

Secret Key Generation from Multipath Fading Channels

Chunxuan Ye, Alex Reznik, Gregory Sternberg and Yogendra Shah
InterDigital Communications Corporation
King of Prussia, PA, 19406, U.S.A.

E-mails: {Chunxuan.Ye, Alex.Reznik, Gregory.Sternberg, Yogendra.Shah}@InterDigital.com

Abstract—We describe a technique to generate a shared perfectly secret key between two wireless terminals based on their correlated observations of a multipath fading channel. In particular, we present a method of decomposing channel observations as a summation of discrete pulses that converts highly correlated random variables to a sequence of independent random variables. This enables the use of existing techniques for the subsequent secret key construction. Simulation results on our secret key generation system are provided.

I. INTRODUCTION AND PROBLEM SETUP

Consider two terminals, Alice and Bob, that communicate with each other on the same frequency in a wireless environment. These two terminals can apply training sequences in their transmissions to enable the receivers to estimate the Channel Impulse Response (CIR) of their reciprocal wireless channel. The estimation is performed assuming a single-input-single-output omni-directional antenna.

It is known (e.g., cf. [1]) that a wireless channel is well modeled by a collection of discrete pulses with different amplitude and delay. Moreover, except in the ultra-wideband scenario, each pulse is usually subject to Rayleigh or Rician fading. This holds true in all SISO situations (whether or not the paths are “resolvable”) and often holds true in MIMO, where the Gaussian r.v.’s now become vector-based themselves. In this paper, we concentrate on secrecy generation from non-ultra-wideband SISO channels. Parallel work concentrating on the ultra-wideband channels can be found in [2].

Mathematically, we have

$$a(t) = \sum_{l=1}^L \alpha_l \delta(t - \tau_l), \quad (1)$$

where $L \in [1, +\infty)$ and α_l, τ_l represent, respectively, the amplitude and delay of the l^{th} path in this wireless L -path fading channel. We shall focus on the Rayleigh fading case only, since Rician fading can be dealt with in the same manner by subtracting the known mean. In the Rayleigh fading case, the amplitudes $\alpha_1, \dots, \alpha_L$ are zero-mean complex Gaussian random variables.

The CIR of a wireless channel is

$$h(t) = p(t) * a(t) = \sum_{l=1}^L \alpha_l p(t - \tau_l), \quad (2)$$

where $p(t)$ is the “pulse shape” resulting from the pre-determined band-limited transmitter and receiver filters. Equation (2) implies that the CIR is the superimposition of multiple delayed and scaled copies of the pulse shape $p(t)$.

Alice and Bob respectively observe a sampled noisy version of the CIR $h(t)$. Their observations can be written as

$$h_A[n] = h(nT_S - \tau_A) + Z_A[nT_S], \quad (3)$$

$$h_B[n] = h(nT_S - \tau_B) + Z_B[nT_S], \quad (4)$$

where

- T_S is the sample interval, which is assumed to be the same at both terminals.
- τ_A and τ_B are the sampling time offsets associated with each receiver. Hence, the sampling time difference between two terminals is $|\tau_A - \tau_B|$.
- $Z_A[nT_S]$ and $Z_B[nT_S]$ are independent additive white Gaussian noise sequences.

Since Alice’s and Bob’s observations $h_A[n], h_B[n]$ are based on their reciprocal wireless channel $a(t)$, they are correlated with each other. On the other hand, a third party’s observations will almost certainly remain uncorrelated with the channel-specific observations of Alice and Bob if the third party is located more than a wavelength away from Alice and Bob.

Based on their correlated channel observations, Alice and Bob wish to generate a common secret key. To do so, they communicate over the same wireless channel. In this paper we consider only passive attackers. Moreover, the use of standard error-correction techniques makes the error probability negligible. Thus, we may assume that the communication required for key generation occurs over a side channel that is public but error-free and authenticated. The generated secret key should be concealed from a potential passive eavesdropper, who will be able to observe the transmissions on the public channel but not to tamper with them. In particular, the generated secret key must be nearly “statistically independent” of the public transmissions. Hence, such a secret key is called a perfectly secret key or an information theoretic secret key [3, 4]. The problem of generating a perfectly secret key from correlated observations of a single path Rayleigh fading channel (i.e., $L = 1$) has been addressed in [5]. In this paper, we shall extend the solutions in [5] to a multipath Rayleigh fading channel (i.e.,

$L \geq 2$). Specifically, we shall concentrate on the 3GPP standard, reference multipath fading channels proposed in [6].

