High Rate Uncorrelated Bit Extraction for Shared Secret Key Generation from Channel Measurements

Neal Patwari, Jessica Croft, Suman Jana, and Sneha Kasera
University of Utah, Salt Lake City, USA

Abstract—Secret keys can be generated and shared between two wireless nodes by measuring and encoding radio channel characteristics without ever revealing the secret key to an eavesdropper at a third location. This paper addresses bit extraction, *i.e.*, the extraction of secret key bits from noisy radio channel measurements at two nodes such that the two secret keys reliably agree. Problems include (1) non-simultaneous bi-directional measurements, (2) correlated secret bit streams, and (3) low secret bit rate. This paper introduces high rate uncorrelated bit extraction (HRUBE), a framework for interpolating, transforming for de-correlation, and encoding channel measurements using a multi-bit adaptive quantization scheme which allows multiple bits per component. We present an analysis of the probability of secret bit disagreement, and we use experimental data to demonstrate the HRUBE scheme and to quantify its experimental performance. As two examples, the implemented HRUBE system can achieve 22 bits per second at a bit disagreement rate of 2.2%, or 10 bits per second at a bit disagreement rate of 0.54%.

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

1 Introduction

This paper investigates the generation of shared secret keys from the observation and processing of reciprocal radio channel properties. Shared secret keys are necessary for private communication over a open channel. Public key cryptography has been the most common method for the establishment of such keys, but concerns about its limitations has spawned interest in new methods for key sharing. For example, quantum cryptography [1], [2], [3] does not use public keys, but is prohibitively expensive for most applications. Shared secret key generation from radio channel measurements, on the contrary, is very inexpensive and can be done with any standard radio devices which can receive and transmit on the same frequency channel. We envision its application in mobile and portable radio communications systems, such as IEEE 802.11 or 802.15.4, which communicate on time-division duplex (TDD) channels.

Shared secret key generation from channel measurements is an application which benefits from the randomness of the multipath channel. It would not, for example, work in a truly free-space environment (such as deep space radio links). Secret sharing benefits from:

- Reciprocity of the wireless radio channel: The multipath properties of the radio channel (gains, phase shifts, and delays) at any point in time and on any given frequency channel are identical on both directions of the link.
- 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 wireless device in order to generate more temporal variation.
- Spatial variations: The properties of the radio channel are unique to the locations of the two endpoints of the link. An eavesdropper at a third location more than a few wavelengths from either endpoint will measure a different, uncorrelated radio channel [4].

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 a to b as for signals sent from b to a.

Although the radio channel is reciprocal, *measurements* of the radio channel are not reciprocal. Additive noise contributes to each measurement as it does in any received signal. Also, the transceiver hardware used by the two nodes are not identical and affect the signal in each direction in a different way. Furthermore, measurements in both directions of the link cannot typically be made simultaneously, as addressed in Section 4.

Finally, interference power is asymmetric. The proposed system is susceptible to denial-of-service by jamming, in the same way that the wireless link is susceptible to jamming. If nodes cannot communicate, then they also cannot measure signal strength and share a

- *Neal Patwari and Jessica Croft are with the University of Utah Department of Electrical and Computer Engineering.* *Neal Patwari is partially supported by a NSF Career Award #0748206.*
- *Suman Jana and Sneha Kasera are with the University of Utah School of Computing.* *Sneha Kasera is partially supported by ONR/ARL MURI grant #W911NF-07-1-0318.*
- *All authors are partially supported by NSF CyberTrust Grant #0831490 and a Technology Commercialization Project Grant from the University of Utah Research Foundation.*

secret key. However, if multiple-access interference is infrequent, and two nodes can receive many packets from each other, they will have many measurements of signal strength, marginally impacted by interference, with which to encode a secret key. This assumes that an acknowledgement protocol is applied so that two nodes agree on which packets are to be used in the proposed system.

We refer to these sources of non-reciprocity collectively as 'noise' because they are the ultimate cause of bit disagreements between the secret keys generated at nodes a and b.

Bit extraction, *i.e.*, the extraction of secret key bits from noisy radio channel measurements at two nodes such that the two secret keys reliably agree, is a major statistical signal processing problem in shared secret key generation. As opposed to communications signal processing, it has no interest in obtaining the transmitted data from another device. We refer to this problem as a *radio channel signal processing* problem since measurements of the radio channel are the signal of interest. This paper contributes a statistical framework and algorithm for bit extraction which extracts a high bit rate with given reliability and ensures a bit vector with nearly zero correlation. This framework is a significant improvement on the state-of-the-art in the research area.

Recent results have both suggested and demonstrated bit extraction from a variety of different radio channel measurement modalities (*e.g.*, time delay, amplitude, phase, and angle). These results are reviewed in Section 2. Several works limit the number of bits per measurement to one or zero. Several works have decreased the measurement rate because correlation between measurements leads to correlation between bits in the secret key, which is detrimental to the security of data encoded with that key. Both compromises reduce the generated secret bit rate. In addition, solutions are ad hoc; each measurement modality requires a separate methodology for secret key generation.

This work provides a framework for bit extraction using three signal processing methods:

1) Fractional interpolation: Introduce different fractional delays at each node to account for the fact that the two directional measurements are not measured simultaneously.
2) De-correlation transformation: Produces a measurement vector with uncorrelated components via a Karhunen-Loève transformation of the original channel measurement vector.
3) Multi-bit adaptive quantization (MAQ): Converts real-valued channel measurements into bits adaptively based on the measured value, using communication so that both nodes agree on the quantization scheme.

The flow chart of the proposed method is shown in Figure 1. In this paper, we use these procedures in order to transform correlated, real-valued radio channel signal measurements at two nodes into uncorrelated binary data which has a high probability of bit agreement. We refer to the combination of the methods as *high rate uncorrelated bit extraction* (HRUBE).

As discussed in Section 2, there are methods, called information reconciliation methods, to resolve bit disagreements between two nodes without giving away the entire secret key. These methods do give away some information to an eavesdropper; if enough information can be obtained, an eavesdropper could perform a brute-force search to find the secret key. It is best to minimize the number of bit disagreements and to know *a priori* the probability of bit disagreement so that an information reconciliation method can be designed efficiently. This paper provides a theoretical framework to design systems with low probability of bit disagreement.

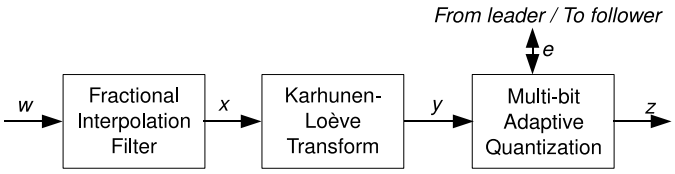

Fig. 1. Flow chart of high rate uncorrelated bit extraction.

This paper is organized as follows. In Section 2, we discuss the work in the area of secret key extraction from radio channel measurements, and position this work in that frame. In Section 3 we provide an adversary model. In Section 4, we describe the fractional interpolation method used to correct for non-simultaneous radio channel measurements. Section 5 presents the decorrelation of the measured radio channel signal, and Section 6 presents the multi-bit adaptive quantization method. Sections 4 through 6 provide the methodology and analysis of high-rate uncorrelated bit extraction. In Section 7, the HRUBE method is implemented in wireless nodes and the bit disagreement rate is shown and compared to the analytical results. Finally, future work and conclusions are presented in Section 8.

2 Related Work

There have been several papers on the topic of secret key generation from radio channel measurements. In the earliest work [5], it was suggested to send two unmodulated continuous wave (CW) signals in both directions through a channel and measure and quantize the phase difference between the two at each end of the link to generate a shared secret. Phase differences between multiple channels has been further explored in [6], [7].

Time delay and gain are also features of the radio channel that are reciprocal and can be used for secret generation. The impulse response, in particular, the amplitude of multipath at many time delays, can be used as a shared secret [8], [9], [10]. While [8], [10] use ultra-wideband (UWB) radios to measure the impulse response, [9] estimates channel gains and delays from relatively narrowband cellular signals.

Amplitude or channel gain is the most common reciprocal channel feature used for secret generation in the literature [11], [12], [13], [14], [15], [16], [17]. Amplitude can be more easily measured than time delay or phase on most existing hardware, and thus is more readily applicable to common wireless networks.

Angle-of-arrival (AOA) itself is not reciprocal – the AOA at at the two ends of a link is different. However, steerable directional antennas can be used as in [14] to measure reciprocal channel gains, which are reciprocal, and generate shared secret keys. The antenna used in [14] is a relatively simple directional antenna, but requires access points to have additional physical space for a multi-element antenna.

In Section 7, we use measurements of amplitude for its ease of implementation. However, the HRUBE methods are generally applicable across channel measurement modalities discussed above.

A critical part of practical systems will be the ability to generate arbitrarily long secret keys. Research has addressed the use of multiple measurements to increase the size of the secret. As mentioned, [14] used multiple different beam patterns to allow for multiple measurements. Multiple measurements can also be made at many different frequencies [12]. Multiple measurements over time are used in [11], [13], [15], [6], [16], [17] to increase the number of bits in the secret. In this work, we provide a method to de-correlate an arbitrary vector of collected measurements. Previous works have often used correlated measurements to generate the secret key, which may be less robust to attacks.

Significant research has addressed the more general topic of obtaining a shared secret key from observations of correlated random variables [18], [19]. An eavesdropper can also observe a correlated random variable, and the secrecy information rate is shown to be a function of the mutual information between these three observed random variables. Analysis of the secrecy rate for UWB channels in [8], [10] and for fading channels in [13] have applied this analysis to the case in which the correlated random variables correspond to measured radio channel characteristics. Information theoretic results have shown that two nodes cannot achieve an arbitrarily small bit disagreement rate without communicating some information over a common channel [18].

