the mismatch rate of Eve's and Bob's bits to be the combination of the actual mismatch rate between them and the failure rate of the random guess. And again, we segment the frequency response curves into four pieces. Fig 5b shows the leakage of our algorithm against that of ASBG over a distance from 10 cm to 50 cm. It is clear that Puzzle is much more insensitive to the threat of planned movement. Furthermore, due to the fact that Puzzle has a much higher secret generation rate ($4 * log(3) \approx 6.3$ bits/pkt) than ASBG (1 bits/pkt), the non-leaked secret produced by Puzzle is much larger, as shown in Fig 5c. Note that although



(a) An object moves between Alice and Bob with a certain temporal pattern. Eve overhears the transmission from Alice to Bob.



(b) Leakage relative to distance between Bob and Eve. Puzzle has a stable low leakage rate irrespective of the distance.



(c) Non-leaked secrete bits produced by Puzzle and ASBG per packet, relative to the distance between Bob and Eve.

Figure 5: Performance: Leakage

4 wavelength might not sound like a large distance in practice, our blocking object is not large either. The variations induced by larger obstacles, like a train passing by, might impact a much larger distance in practice.

References

[1] Argyraki, K., et al. Creating secrets out of erasures. In *MobiCom '13*.

[2] Azimi-Sadjadi, et al. Robust key generation from signal envelopes in wireless networks. In *CCS '07*.

[3] Cleveland, W. S. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association 74* (1979), 829–836.

[4] Jana, S., et al. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *MobiCom '09*.

[5] Liu, H., et al. Fast and practical secret key extraction by exploiting channel response. In *INFOCOM '13*.

[6] Madiseh, M. G., et al. Secret key generation and agreement in UWB communication channels. In *GLOBECOM '08*.

[7] Mathur, S., et al. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *MobiCom '08*.

[8] Patwari, N., et al. High-rate uncorrelated bit extraction for shared secret key generation from channel measurements. *IEEE Transactions on Mobile Computing 9*, 1 (Jan. 2010).

[9] Pinto, P. C., et al. Wireless physical-layer security: The case of colluding eavesdroppers. In *ISIT '09*.

[10] Rappaport, T. *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall PTR, 2001.

[11] Wien, T. U., et al. Computing discrete Fréchet distance. Tech. rep., 1994.

[12] Wilson, R., et al. Channel identification: Secret sharing using reciprocity in UWB channels. *IEEE Transactions on Information Forensics and Security* (2007), 364–375.

[13] Ye, C., et al. On the secrecy capabilities of ITU channels. In *VTC Fall* (2007), IEEE, pp. 2030–2034.

[14] Zeng, K., et al. Exploiting multiple-antenna diversity for shared secret key generation in wireless networks. In *INFOCOM '10*.