II. BACKGROUND

A. Upper Bound on Secret Key Rates

It is known [3, 4] that the mutual information between Alice and Bob's CIR observations $h_A[n]$, $h_B[n]$ is the upper bound on the achievable secret key rate. We start by quantifying the mutual information for our model.

The algorithms presented here hold for any values of noise powers at Alice and Bob. However, in order to present the results as a function of a single variable, we assume hereafter that the additive Gaussian noise power at both terminals is N . Additionally, we let the wireless channel between Alice and Bob be an L -path fading channel with average path powers (p_1, \dots, p_L) . Then the mutual information between Alice and Bob's CIR observations on the l^{th} path is given by [5]

$$I_l = \log \left(1 + \frac{p_l / N}{2 + N / p_l} \right). \quad (5)$$

By the union bound, the mutual information between Alice and Bob's overall CIR observations is upper bounded

$$\text{by } \sum_{l=1}^L I_l.$$

When the first path in an L -path fading channel is set as the reference path, the relative average path power of this channel can be written as $(\bar{p}_1, \dots, \bar{p}_L)$, with $\bar{p}_l = p_l / p_1$. Then, the secret key rate is upper bounded by

$$\sum_{l=1}^L \log \left(1 + \frac{\text{SNR} \cdot \bar{p}_l}{2 + 1/(\text{SNR} \cdot \bar{p}_l)} \right), \quad (6)$$

where $\text{SNR} = p_1 / N$ is defined for the reference path. The convention of letting the first path have a nominal 0 dB power is consistent with the channel definitions used in the 3GPP specification [6], and we shall follow it here. For the reader's convenience, we list some 3GPP specified channels in Table I.

Note that the upper bound given by (6) is not generally tight, due to inter-path interference. However, we expect this upper bound to be tight for most of the 3GPP standard, reference multipath fading channels, since almost all paths in these channels are separable paths (see Table II). Hence, we shall still use this upper bound in the rest of this paper. More generally, it may be possible to estimate the mutual information of wireless channels using non-parametric estimation techniques, such as, e.g., [7, 8].

B. Secret Key Generation System

We now give an overview of our secret key generation system. The secret key generation sub-systems for Alice and Bob are shown in Figures 1 and 2, respectively. As the figures show, both transceivers estimate the CIR of the reciprocal wireless channel based, e.g., on their received training signals. The estimation can be performed through a number of

standard methods. The output of the channel estimation is the sampled CIR $h_A[n]$ or $h_B[n]$.

Table I: PROPAGATION CONDITIONS FOR MULTIPATH FADING ENVIRONMENTS

Channel model	WG4 Case 1	WG4 Case 2	WG4 Case 3	ITU PA3	ITU PB3	ITU VA30
Relative delay (ns)	0 976	0 976 20000	0 260 521 781	0 110 190 410	0 200 800 1200 2300 3700	0 310 710 1090 1730 2510
Relative average power (dB)	0 -10.0	0 0	0 -3.0 -6.0 -9.0	0 -9.7 -19.2 -22.8	0 -0.9 -4.9 -8.0 -7.8 -23.9	0 -1.0 -9.0 -10.0 -15.0 -20.0

The sampled CIR should be processed in order to eliminate both the redundancy in the samples and the discrepancies caused by the sampling time difference at both terminals. This is done by the CIR post-process block. In this paper, we focus on the implementations of the CIR post-process block.

Alice's post-processed CIR samples are quantized to a bit string, which is then encoded by a block error code. The resulting parity bits (or, more generally, the syndrome) are transmitted to Bob through an error-free channel. Bob tries to recover Alice's bit string by using the received parity bits and his own post-processed CIR samples. Both terminals then extract a perfectly secret key from this bit string, by means of hashing in the Privacy Amplification (PA) process block. We shall rely on the results in [4, 5] for the latter 2 aspects of secret key generation.