Because of these results, most reported research has used limited feedback [11], [9], [10], [13], [14], [6], [17] to correct small amounts of disagreement between the secret generated a the two ends of a link. This communication is referred to as 'information reconciliation', 'public feedback', or 'error correction'. In this paper we assume that information reconciliation will be part of the system design, but we do not explore its use. We explore minimizing the rate of bit disagreement in order to reduce the quantity of information reconciliation that must be performed in order to reliably agree on a shared secret key.

As an alternative to information reconciliation, [7] suggest that when bit disagreements are seen in a secret key, nodes should simply regenerate a new secret until the secret key agrees. The work in [7] explores the energy tradeoff between secret regeneration and transmit power (to increase the SINR at the ends of the link to reduce the probability of bit disagreement).

The majority of reported research has addressed system development from simulation and analysis using standard statistical channel models [8], [7], [11], [9], [10], [13], [15], [6]. Statistical channel fading models such as Rayleigh, Ricean, and Gaussian have been applied. These models can be used to compute the probabilities of secret bit agreement, the probabilities that retransmission is necessary, and the secret rate which can be generated either in theory or by a particular encoding algorithm.

Notably, several researchers have reported experimental results and implementations [10], [12], [14], [16]. In [10], several bi-directional UWB measurements are made and used to compute the number of secret bits which could be generated. In [12], 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. Two implementations in [16] use both off-the-shelf and testbed 802.11a devices to implement secret sharing and achieve about a 1 secret bit per second (s-bits/sec) rate. In [14], 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. This work also demonstrates its techniques using a Zigbee radio implementation, but the techniques reported here are fundamentally unlike those reported in [14]. This paper reports in detail a new method for reliably estimating a high-rate uncorrelated bit stream from radio channel measurements.

We have not seen any secret bit rate reported as high as achieved in our experiments, up to 22 bits/sec. In [16], an experimental implementation achieves about a 1 bit/sec rate, for which the authors target a 10^{-8} bit disagreement rate. In this paper, we report experimental bit rates from 3 bits/sec at the lowest probability of bit disagreement (0.04%) to a rate of 22 bits/sec at the highest probability of bit disagreement (2.2%). There is a tradeoff in secret key generation between high secret bit rate and low probability of bit disagreement, and the proposed method can be used to provide a variety of achieveable points, particular at higher secret bit rates than previously reported.

3 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 between herself and Bob at the same time when Alice and Bob measure the channel between them 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 [16]) to either Alice or Bob while they are extracting their shared key. This will ensure that Eve measures a different, uncorrelated radio channel [4]. We assume that Eve cannot jam the communication channel between Alice and Bob. We also assume that Eve cannot cause a man-in-the-middle attack, *i.e.*, our methodology does not authenticate Alice or Bob. In other words, the proposed system works against passive adversaries. In this aspect, the technique of key extraction from wireless signal strengths is comparable with classical key establishment techniques such as Diffie-Hellman, which also use message exchanges to establish keys and do not authenticate Alice or Bob. However, classical key exchange techniques make use of unproven assumptions about computational hardness of different problems such as the discrete logarithm problem. In contrast, the proposed key extraction technique provides information-theoretic secrecy and does not assume any bounds on the computation power of the adversary.

Even without an authentication mechanism, the Diffie-Hellman 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.*, [17], [20]). These and future authentication mechanisms can be used in conjunction with our secret key establishment scheme.

4 Fractional Interpolation Filtering

Channel measurements for secret key generation will almost certainly be half-duplex, that is, the transmission from node a to node b is not made at the same time as the transmission from node b to node a. Standard transceivers cannot transmit and receive simultaneously, so non-simultaneous measurements in the two directions must be dealt with regardless of the type of radio channel measurements made. As described in Section 2, many methods use sequential measurements over time in order to generate arbitrarily long secret keys. In this paper, we use fractional interpolation filters to allow nodes to estimate what measurements would have been if they had been measured simultaneously.

For channel measurement, we assume nodes repeatedly transmit packets in a TDD protocol [1] Each node receives a packet from the other node and uses it to measure the channel characteristic. Let $\mathbf{w}_a(i)$ be the ith channel measurement at node a made at time $\tau_a(i)$. We assume that the constant packet rate means the ith measurement is made at time $\tau_a(i) = \tau_a(i-1) + T_{R,a}$ where T_R is the time since the previous measurement. Similarly, node b makes ith measurement $\mathbf{w}_b(i)$ at time $\tau_b(i) = \tau_b(i-1) + T_{R,b}$.

We assume that channel measurements are made at greater than the Nyquist rate for the channel. For a fading channel, the Nyquist rate is twice the maximum Doppler frequency, f_d. Previous works have assumed that channel probes in the two directions are much smaller than $1/f_d$ [16]. In this work, probing requirements are relaxed somewhat because the signal can be reconstructed at simultaneous probe times as long as the probes are taken more often than $\frac{1}{2f_d}$, *i.e.*, we always have $T_{R,c} < \frac{1}{2f_d}$ for $c \in \{a, b\}$.

In general, given that samples are taken at least as fast as the Nyquist rate, interpolation can be performed regardless of whether $T_{R,a}$ and $T_{R,b}$ are constant over time, or if $T_{R,a} \neq T_{R,b}$. Thus multiple-access delays, or delays due to retransmission, are acceptable within a limit. As a simple example, one might perform linear interpolation between samples to approximate the value of $\mathbf{w}_a$ at time τ with $\tau_a(i-1) \leq \tau \leq \tau_a(i)$ as

$$\mathbf{w}_a(i-1) + [\tau - \tau_a(i-1)]\frac{\mathbf{w}_a(i) - \mathbf{w}_a(i-1)}{\tau_a(i) - \tau_a(i-1)} \qquad (1)$$

If nodes agreed to interpolate to half-way between their probing times, then both nodes would use $\tau = \frac{1}{2}[\tau_a(i) + \tau_b(i)]$ in (1. Since each node would be able to record transmission or reception times, $\tau_a(i)$ and $\tau_b(i)$ for all i, on its own clock, no explicit synchronization is required in this procedure. Other approximation algorithms can use τ_a and $\mathbf{w}_a$ from multiple recent samples to perform higher-order interpolation.

Here, for computational simplicity, we choose to implement a protocol which allows nodes a nearly constant packet rate. Thus we assume $T_R = T_{R,a} = T_{R,b}$ for all time. We define two fractional interpolation filters, one for each node. We define the fractional sampling offset μ as

$$\mu = \frac{1}{2}\left[\frac{\tau_b(i) - \tau_a(i)}{T_R}\right]. \qquad (2)$$

Without loss of generality, assume that $\tau_a(i) < \tau_b(i)$, so that $\mu > 0$. If node a delayed its ith sample by $(1+\mu)T_R$, and simultaneously, node b delayed its samples by $(1-\mu)T_R$, we would have had simultaneous measurements. To estimate these delayed samples, we specify a fractional delay μ_c and integer delay $\bar{m}_c$ for each node:

$$\mu_a = \mu \qquad \mu_b = 1 - \mu$$
$$\bar{m}_a = 1 \qquad \bar{m}_b = 0$$

After using a fractional interpolation filter to delay its signal by $\mu_a = \mu$, node a will also add an additional unit sample delay. Node b uses its fractional interpolation filter to delay its signal by $\mu_b = 1 - \mu$, and does not add any additional unit delays.

1. Whether or not the transmission is a data packet or any arbitrary finite-duration signal, we use the term 'packet' to describe the transmission.

We use the Farrow filter, a finite impulse response implementation which introduces an arbitrary fractional delay [21]. Standard implementations of the Farrow filter are four-tap FIR filters which provide a parabolic or cubic interpolation between samples. Such filters are often used in discrete-time implementations of receiver timing synchronization loops, and are well-suited for DSP implementation [22]. We implemented a cubic Farrow filter, parameterized by μ_a or μ_b. For $c \in \{a, b\}$,

$$\begin{aligned}\mathbf{h}_c &= \Big[\frac{\mu_c^3}{6} - \frac{\mu_c}{6},\ -\frac{\mu_c^3}{2} + \frac{\mu_c^2}{2} + \mu_c, \\ &\quad \frac{\mu_c^3}{2} - \mu_c^2 - \frac{\mu_c}{2},\ -\frac{\mu_c^3}{6} + \frac{\mu_c^2}{2} - \frac{\mu_c}{3}\Big]^T \end{aligned} \quad (3)$$

Note that $\mathbf{h}_c$ requires recalculation only when μ_c changes. The input of filter $\mathbf{h}_c$ are the measurements $\{\mathbf{w}_c(i)\}_i$, and we define the output as $\{\mathbf{x}_c(i)\}_i$. These outputs are estimates of the radio channel at simultaneous instants, halfway between the original non-simultaneous measurements. These vectors $\mathbf{x}_a$ and $\mathbf{x}_b$,

$$\begin{aligned}\mathbf{x}_a &= [\mathbf{x}_a(1)^T, \ldots, \mathbf{x}_a(N)^T]^T \\ \mathbf{x}_b &= [\mathbf{x}_b(1)^T, \ldots, \mathbf{x}_b(N)^T]^T,\end{aligned}$$

where N is the desired length of the vector. These vectors become the input for the de-correlation transformation discussed in the following section.