C. Challenges for Multipath Fading Channels

Note that the outputs of the channel estimation blocks $h_A[n]$ (or $h_B[n]$) usually consist of highly correlated samples. In order to apply the techniques developed in [5], it is necessary to remove the correlation among the samples while still keeping the cross-correlation among Alice's and Bob's post-processed samples.

For a single-path Rayleigh fading channel, a single well-selected sample is sufficient. However, for a multipath fading channel, we cannot simply select several samples (e.g. one sample per path) and hope that they will be independent. Hence, removal of the

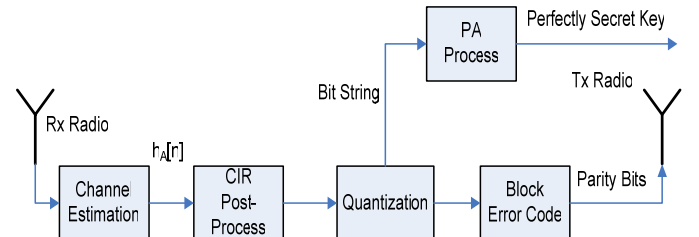


Figure 1: Block diagram of secrecy processing for Alice

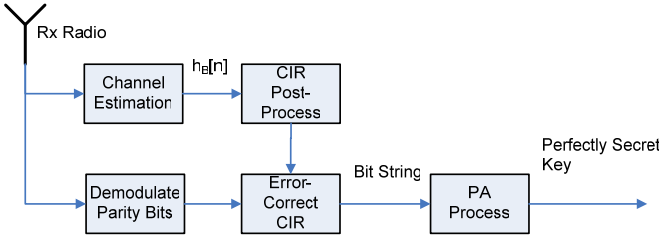


Figure 2: Block diagram of secrecy processing for Bob

correlation among samples for multipath fading channels is the main challenge here.

The sampling time difference at the terminals is another source of practical challenges. Sampling the same CIR with different sampling time offsets may lead to data that, while highly correlated, will in fact be quite distinct. While increasing the sampling rate mitigates this problem, it also leads to highly redundant samples with a complicated structure that is difficult to remove. Hence an alternative approach to the problem is required.

Our approach is to look for the discrete constituent multipath components that produce the observed CIR. This is done independently by Alice and Bob. The discrete path amplitudes derived at these two terminals should exhibit the following properties:

- *Consistency*: Alice and Bob should arrive at the same path locations/amplitudes.
- *Independence*: The resulting random variables should be independent (or nearly so).

Note, however, that we do not require *correctness*: i.e., as long as Alice and Bob identify the same candidate path locations leading to independent random variables, we do not require that these be the actual path locations that created the observed CIRs. In a Rayleigh (i.e., complex Gaussian) setting, the answer that suggests itself is to look for the basis vector of some appropriately defined vector space spanned by the observed CIRs. In one particular interpretation, this could be understood as looking for the sparsest possible representation of the observed CIRs in a dictionary of possible pulse shapes. Such an interpretation relates the problem to the now well-studied field of sparse representations [9-14] and suggests the use of the Orthogonal Greedy Algorithm (OGA), which we review next.

III. ALGORITHM DESCRIPTION

Our main results are presented in this section. We begin with a short overview of the OGA as applied to the sparse representation problem and develop a basic OGA algorithm based on this. While a direct application of this basic OGA algorithm in the CIR post-process block offers reasonable performance for multipath channels, we demonstrate that a double pass of the basic OGA algorithm yields further improvements.

A. Sparse Representation: OGA Overview

Consider a full column rank matrix $\mathbf{D}_{K \times N}$, $K < N$, which is called a dictionary. If $\mathbf{s} = \mathbf{D}\mathbf{a}$, then \mathbf{a} is called a coefficient vector for the representation of \mathbf{s} in \mathbf{D} . As $K < N$, a given \mathbf{s} corresponds to an infinite number of coefficient vector

representations in \mathbf{D} . We are interested in finding the sparsest coefficient vector, i.e., the one with the maximum number of zeros, among all possible coefficient vector representations of \mathbf{s} . It is shown in [9] that the sparsest coefficient vector representation of \mathbf{s} ($=\mathbf{D}\mathbf{a}$) is *unique* if the dictionary \mathbf{D} satisfies

$$\|\mathbf{a}\|_0 < Z(\mathbf{D})/2, \quad (7)$$

where $\|\mathbf{a}\|_0$ stands for the l^0 -norm of \mathbf{a} (i.e., the number of non-zero elements in \mathbf{a}), and $Z(\mathbf{D})$ is the *spark* of the dictionary \mathbf{D} , defined as

$$Z(\mathbf{D}) = \min_{\mathbf{x} \in \{\mathbf{x} \neq \mathbf{0} : \mathbf{D}\mathbf{x} = \mathbf{0}\}} \|\mathbf{x}\|_0. \quad (8)$$