5 De-Correlation Transformation

In this paper, we use the discrete Karhunen-Loève transform (KLT) in order to convert the measured channel vectors $\mathbf{x}_a$ and $\mathbf{x}_b$ into uncorrelated components. The KLT has been applied for many different types of signals for purposes of noise reduction and data compression. We apply the KLT for the purpose of generating nearly uncorrelated elements for our secret, which for robustness to attacks, should not contain significant correlation between elements.

The discrete KLT provides an orthogonal basis which de-correlates the input vector, assuming a known model for the covariance structure of the original vector. For particular classes of signals, we can find such statistical models; *e.g.*, for electrocardiogram signals [23], voice signals, internet traffic measurements [24], and fingerprints [25], models have been developed from large sets of measurements. In this paper, we develop such a covariance model using a large set of measurements, and use it to calculate the appropriate KLT.

In the discrete KLT, a linear transformation of an input vector is taken, which results in a vector with uncorrelated elements. Assume that the (length N) input vector at node $c \in \{a, b\}$, $\mathbf{x}_c$, has mean μ_c and covariance matrix $R_{\mathbf{x}}$ [2]. A linear transform of the data is

$$\mathbf{y}_c = A^T(\mathbf{x}_c - \mu_c),$$

where A is an $N \times N$ matrix. The mean of $\mathbf{y}_c$ is zero, and the covariance matrix of $\mathbf{y}_c$, $R_{\mathbf{y}}$, is given by

$$R_{\mathbf{y}} = E\left[\mathbf{y}_c \mathbf{y}_c^T\right] = A^T R_{\mathbf{x}} A \quad (4)$$

Assume the singular value decomposition of $R_{\mathbf{x}}$ is,

$$R_{\mathbf{x}} = USU^T, \quad (5)$$

where U is the matrix of eigenvectors, and $S = \text{diag}\left\{\sigma_1^2, \ldots, \sigma_N^2\right\}$, a diagonal matrix of the corresponding eigenvalues. We assume that the eigenvectors have been sorted in order of decreasing eigenvalue, so that $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_N^2 \geq 0$. Note that $U^T U = \mathrm{I}_N$, where I_N is the $N \times N$ identity matrix.

The discrete Karhunen-Loève transform simply assigns $A = U$,

$$\mathbf{y}_c = U^T(\mathbf{x}_c - \mu_c). \quad (6)$$

The vectors $\mathbf{y}_a$ and $\mathbf{y}_b$ for nodes a and b, respectively, become the input for the MAQ scheme described in Section 6. When simplifying (4) we have the result that,

$$R_{\mathbf{y}} = U^T R_{\mathbf{x}} U = U^T USU^T U = S,$$

and the output vector $\mathbf{y}_c$ in fact has a diagonal covariance matrix, indicating uncorrelated elements.

Note that uncorrelated is not the same as independent; while they are the same for Gaussian random vectors, they are not equivalent in general. The KLT guarantees zero covariance between elements, but not higher-order cross-moments. Uncorrelated but not independent bits may result in an a per bit entropy of less than 1.0; we show in Section 7.7 from experiments an entropy estimate of between 0.96 to 0.98. Depending on the true joint distribution, it may be possible for an attacker to use the higher-order cross-moments to predict one value from others. Investigation of this possibility must involve extensive measurements to allow inference on the joint distribution of the data $\mathbf{y}_c$, and we leave this to future research.

5.1 Bi-Directional Measurement Covariance

Although the elements of $\mathbf{y}_c$ are decorrelated by the KLT, there is still covariance between the $\mathbf{y}_c$ and $\mathbf{y}_{\bar{c}}$, where $c, \bar{c} \in \{a, b\}$ and $\bar{c} \neq c$. That is, c and $\bar{c}$ are the indices for the opposite direction measurements of the link between nodes a and b. In fact, a high positive correlation between the two different directions of the link is what enables secret sharing. We denote the covariance matrix of the original measurements to be $R_{\mathbf{x}_c, \mathbf{x}_{\bar{c}}}$,

$$R_{\mathbf{x}_c, \mathbf{x}_{\bar{c}}} \triangleq E\left[\mathbf{x}_c \mathbf{x}_{\bar{c}}^T\right] = E\left[\mathbf{x}_{\bar{c}} \mathbf{x}_c^T\right].$$

After the KLT, the vectors $\mathbf{y}_c$ and $\mathbf{y}_{\bar{c}}$ have covariance matrix denoted as $R_{\mathbf{y}_c, \mathbf{y}_{\bar{c}}}$,

$$R_{\mathbf{y}_c, \mathbf{y}_{\bar{c}}} \triangleq E\left[\mathbf{y}_c \mathbf{y}_{\bar{c}}^T\right] = U^T R_{\mathbf{x}_c, \mathbf{x}_{\bar{c}}} U. \quad (7)$$

The ith diagonal element of $R_{\mathbf{y}_c, \mathbf{y}_{\bar{c}}}$, denoted here as $[R_{\mathbf{y}_c, \mathbf{y}_{\bar{c}}}]_{i,i}$, is the covariance of $y_c(i)$ and $y_{\bar{c}}(i)$. The variance of $\mathbf{y}_c$ is given by σ_i^2 and is equal to the variance of

2. We assume that the mean at each node can be different due to the different transmit powers, but that the covariance matrix is the same at both nodes.

$\mathbf{y}_{\bar{c}}$ since $c, \bar{c} \in \{a, b\}$. So the correlation coefficient of the ith component, denoted ρ_i, is

$$\rho_i^2 = \frac{[R_{\mathbf{y}_c,\mathbf{y}_{\bar{c}}}]_{i,i}}{\sigma_i^2}. \quad (8)$$

The correlation coefficient ρ_i is effectively a measure of the SNR of the measurement of the bi-directional ith component of $\mathbf{y}$. When the 'noise' contributing to both $\mathbf{y}_a(i)$ and $\mathbf{y}_b(i)$ is high, the value of $[R_{\mathbf{y}_c,\mathbf{y}_{\bar{c}}}]_{i,i}$ is low compared to σ_i^2, and ρ_i is closer to zero. When there is very little noise, ρ_i is close to 1. In Section 6.4, we show that the value of ρ_i is the critical component to determine both how many bits to which the ith component can be quantized, and the performance of the quantization method, *i.e.*, the probability that the bits generated agree at the two nodes.

6 Multi-bit Adaptive Quantization

The objective of this section is to describe the quantization of the transformed vector $\mathbf{y}_c$, for $c \in \{a, b\}$, into a secret bit vector. Quantization in this application has two conflicting goals:

1) *Secret length maximization*: Obtain as long of a secret (in bits) as possible.
2) *Error minimization*: Keep the probability that the secret key will not match at nodes a and b as low as possible.

We also must maintain zero covariance between elements of secret key, and thus do not use vector quantization. Instead, we use scalar quantization on each element of $\mathbf{y}_c$. Consider this tradeoff on a single component, component i. The quantizer for component i is a function $Q_i : \mathbb{R} \to \{1, \ldots, 2^{m_i}\}$, where m_i is the number of bits to which we quantize the ith component.

6.1 Related Work: Censoring Scheme

Several past works in bit extraction from channel measurements have quantized each measurement to one of two bins and a 'censor' region [16], [17], [14], [15]. In these methods, when a measurement has a value near zero, the measurement is not used in the secret, and otherwise, the measurement is quantized by its sign to a '0' or '1'. This one bit and censor method, as a standard method used in related work, provides a good point of comparison for the MAQ method presented in this paper.

In the standard censoring method, a low threshold and a high threshold are specified, for example, $[-\gamma, \gamma]$. The indices of the values that fall within the threshold region are not encoded. Specifically, node a forms the set $\mathcal{T}_a = \{i : -\gamma \le x_a(i) \le \gamma\}$ and transmits the list of the elements in $\mathcal{T}_a$ to node b. Node b then similarly forms $\mathcal{T}_b$ and transmits $\mathcal{T}_b \setminus \mathcal{T}_a$ to node a. The union of both sets, $\mathcal{T} = \mathcal{T}_a \cup \mathcal{T}_b$ are the indices not used in the shared secret. Let $\bar{\mathcal{T}} = \{1, \ldots, N\} \setminus \mathcal{T}$ be the indices that will be used in the secret, and let $t_j \in \{1, \ldots, N_{\mathcal{T}}\}$ be the jth element of $\bar{\mathcal{T}}$, where $N_{\mathcal{T}} = |\bar{\mathcal{T}}|$. Then the secret bit vector of node $c \in \{a, b\}$ is given by $\mathbf{z}_c = [z_c(1), \ldots, z_c(N_{\mathcal{T}})]^T$, where

$$z_c(j) = \begin{cases} 1, & x_c(t_j) > \gamma \\ 0, & x_c(t_j) < -\gamma \end{cases} \quad (9)$$

We note that the work in [16] does not directly encode $x_c(i)$. Instead, only one sample from each 'excursion' of $\{x_c(i)\}_i$ is encoded. An excursion is any sequence of $\{x_c(i)\}_i$ which are either all above the high threshold or below the low threshold. This method allows high reliability in bit encoding while reducing correlation between subsequent secret bits.

The benefit of the one bit and censor scheme in general is that data that falls near zero, which would be likely to have opposite sign at the opposite end of the link, is not used in the secret. The value of $y_a(i)$ would be known to an eavesdropper to be either within or outside of the threshold region, but this provides no information about whether the value is above γ or below $-\gamma$. If $i \in \mathcal{T}$, the measurement is not used in the secret, and if $i \notin \mathcal{T}$, the eavesdropper has no information about whether $y_a(t_j)$ is greater than γ or less than $-\gamma$. So, the communicated data does not reveal anything about the secret bits.

Compared to the method proposed in this paper, the censoring scheme has two main drawbacks. First, the use of censoring causes loss of bits. The number of bits lost to censoring depends on the measured data. In order to ensure a secret with a consistent number of bits, extra measurements must be gathered prior to bit extraction. In numerical results related in Section 7.6, between 5% to 27% of measurements are lost to censoring. The second drawback is that it is not possible to generate more than one bit from each real-valued measurement, that is, $m_i = 1$ for all i.

6.2 Formulation

We propose a multi-bit adaptive quantization (MAQ) scheme for secret sharing. The MAQ scheme adaptively quantizes each measurement to an arbitrary number of bits without censoring. No fixed quantization scheme is able to achieve a low error rate because when $y_a(i)$ is very near to a threshold, there is a high probability that $y_b(i)$ crosses to the other side of that threshold. As a solution, we propose to change the quantization scheme at both a and b based on the measurement at one of the nodes. For this discussion, without loss of generality, we assume that node a is the 'leader' node in the multi-bit adaptive quantization scheme, and that node b is the 'follower'.

In the MAQ scheme, we first quantize $y_a(i)$ to $K \triangleq 2^{m_i+2} = 4 \times 2^{m_i}$ equally-likely quantization levels. To achieve equally-likely quantization levels, we require the distribution for $y_a(i)$. In general, let $F_i(y) = P[y_a(i) \le y]$ be the cumulative distribution function (CDF) of $y_a(i)$. Thresholds for equally likely quantization bins are generated by using the inverse of the CDF,

$$\eta_k = F_i^{-1}\left(\frac{k}{4 \times 2^{m_i}}\right), \text{ for } k = 1, \ldots, K-1. \quad (10)$$

Bin k	Codeword d_1	Codeword d_0	e	Interval of $y(i)$
1	0	0	0	$(-\infty, F_i^{-1}(0.125)]$
2	0	0	1	$(F_i^{-1}(0.125), F_i^{-1}(0.25)]$
3	0	1	1	$(F_i^{-1}(0.25), F_i^{-1}(0.375)]$
4	0	1	0	$(F_i^{-1}(0.375), F_i^{-1}(0.5)]$
5	1	1	0	$(F_i^{-1}(0.5), F_i^{-1}(0.625)]$
6	1	1	1	$(F_i^{-1}(0.625), F_i^{-1}(0.75)]$
7	1	0	1	$(F_i^{-1}(0.75), F_i^{-1}(0.875)]$
8	1	0	0	$(F_i^{-1}(0.875), +\infty)$

TABLE 1
Example $m_i = 1$-bit adaptive quantization scheme.

In the following for ease of notation, let $\eta_0 = -\infty$ and $\eta_K = \infty$. The kth quantization bin is the interval $(\eta_{k-1}, \eta_k]$ for $k = 1, \ldots, K$, so $k(i)$ is given by

$$k(i) = \max_k \{k \text{ s.t. } y_a(i) > \eta_{k-1}\}. \quad (11)$$

Next, we use a particular type of Gray code to adaptively assign a binary codeword to each quantization bin. We do this by defining the following binary variables:

- Define $e(k)$, for $k = 1, \ldots, K$ as

$$e(k) = \begin{cases} 1, & k \bmod 4 \geq 2 \\ 0, & o.w. \end{cases}$$

  In other words, $e(k)$ is the twos bit in the binary representation of integer k.
- Create a Gray codeword with m_i bits, that is, an ordered list of 2^{m_i} possible m_i-bit codewords. A Gray codeword list changes only one bit between neighboring codewords in the list.
- Let $f_1(k) = \left\lfloor \frac{k-1}{4} \right\rfloor$. Define $d_1(k) \in \{0,1\}^{m_i}$ to be equal to the $f_1(k)$th Gray codeword. That is, it is the same Gray codeword list but with each element repeated four times.
- Let $f_0(k) = \left\lfloor \frac{k+1 \mod K}{4} \right\rfloor$. Define $d_0(k) \in \{0,1\}^{m_i}$ to be equal to the $f_0(k)$th Gray code. That is, it is the same list as $d_1(k)$ but circularly shifted by two.

Two examples are presented in Table 1 and Table 2, for the case of $m_i = 1$ and $m_i = 2$, respectively.

Multi-bit adaptive quantization proceeds as follows. First, from the values of $y_a(i)$, leader node a determines the quantization bin $k(i)$ for all components i. Node a transmits the bit vector $\mathbf{e} = [e(k(1)), \ldots, e(k(N))]^T$ to the follower node b. Both nodes then encode their secret key using codeword d_1 whenever $e = 1$, and codeword d_0 whenever $e = 0$. Specifically, the secret key is

$$\mathbf{z} = \left[d_{e(k(1))}(k(1)), \ldots, d_{e(k(N))}(k(N))\right]$$

where $k(i)$ is given in (11).

6.3 Discussion

The above MAQ scheme provides a new method for a multi-bit adaptive quantization for each component of vector $\mathbf{y}$. No components are 'censored', and instead, a leader node decides upon the quantization scheme from two options which is least likely to cause disagreement between the two nodes. When disagreements occur, due to the use of Gray coding, it is very likely that only one bit of the multi-bit codeword will be in error.

Bin k	Codeword d_1	Codeword d_0	e	Interval of $y(i)$
1	01	00	0	$(-\infty, F_i^{-1}(0.0625)]$
2	01	00	1	$(F_i^{-1}(0.0625), F_i^{-1}(0.125)]$
3	01	01	1	$(F_i^{-1}(0.125), F_i^{-1}(0.1875)]$
4	01	01	0	$(F_i^{-1}(0.1875), F_i^{-1}(0.25)]$
5	11	01	0	$(F_i^{-1}(0.25), F_i^{-1}(0.3125)]$
6	11	01	1	$(F_i^{-1}(0.3125), F_i^{-1}(0.375)]$
7	11	11	1	$(F_i^{-1}(0.375), F_i^{-1}(0.4375)]$
8	11	11	0	$(F_i^{-1}(0.4375), F_i^{-1}(0.5)]$
9	10	11	0	$(F_i^{-1}(0.5), F_i^{-1}(0.5625)]$
10	10	11	1	$(F_i^{-1}(0.5625), F_i^{-1}(0.625)]$
11	10	10	1	$(F_i^{-1}(0.625), F_i^{-1}(0.6875)]$
12	10	10	0	$(F_i^{-1}(0.6875), F_i^{-1}(0.75)]$
13	00	10	0	$(F_i^{-1}(0.75), F_i^{-1}(0.8125)]$
14	00	10	1	$(F_i^{-1}(0.8125), F_i^{-1}(0.875)]$
15	00	00	1	$(F_i^{-1}(0.875), F_i^{-1}(0.9325)$
16	00	00	0	$(F_i^{-1}(0.9325), +\infty)$

TABLE 2
Example $m_i = 2$-bit adaptive quantization scheme.

Further, the passing of the vector $\mathbf{e}$ does not provide an eavesdropper with any information about the secret key bits. Knowing $e(k(i))$ eliminates half of the possible quantization levels. But codewords are equally likely given the knowledge of $e(k(i))$. For example, for the $m_i = 1$ case, if $e = 1$, then the eavesdropper knows that $y_a(i)$ is neither very high in magnitude nor very low in magnitude. However, there are four possible equally-likely bins with $e = 1$, two which would be encoded with $d_1 = 1$ and two with $d_1 = 0$. The eavesdropper has no information to which bit the component will be encoded.

6.4 Analysis of Probability of Bit Disagreement

The performance of a particular MAQ scheme is measured by its probability of bit disagreement (P_{BD}), that is, the probability that nodes a and b encode a bit differently. We use the term 'disagreement' rather than 'error' because there is no notion of the 'correct' bit to which a and b should have encoded when they disagree.

In this section, we analyze bit disagreement probabilities as a function of the number of quantization bits m_i and the joint distribution of $y_a(i)$ and $y_b(i)$. Let the joint probability distribution function (pdf) be $f_{Y_a(i),Y_b(i)}(y_a, y_b)$. As discussed, the two marginal distributions are identical, so we refer to the marginal pdf as $f_i(y)$ and the cumulative distribution function (CDF) as $F_i(y)$. The conditional CDF of $y_b(i)$ given $y_a(i)$ as written as $F_{Y_b(i)|Y_a(i)}(y_b|y_a)$.

We first discuss the probability of codeword disagreement, and then provide an approximation for the probability of bit disagreement. The two are the same in the $m_i = 1$ case, but different in general.

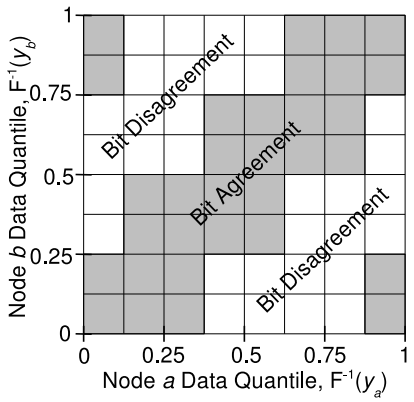

Fig. 2. Diagram showing area of $F_i^{-1}(y_a), F_i^{-1}(y_b)$ where generated bits at a and b will agree (gray area) and disagree (white area) for the 1-bit adaptive quantization scheme.