For a dictionary \mathbf{D} satisfying (7), finding the unique sparsest coefficient vector involves an exhaustive search among all possible coefficient vectors and is therefore infeasible in most situations. However, it has recently been shown in [9-14] that the process of finding the unique sparsest coefficient vector can be simplified by using linear programming or the OGA when the dictionary \mathbf{D} satisfies additional conditions (cf. [14]).

The basic principle of OGA is that, given a vector \mathbf{s} and a dictionary of basis vectors \mathbf{D} , it selects the single-element vector that provides the best possible fit according to some criterion. The contribution of this vector is then computed and subtracted from the source vector \mathbf{s} . The process is repeated until some stopping threshold is reached. The output of OGA is hence the sum of those selected single-element vectors.

B. Basic OGA

In this sub-section we introduce a basic OGA algorithm, which will be used in the CIR post-process block.

As mentioned, we desire the CIR post-process block to decompose a sampled CIR into a summation of discrete independent pulses. This can be done by OGA, by regarding the sampled CIR as a vector, the pulse shape with all possible delays as a dictionary, and the discrete independent pulses as a coefficient representation. A basic OGA algorithm is given below.

Basic OGA: Let $H(f)$ and $P(f)$ be the Fourier transforms of the sampled CIR $h[n]$ (can be either $h_A[n]$ or $h_B[n]$) and sampled pulse shape $p[n] = p(nT_s)$, respectively. Let THR be a pre-determined threshold. Set $H_1(f) = H(f)$, and $l = 1$.

Step 1: Find $m > 0$, $\phi \in [0, 2\pi)$ and $\tau \in \mathcal{R}$, which minimize

$$\|H_l(f) - me^{j\phi} P(f) e^{-j2\pi f\tau}\|_2. \quad (9)$$

Denote these by m_l, ϕ_l, τ_l , and let $\alpha_l = m_l e^{j\phi_l}$.

Step 2: Set $H_{l+1}(f) = H_l(f) - \alpha_l P(f) e^{-2\pi f\tau_l}$.

Step 3: If $m_l < THR$, then output $(\alpha_1, \tau_1), \dots, (\alpha_{l-1}, \tau_{l-1})$ and stop. Otherwise, let $l = l + 1$ and return to Step 1. ■

It can be derived from Step 1 that

$$(\phi_l, \tau_l) = \arg \max_{(\phi, \tau)} \operatorname{Re} \left\{ e^{j\phi} \sum_{n=-\infty}^{\infty} h_l[n] p_\tau^*[n] \right\} \quad (10)$$

and

$$m_l = \frac{\text{Re} \left\{ e^{j\phi_l} \sum_{n=-\infty}^{\infty} h_l[n] p_{\tau_l}^*[n] \right\}}{\|P(f)\|_2^2}, \quad (11)$$

where $p_\tau[n] = p(nT_s - \tau)$, and $h_l[n]$ is the inverse Fourier transform of $H_l(f)$. Equations (10) and (11) suggest that we first correlate $h_l[n]$ against all delayed-and-sampled versions of $p(t)$. The optimum τ_l is the delay for which the absolute value of the correlation is maximum; the optimum ϕ_l is minus the angle of the correlation at τ_l ; and the optimum m_l is the absolute value of the correlation at τ_l , divided by the square of the l^2 -norm of $P(f)$.

In practice, it is impossible to correlate against all the values of τ . Note that $p_{\tau_1}[n]$ and $p_{\tau_2}[n]$ are delayed versions of each other if $(\tau_1 - \tau_2)$ is an integer. If we discretize the time line such that the time grid spacing is $1/G$, for some integer G , we need to implement a finite bank of filters, each representing $p_\tau[n]$ for a different fractional delay $\tau \in [0,1)$. Hence, the dictionary in our implementation is actually the set of pulse shapes, each delayed by $1/G$.