For certain combinations of $(y_a(i), y_b(i))$, the codeword at node a will be encoded differently than at node b, and these combinations can be viewed graphically on a 2-D plot. Recall that node a is the leader node, and so y_a decides the quantization scheme. Given $y_a(i)$, the value of $y_b(i)$ thus decides whether or not a bit disagreement occurs. For example, for the 1-bit adaptive quantization scheme displayed in Table 1, the combinations of $(y_a(i), y_b(i))$ which result in bit agreement are shown in Figure 2 as the gray shaded area. There is a wide diagonal area, where $y_a(i)$ is close to or equal to $y_b(i)$, which would result in codeword agreement.

If we refer to the shaded area of the diagram (the bit agreement area) as A, then the probability of codeword agreement, denoted P_{CA}, is

$$P_{CA} = \iint_A f_{Y_a(i),Y_b(i)}(y_a, y_b) dy_a dy_b.$$

Another way to write this equation is

$$P_{CA} = \int_{y_a} P\left[\text{CA}|y_a\right] f_i(y_a) dy_a \tag{12}$$

where $P\left[\text{CA}|y_a\right]$ is the probability of code agreement given y_a,

$$P\left[\text{CA}|y_a\right] = \int_{y_b \in A(y_a)} f_{Y_b(i)|Y_a(i)}(y_b|y_a) dy_b,$$

and $A(y_a) = \{y_b : (y_a, y_b) \in A\}$.

6.5 MAQ Performance in Gaussian Case

For the case that $y_a(i)$ and $y_b(i)$ are jointly Gaussian and zero mean, we can find a more direct expression for the probability of code agreement. Note that we know that the marginal variances of $y_a(i)$ and $y_b(i)$ are identical, which we denote as σ_i^2. Let the correlation coefficient of $y_a(i)$ and $y_b(i)$ be ρ_i. Then the conditional pdf of $Y_b(i)|Y_a(i)$ has mean $\rho_i Y_a(i)$ and variance $\sigma_i^2(1-\rho_i^2)$. Thus, $P\left[\text{CA}|y_a\right]$ for the 1-bit adaptive quantization case can be written as:

$$\begin{aligned} P\left[\text{CA}|y_a\right] \quad \geq \quad & \Phi\left[\frac{F^{-1}(\alpha_{y_a}) - \rho_i y_a}{\sigma_i\sqrt{1-\rho_i^2}}\right] \\ & -\Phi\left[\frac{F^{-1}(\beta_{y_a}) - \rho_i y_a}{\sigma_i\sqrt{1-\rho_i^2}}\right] \end{aligned} \tag{13}$$

where $\Phi(x)$ is the unit variance Gaussian CDF, and α_{y_a} and β_{y_a} are the high and low limits for a given y_a of the segment of y_b which results in codeword agreement. For the 1-bit adaptive quantization case, it can be seen from Figure 2 that

$$\alpha_{y_a} = \begin{cases} 0.25, & F_i(y_a) \leq 0.125 \\ 0.5, & 0.125 < F_i(y_a) \leq 0.375 \\ 0.75, & 0.375 < F_i(y_a) \leq 0.625 \\ 1, & 0.625 < F_i(y_a) \end{cases}$$

$$\beta_{y_a} = \begin{cases} 0, & F_i(y_a) < 0.375 \\ 0.25, & 0.375 \leq F_i(y_a) < 0.625 \\ 0.5, & 0.625 \leq F_i(y_a) < 0.875 \\ 0.75, & 0.875 \leq F_i(y_a) \end{cases}$$

In general, for an m_i-bit quantization scheme,

$$\begin{aligned} \alpha_{y_a} &= \min\left\{1, \left\lceil F_i(y_a) + 2^{-(m_i+2)}\right\rceil_{2^{-(m_i+1)}}\right\} \\ \beta_{y_a} &= \max\left\{0, \left\lfloor F_i(y_a) - 2^{-(m_i+2)}\right\rfloor_{2^{-(m_i+1)}}\right\} \end{aligned} \tag{14}$$

where we define the u-multiple floor and ceiling functions which return the highest multiple of u lower than its argument, and the lowest multiple of u higher than its argument, respectively:

$$\lfloor x \rfloor_u = u\left\lfloor \frac{x}{u} \right\rfloor, \quad \lceil x \rceil_u = u\left\lceil \frac{x}{u} \right\rceil \tag{15}$$

Equation (13) is less than or equal to the exact $P\left[\text{agreement}\right]$ because we only consider the main diagonal area of agreement. For example, we do not consider the top-left area and bottom-right area in Figure 2. These non-diagonal elements will typically have a very small probability, because they correspond to observing nearly opposite values on the two directional measurements. Since the two measurements have high positive correlation, it is highly unlikely to observe nearly opposite values.

Using (12) and (13) we can solve for a lower bound on the probability of agreement. We note that for the Gassian case, $F_i^{-1}(x) = \sigma_i \Phi^{-1}(x)$ where $\Phi^{-1}(x)$ is the inverse of the zero-mean unit variance CDF. A substitution of $v = y_a/\sigma_i$ results in the expression,

$$\begin{aligned} P_{CA} \quad \geq \quad & \int_{v=-\infty}^{\infty} \left\{ \Phi\left[\frac{\Phi^{-1}[\alpha_{\sigma_i v}] - \rho_i w}{\sqrt{1-\rho_i^2}}\right] - \right. \\ & \left. \Phi\left[\frac{\Phi^{-1}[\beta_{\sigma_i v}] - \rho_i v}{\sqrt{1-\rho_i^2}}\right] \right\} \frac{e^{-v^2/2}}{\sqrt{2\pi}} dv \end{aligned} \tag{16}$$

Note that (16) is not a function of σ_i, the variance of the ith component of $\mathbf{y}$. Instead, it is solely a function of ϕ_i

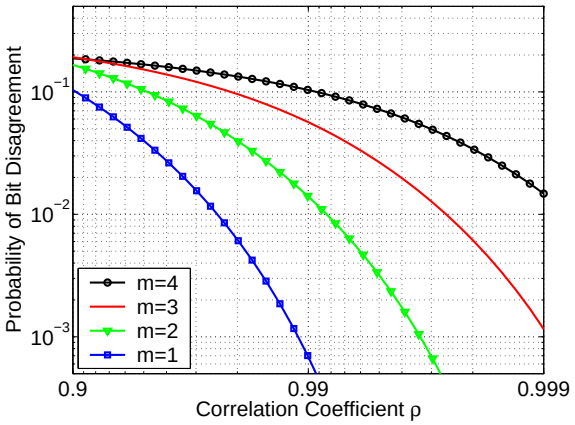

Fig. 3. Analytical approximation for the probability of bit disagreement from (17) as a function of number of bits per codeword, m, and correlation coefficient, ρ.

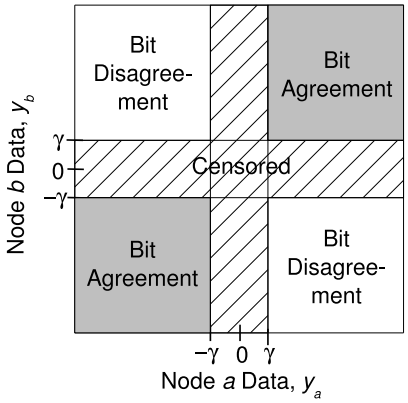

Fig. 4. Diagram showing area of (y_a, y_b) where generated bits at a and b will agree (gray area), disagree (white area), and will be censored (crosshatched area) for the censoring scheme.

and m_i. Note that the probability of code disagreement $P_{CD} = 1 - P_{CA}$, from the lower bound in (16), we have a upper bound on P_{CD}. Borrowing from digital communications analysis of Gray coded symbol constellations, for low P_{CD}, we can approximate the probability of *bit* disagreement, P_{BD}, as

$$P_{BD} \approx P_{CD}/m_i. \tag{17}$$

This expression (16) is solved numerically, and results for the P_{BD} are plotted in Figure 3.

6.6 Censoring Scheme Performance in Gaussian Case

The multi-bit adaptive quantization scheme offers the possibility of encoding a component with more than one bit, which is not possible in the existing censoring scheme. However, for the $m_i = 1$ case, we can compare the two bit extraction schemes and evaluate their relative benefits.

We begin by formulating the probability of bit disagreement in the censoring scheme. The bit is censored whenever either a or b measures a value between $-\gamma$ and γ. Agreement occurs whenever both $y_a(i) < -\gamma$ and $y_b(i) < -\gamma$, or whenever both $y_a(i) > \gamma$ and $y_b(i) > \gamma$. These cases are shown in Figure 4, which is analogous to Figure 2 for the 1-bit MAQ scheme.

Assuming a joint Gaussian distribution for (y_a, y_b), as assumed in Section 6.5, we can calculate the probability of censoring, $P[\text{Cens}]$, and the probability of bit disagreement.

$$P[\text{Cens}] = \int_v P[\text{Cens}\,|v]\,\frac{e^{-v^2/2}}{\sqrt{2\pi}}dv \tag{18}$$

$$P[\text{Cens}\,|v] = \begin{cases} 1, \text{ if } -\frac{\gamma}{\sigma_i} \le v < \frac{\gamma}{\sigma_i} \\ \Phi\left[\frac{\gamma/\sigma_i - \rho_i v}{\sqrt{1-\rho_i^2}}\right] - \Phi\left[\frac{-\gamma/\sigma_i - \rho_i v}{\sqrt{1-\rho_i^2}}\right], \text{ o.w.} \end{cases}$$

This expression is only a function of γ/σ_i and ρ, and is plotted in Figure 5. Similarly, the probability of bit

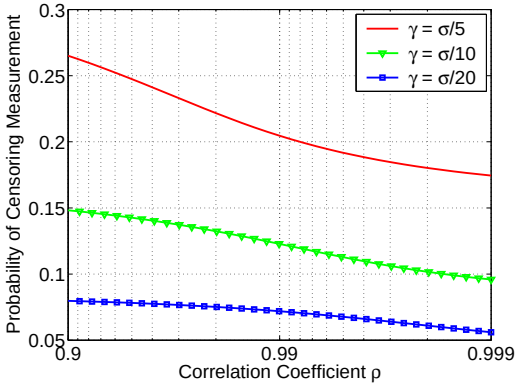

Fig. 5. Analytical probability of a bit being censored in the censoring scheme vs. ρ and γ.

disagreement for the censoring scheme (given that it is not censored) is given by,

$$P_{BD} = \frac{1}{1 - P[\text{Cens}]}\int_v P[\text{BD}\,|v]\,\frac{e^{-v^2/2}}{\sqrt{2\pi}}dv$$

$$P[\text{BD}\,|v] = \begin{cases} 1 - \Phi\left[\frac{\gamma/\sigma_i - \rho_i v}{\sqrt{1-\rho_i^2}}\right], & v < \frac{-\gamma}{\sigma_i} \\ \Phi\left[\frac{-\gamma/\sigma_i - \rho_i v}{\sqrt{1-\rho_i^2}}\right], & v > \frac{\gamma}{\sigma_i} \\ 0, & \text{o.w.} \end{cases} \tag{19}$$

The performance of the censoring scheme should be judged on the probability of bit disagreement given that the bit is used in the secret (is not censored). This is why the conditional probability of bit disagreement is divided by the factor of $(1 - P[\text{Cens}])$. The result is plotted in Figure 6.

6.7 Performance Discussion and Comparison

We can compare the results in Figure 6 to the $m = 1$ line in Figure 3. For low thresholds γ, the censoring scheme results in a higher probability of bit disagreement than the 1-bit MAQ scheme. At high γ (*e.g.*, the $\gamma/\sigma = 0.2$

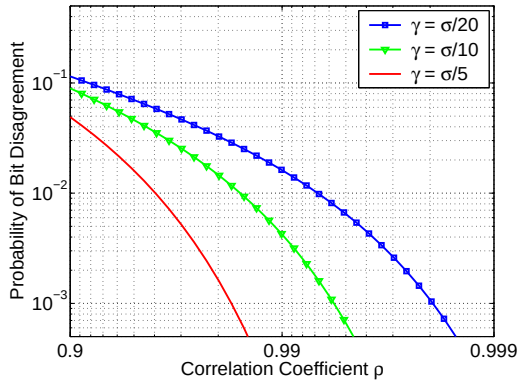

Fig. 6. Analytical probability of bit disagreement from (19) vs. ρ and censoring threshold γ, given that the bit is not censored.

case) censoring can provide a lower bit disagreement probability than 1-bit MAQ. However, at high γ, many bits are censored. For example, for $\gamma/\sigma = 0.2$ and $\rho = 0.96$, the censoring scheme achieves conditional probability of bit disagreement of 0.01 compared to 0.03 for the 1-bit MAQ scheme, but the censoring scheme must censor 24.7% of components.

For a moment, ignoring the loss of bits caused by the censoring scheme, we consider when to use the censoring scheme instead of the MAQ scheme. When the correlation coefficient ρ is low, the threshold can be set high, and the conditional probability of bit disagreement can be made lower in the censoring scheme. For example, if one wishes to design for highest possible secret bit rate with $P_{BD} \leq 0.04$, one would choose $m = 1$ for $0.95 < \rho < 0.98$, $m = 2$ for $0.98 < \rho < 0.993$, and $m = 3$ for $0.993 < \rho < 0.998$. For the $m = 1$ case, the censoring scheme could be used in order to lower the probability of bit disagreement at the expense of higher probability of censoring. In this example, if we have components i with $\rho_i > 0.98$, the MAQ scheme offers more bits per measurement component.

7 Experimental Implementation

In this section, we present the collection of a set of testbed RSS measurements on a bi-directional link and use the data to provide an example of the implementation and performance of the HRUBE method.

7.1 Setup

We use Crossbow TelosB wireless sensors which are connected via a USB connection to a laptop in order to record the collected data for post-processing and analysis. The TelosB mote is a low power wireless sensor module equipped with an IEEE 812.15.4-compliant RF transceiver (the TI CC2420), built-in antenna and a micro-controller. In general, wireless sensors are designed for low data rate and low computation and memory capabilities, and the TelosB can send 250 kbps. We choose the hardware to show the ability of simple hardware devices to collect channel data useful for secret key exchange.

We program the TelosB 802.15.4 radios to transmit and receive packets which are used as channel probes. In general, any data could be sent in the probe packets, including application data. In our implementation, the packets include only minimal data: the packet header, a node id, and a sequence number. When one radio receives a packet, it measures and records the RSS and stores it with the packet sequence number. Then, it increments the sequence number and replies with a packet with the incremented sequence number. This process repeats until the two nodes have collected enough probe data. Each packet contains a total of 10 bytes and thus has a duration of 40 μs. Processing time dominates the packet duration, and the channel is measured at a rate of approximately 100 probe packets per second, or 50 probes per node.

In any implementation, the RSS value measured by a receiver is device-specific. On the TelosB, the RSS is a signed 8-bit integer value, a value proportional to the measured average received power over the duration of the packet, in decibel milliwatts (dBm). While this integer value could be converted to a dBm value, it is unnecessary for our purposes, since all testbed radios are TelosB devices and thus have the same RSSI characteristic.

Our experimental tests involve two nodes, a and b, about 0.5 meters apart, measuring link (a, b). The original channel measurement vectors $\mathbf{w}_a$ and $\mathbf{w}_b$ are the lists of measured RSS values at nodes a and b, respectively. Figure 7 shows an example of $\mathbf{w}_a$ and $\mathbf{w}_b$ measured over the course of 100 channel probes. It shows some differences between the two directional measurements, but the large-scale changes of the two signals over time are remarkably similar. The tests also include the measurement at an eavesdropper node e which receives but never transmits. Node e is located about 1.5 m from node b and can measure the RSS on the channels (a, e) and (b, e). Figure 7 also shows the measurement of RSS on link (a, e). The signal at the eavesdropper is qualitatively very different from the true link measurements.

Motion of one of the nodes is used to generate fading changes in the RSS measurements, which is critical to observing fading during short-term experiments in indoor environments. During the experiment, the experimenter continuously moves one of the nodes in a 'random' manner, above, below, and around its starting position. Three experiments are conducted, each for a duration of about 1000 seconds. In this paper, we chose to measure vectors with length $N = 50$. Since measurements are taken approximately 50 per second, each vector takes 1 second to record. The long duration of the experiment ensures that we have enough data to characterize mean and covariance statistics.

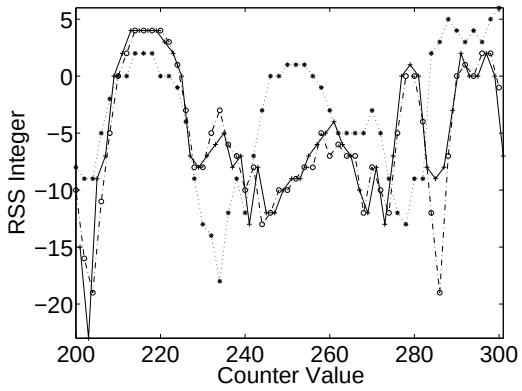

Fig. 7. Raw measured RSS values from a to b (+), from b to a (o) and from b to e (*).

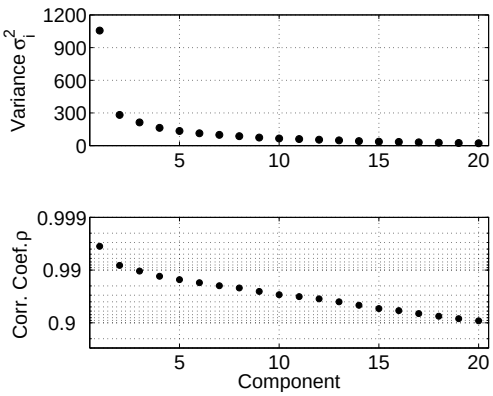

Fig. 8. Variances σ_i^2 and correlation coefficients ρ_i of first 20 components of $\mathbf{y}$.

7.2 Interpolation

Our measurement protocol lacks explicit time-synchronization, but the protocol approximately achieves a fractional time offset of $1/2$ at all times. The protocol is synchronized in the sense that each node transmits a packet as soon as it finishes receiving a probe packet from the other node. The delay between reception and transmission is nearly the same at each node, because they have the same hardware and run the same software. Packets can be delayed or dropped due to interference (nodes operate in the same band as WiFi), but we ignore these effects. We assume that the fractional sampling offset, $\frac{\tau_b(1)-\tau_a(1)}{T_R}$ from Section 4, is approximately equal to $1/2$. That is, node b's probes are sent halfway in between the previous and subsequent node a probes. This leads to a $\mu = 0.25$ in (2). Our measured values $\mathbf{w}_a$ and $\mathbf{w}_b$ are run through the interpolation procedure described in Section 4 and the synchronized data $\mathbf{x}_a$ and $\mathbf{x}_b$ are then formed using (4).

7.3 Data Model

We use the measured data from the first experiment (of three) to estimate the mean vectors $\mu_{\mathbf{x}_a}$ and $\mu_{\mathbf{x}_b}$, and covariance matrix $R_{\mathbf{x}}$ of the data vectors, which is estimated as:

$$\begin{aligned}\mu_{\mathbf{x}_a} &= \frac{1}{C}\sum_{i=1}^{C}\mathbf{x}_a^{(i)}, \quad \mu_{\mathbf{x}_b} = \frac{1}{C}\sum_{i=1}^{C}\mathbf{x}_b^{(i)} \\ \hat{R}_{\mathbf{x}} &= \frac{1}{2C-1}\left[\sum_{i=1}^{C}(\mathbf{x}_a^{(i)}-\mu_a)(\mathbf{x}_a^{(i)}-\mu_a)^T \right. \\ &\quad \left. + \sum_{i=1}^{C}(\mathbf{x}_b^{(i)}-\mu_b)(\mathbf{x}_b^{(i)}-\mu_b)^T\right] \qquad (20)\end{aligned}$$

where $\mathbf{x}_c^{(i)}$ is the ith 50-length measured RSS vector at node c, and $W = 49951$ is the total number of RSS vectors which can be formed from the measured data at each node.

Next, we also calculate the covariance between a measurement and the same measurement on the reverse link:

$$\begin{aligned}\hat{R}_{\mathbf{x}_c,\mathbf{x}_{\bar{c}}} &= \frac{1}{2C-1}\left[\sum_{i=1}^{C}(\mathbf{x}_a^{(i)}-\mu_a)(\mathbf{x}_b^{(i)}-\mu_b)^T \right. \\ &\quad \left. + \sum_{i=1}^{C}(\mathbf{x}_b^{(i)}-\mu_b)(\mathbf{x}_a^{(i)}-\mu_a)^T\right]. \qquad (21)\end{aligned}$$

7.4 De-Correlation Transform

From $\hat{R}_{\mathbf{x}}$, we calculate the SVD as given in (5). We plot the eigenvalues of each eigenvector (the variance of each component) in the top subplot of Figure 8. Then, using the KLT matrix U, we calculate the cross-directional covariance matrix from (7) and use it to calculate the correlation coefficients ρ_i as given in (8). These correlation coefficients are plotted in the bottom subplot of Figure 8). The highest correlation coefficient is 0.9965; there are seven components with $\rho > 0.98$ and fourteen components with $\rho > 0.95$.

Figure 9 shows the first nine eigenvectors of $\hat{R}_{\mathbf{x}}$, the columns of U, both in the time domain and in the frequency domain. The results show that the eigenvectors are nearly pure sinusoids. For short periods of time (our measurement vectors are recorded within one second), we can consider the fading signal to be a wide-sense stationary (WSS) random process, and as such, its eigenvectors should be complex exponentials. For longer periods of time, motion may not be WSS because of changes in the nature of the motion in the channel or of the nodes themselves. Future work should address the stationarity of the fading process during secret generation. Future implementations may also wish to use an FFT of the measured RSS values rather than the KLT in order to reduce computational and memory complexity.

Next, we apply the data model to experimental data sets two and three. We transform the measured data vectors $\mathbf{x}_c$ from these latter two data sets using the KLT to compute the values of $\mathbf{y}_c = U^T(\mathbf{x} - \mu_c)$, for $c \in \{a, b\}$.

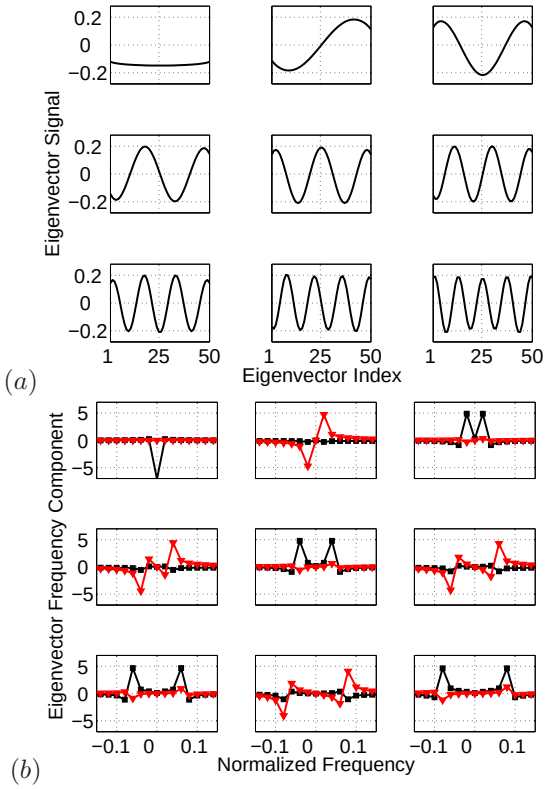

Fig. 9. First nine eigenvectors in the (a) temporal and (b) frequency domains. Frequency domain shows real (■) and imaginary (▼) components.

System Number	Total Bits	Bit Disagreement: Design	Actual Rate
1	22	0.040	0.0220
2	10	0.010	0.0054
3	3	0.001	0.0004

TABLE 3
Three System Designs: Analytical Gaussian P_{BD} and Actual Bit Disagreement Rates

Since we have recorded many seconds of RSS values in the two experiments, we have several realizations of $\mathbf{y}_a$ and $\mathbf{y}_b$ at nodes a and b, respectively. For example, Figure 10 shows the values of the second component at nodes a and b for direct comparison. It is clear that the two directional measurements are very similar to each other in comparison with the changes from realization to realization.

7.5 MAQ Implementation

Next, we perform multi-bit adaptive quantization on the measurements $\mathbf{y}_a$ and $\mathbf{y}_b$. In this implementation, node a is used as the leader, and node b is the follower. We design three systems for different desired bit disagreement rates. For example, in the first system, we design for a $P_{BD} = 0.04$. We are limited in the number of bits per component by the component's correlation coefficient. The correlation coefficients of the first dozen

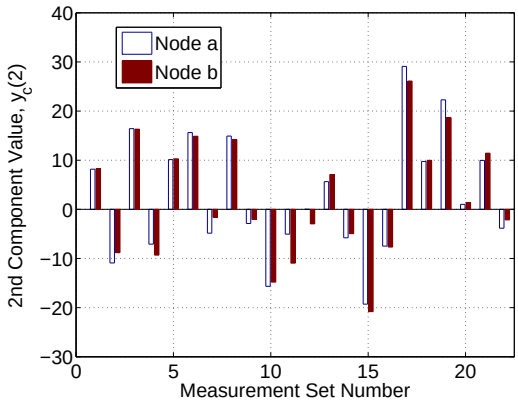

Fig. 10. Series of $y_a(2)$ and $y_b(2)$ for 23 subsequent measured vectors which show the level of agreement between bi-directional measurements.

i	ρ_i	m_i for System #1	#2	#3	i	ρ_i	m_i for System #1	#2	#3
1	0.9965	3	2	1	**9**	0.9746	1	0	0
2	0.9919	2	2	1	**10**	0.9706	1	0	0
3	0.9896	2	1	1	**11**	0.9681	1	0	0
4	0.9869	2	1	0	**12**	0.9649	1	0	0
5	0.9848	2	1	0	**13**	0.9600	1	0	0
6	0.9827	2	1	0	**14**	0.9535	1	0	0
7	0.9802	2	1	0	**15**	0.9464	0	0	0
8	0.9781	1	1	0	**16**	0.9413	0	0	0

TABLE 4
Three System Designs: Bits by Component

components are listed in the first column of Table 4. Using Figure 3, for each ρ_i, we look up to find the maximum number of bits m_i which achieves $P_{BD} \leq 0.04$ (or other system design specification). These are the values of m_i listed in a column of Table 4. We attempt three different system designs: $P_{BD} = 0.040$, 0.010, and 0.001, for system designs 1, 2, and 3, respectively. These specifications are listed in Table 3, along with the total number of secret bits generated from all components for each system design. Since m_i must be an integer and is chosen conservatively ($P_{BD} \leq 0.04$ for all i for system design 1), we would expect a lower actual bit disagreement probability than the design.

Note that a MAQ implementation needs to assume the marginal distributions in order to calculate the thresholds η_i. We assume, again, that $y_a(i)$ and $y_b(i)$ are zero mean and jointly Gaussian. However, rather than exchange measured marginal variances via radio communication, which might give away to an eavesdropper some information about the spread of measured values, we calculate locally the variance σ_i^2 at each node from its own collection of measured $\mathbf{y}$ values. For example, at node a, we use the collected realizations of $y_a(i)$ through the course of the experiment to compute the variance of $y_a(i)$ for use at node a. Future implementation work will need to study the tradeoff between latency and other methods for estimating the variances of each component.

7.6 Experimental Performance

Finally, we run each system through the data collected experimentally. The experimental bit disagreement rate is computed by counting the number of bits for which nodes a and b disagree and dividing by the total number of secret bits generated in the course of the experiments. Table 3 shows the results. In each system design, the actual bit disagreement rate is lower than the P_{BD} for which the system was designed. Since m_i is chosen conservatively, as described in Section 7.5, we expect the design specification to be an upper bound on P_{BD}.

The system designs also show that a large number of bits can extracted from the channel measurements. At the highest P_{BD}, we can extract 22 bits per second. This compares well to [16] which describes an implementation which achieves approximately one secret bit per second.

Note that the secret key generation rates from the HRUBE method, including the 22 bits/sec system, are not reconciled; at a bit disagreement rate of 2.2%, on average, one out of the 22 bits will disagree about 40% of the time. Other research has used information reconciliation procedures to reveal small amounts of information between two nodes that permitted correction of limited numbers of bit disagreements, as discussed in Section 2. We assume that such reconciliation will be implemented as part of a secret key generation system which reliably finds keys which agree at the two ends of a link.

7.7 Statistical Tests of Correlation

We have designed the HRUBE method so that, any pair of bits within secret key vector $\mathbf{z}$ has zero correlation. In this section, we use our large set of measured data to estimate the correlation coefficient between bits and test for non-zero correlation. We estimate two types of correlation coefficients:

1) *Pair-wise bit correlation coefficients*. We denote ρ_{z_i,z_j} as the correlation coefficient between the ith and jth component of vector $\mathbf{z}$, for $i \neq j$.
2) *Global bit correlation coefficient*. We denote $\rho_{\mathbf{z}}$ as the correlation coefficient between all pairs of different components of $\mathbf{z}$. A single correlation coefficient is only valid if the correlation between any pair of bits in $\mathbf{z}$ is assumed identical.

From the measured data, we generate $n = 833$ realizations of the vector $\mathbf{z}$ from our data, from which estimated pair-wise bit correlation coefficients $\{\hat{\rho}_{z_i,z_j}\}_{i,j}$ range from -0.10 to +0.12. The estimated global bit correlation coefficient $\rho_{\mathbf{z}}$ is -0.0036.

Clearly, the *estimated* correlation coefficients will never precisely zero, even if $\rho = 0$ exactly. So, for the given number of realizations, we provide hypothesis tests to quantify if these non-zero correlation coefficient estimates are likely, or unlikely, to have been generated if the true $\rho = 0$. Formally, the decision is:

$$\begin{aligned} H_0: &\quad \rho = 0 \\ H_1: &\quad \rho \neq 0 \end{aligned} \tag{22}$$

The hypothesis test is performed on statistic t [26],

$$t = \hat{\rho}\sqrt{\frac{1-\hat{\rho}2}{n-2}} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$$

where $\hat{\rho}$ is the correlation coefficient estimated from the data and n is the number of realizations used in the estimate, and γ is a threshold, set based on the desired probability of false alarm. The statistic t has the student-t distribution with $n-2$ degrees of freedom. Furthermore, in the limit for high n, the distribution of t approaches the zero-mean unit-variance Gaussian distribution. In practice, $n > 100$ is high enough for this approximation to hold.

For the pair-wise bit correlation coefficients, the estimated t-statistics for each pair are group-tested; that is, the threshold γ is set so that the total probability of false alarm is 5%. For the given parameters, $\gamma = 3.70$. For the three systems in Table 3, all of the correlation coefficients ρ_{z_i,z_j} have $t < gamma$. The global bit correlation coefficient, $\hat{\rho}_{\mathbf{z}}$, has $t = 1.74$, below the threshold $\gamma = 1.96$ chosen for false alarm rate 5%.

We also estimate the entropy of the bit sequence output from the HRUBE system, using the NIST random generator test suite [27]. For systems 1, 2, and 3 (shown in Table 3), the experimental entropies are estimated to be 0.959, 0.981, and 0.981, respectively, per generated bit.

7.7.1 Discussion

Both tests indicate that the correlation coefficient between bits in $\mathbf{z}$ is not statistically significantly different from zero. The best estimate of $\rho_{\mathbf{z}}$ is -0.0036. We have similarly run tests on other sets of collected data; some data sets pass the correlation test and decide H_0, those that decide H_1 cross the threshold γ only by a small margin. It is fair to say that there may exist small correlations, perhaps $|\rho| < 0.01$, but when given a high number of realizations n, a test may detect these small correlations.

Non-zero correlations can be caused by imperfect knowledge of the covariance matrix of the interpolated RSS sample vector $\mathbf{x}$. Small changes in the covariance structure over time may occur, and when this occurs, the KLT based on a static transformation matrix U in (6) does not produce a completely uncorrelated sample vector output. Future work should develop methods to adaptively update the KLT from recently measured data to further reduce correlations in bit outputs.

Note that subsequent bit vectors $\mathbf{z}$ are not designed for zero correlation with each other. The HRUBE decorrelates a set of N RSS samples and then produces a bit vector $\mathbf{z}$. A later set of N RSS samples produces another bit vector $\mathbf{z}'$ which may be correlated with the first vector $\mathbf{z}$. A system designer should either (1) set N high enough such that enough bits can be obtained from one bit vector $\mathbf{z}$, or (2) allow time to pass between the first and second set of RSS samples such that the two sets of samples

are uncorrelated. This is a tradeoff between latency, computational complexity, and correlation, which must be studied prior to system deployment.

8 Conclusion

This paper provides a general framework, which we call HRUBE, for the extraction of secret uncorrelated bit vectors from a series of radio channel measurements. The framework includes interpolation, de-correlation transformation of the data, and then an adaptive quantization scheme to allow each component to be quantized to an arbitrary number of bits. Analysis has been developed to calculate the probability of bit disagreement using the HRUBE scheme, which is used to design systems with a given bound on probability of bit disagreement. Numerical results are reported for the case of bi-variate Gaussian measurements. We show, via an implementation of HRUBE, some examples of the secret bit rate vs. P_{BD} tradeoff, including the possibility of achieving 22 secret bits per second at a bit disagreement rate of 2.2%, or achieving 10 bits per second at a bit disagreement rate of 0.54%. Experimentally, pairs of bits within one bit vector show a correlation coefficient of -0.0036, close to perfect de-correlation.

Future work must apply the HRUBE method to other modes of radio channel measurements and improve the distributional assumptions. In particular, the joint distribution of $y_a(i)$ and $y_b(i)$ should be thoroughly investigated using large sets of measured data. Many of the system design choices (distribution parameters, number of bits per component) might be trained in real-time from initial measurements, and research should investigate such methods. For example, quantization levels $\{\eta_k\}$ could be determined by each node on its own from measured vectors $\mathbf{y}$; or nodes could periodically (perhaps once per secret) communicate to set them. Research should also investigate adaptive update of the KLT in order to achieve even lower correlation between bits. Of course, additional research should be performed from an attacker's perspective: to attempt to 'break' secret keys generated from extraction methods which are seen as weak; or to deny the ability of two radios to agree on a secret key. Research must design key generation systems to be as robust as possible to such attacks. The general problem of secret bit extraction, *i.e.*, translating radio channel measurements into shared secret bits, is generally an open statistical signal processing problem, and one that may have direct impact for the improvement of wireless network security.

References

[1] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," *J. Cryptol.*, vol. 5, no. 1, pp. 3–28, 1992.

[2] S. Wiesner, "Conjugate coding," *SIGACT News*, vol. 15, no. 1, pp. 78–88, 1983.

[3] L. Greenemeier, "Election fix? Switzerland tests quantum cryptography," *Scientific American*, October 2007.

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

[5] J. E. Hershey, A. A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *IEEE Trans. Commun.*, vol. 43, no. 1, pp. 3–6, Jan. 1995.

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

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

[8] 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 *6th Annual Conference on Communication Networks and Services Research (CNSR2008)*, May 2008.

[9] C. Ye, A. Reznik, G. Sternberg, and Y. Shah, "On the secrecy capabilities of ITU channels," in *IEEE Vehicular Technology Conf. (VTC'07-Fall)*, Oct. 2007, pp. 2030–2034.

[10] R. Wilson, D. Tse, and R. A. Scholtz, "Channel identification: Secret sharing using reciprocity in UWB channels," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 364–375, Sept. 2007.

[11] 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*, Nov. 2007, pp. 401–410.

[12] 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)*, Sept. 2006, pp. 33–42.

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

[14] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3776–3784, Nov. 2005.

[15] M. A. Tope and J. C. McEachen, "Unconditionally secure communications over fading channels," in *Military Communications Conference (MILCOM 2001)*, vol. 1, Oct. 2001, pp. 54–58.

[16] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *ACM Intl. Conf. on Mobile Computing and Networking (Mobicom 2008)*, Sept. 2008, (to appear).

[17] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized access points using clock skews," in *ACM Intl. Conf. on Mobile Computing Networking (Mobicom'08)*, Sept. 2008.

[18] U. M. Maurer and S. Wolf, "Unconditionally secure key agreement and the intrinsic conditional information," *IEEE Trans. Info. Theory*, vol. 45, no. 2, pp. 499–514, 1999.

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

[20] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "PARADIS: Physical 802.11 device identification with radiometric signatures," in *ACM Mobicom*, Burlingame, CA, Sep. 2008.

[21] C. Farrow, "A continuously variable digital delay element," in *IEEE Intl. Symposium on Circuits and Systems*, vol. 3, June 1998, pp. 2641–2645.

[22] M. Rice, *Digital Communications: a Discrete-Time Approach*. Pearson Prentice Hall, 2009.

[23] A. Nasir and K. R. Rao, *Orthogonal Transformations for Digital Signal Processing*. Springer Verlag, 1975.

[24] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *ACM SIGCOMM*, Aug. 2004.

[25] A. Luminia, D. Maioa, and D. Maltoni, "Continuous versus exclusive classification for fingerprint retrieval," *Elsevier Pattern Recognition Letters*, vol. 18, no. 10, pp. 1027–1034, Oct. 1997.

[26] W. W. Hines and et.al., *Probability and Statistics in Engineering*, 4th ed., 2003.

[27] NIST. A statistical test suite for random and pseudorandom number generators for cryptographic applications. http://csrc.nist.gov/publications/nistpubs/800-22/sp-800-22-051501.pdf.