Remarks:

1. Note that the above dictionary usually does not satisfy the conditions necessary for OGA to solve the sparsest problem as stated in [14]. In particular, the solution generated may not be the sparsest one. However, subject to a small error probability, it will be *consistent*, i.e., the solutions generated by Alice and Bob will be identical. Ensuring full consistency motivates a double use of the OGA processing.

2. The fitness criterion in (9) guarantees that the residual signal is uncorrelated with the selected signal. Recall that a sampled CIR is a jointly Gaussian random vector, since it is composed of Gaussian noise and Gaussian pulse amplitude. Because OGA conducts linear operations, the outputs of OGA are also Gaussian random variables. Thus, the resulting Gaussian random variables are independent -- i.e., for Rayleigh and Rician fading, *OGA guarantees independence of the output data*.

3. Finding a good threshold (*THR*) in Step 3 is not easy, as the proper threshold may vary with the signal itself and SNR. Instead of a constant threshold, we propose one that depends on the signal itself and the SNR. This has two important benefits: i) It is much more robust in the real-world scenario, as it depends significantly less on knowing actual channel conditions. ii) It guarantees that the OGA always outputs at least one value.

C. Applying OGA in the CIR Post-Process Block: Single Pass Scheme

A block diagram implementing the CIR post-process block is given in Figure 3.

We observe that each terminal independently processes its sampled CIR by the OGA block, which runs the basic OGA

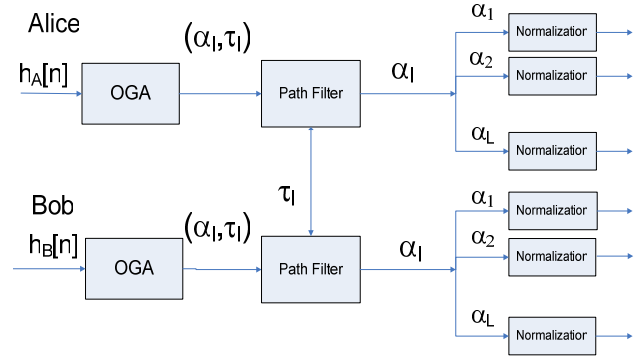


Figure 3: Implementing the CIR post-process block by the single pass scheme

algorithm. The outputs of the OGA block are a sequence of pairs of path delay τ_l and path amplitude α_l .

Each terminal then processes its detected paths in the path filter block. The main purpose of the path filter block is to discard some paths which are detected at the local terminal but not at the other terminal. To do this, both terminals need to exchange their detected path delays through the public error-free channel. If, for a path delay detected by Alice, there is a “corresponding” value on Bob’s side¹, then this pair of paths is assumed to be the same path. All “unpaired” paths are discarded.

Because we rely on [5] for randomness extraction, we only use the complex path amplitudes as the source of secrecy. Thus, the public transmissions of path delays will not leak any information on the subsequently generated secret key.

The outputs of the path filter block are a series of path amplitudes. These path amplitudes are independent Gaussian random variables with different variances. The variance of path amplitude depends on the average path power. For unified post-processing, it is desirable to normalize these random variables based on their estimated variances.

D. Applying OGA in the CIR Post-Process Block: Double Pass Scheme

A significant downside of the scheme proposed above is the probability of missing a path by at least one of the two terminals. To mitigate this, we propose a modified dual pass scheme, described below and illustrated in Figure 4.

The double pass scheme in Figure 4 applies OGA twice at each terminal. OGA1 is used as a path searcher, and OGA2 works as a generator of independent samples.

With the sampled CIR as the input signal, OGA1 performs the basic OGA operations. However, instead of pairs of path delay and path amplitude, only the path delay outputs of OGA1 are considered. Note that the path delays detected for each individual channel observation are not guaranteed to be identical, but they are expected to be around the fundamental underlying path delays. In most mobile systems, path locations change significantly slower than CIR estimates become available. Thus, each terminal is able to accumulate a

¹ Here, the effect of sampling time difference between two terminals should be considered. This sampling time difference can be estimated.

large number of path location vectors (OGA1 outputs). Under the assumption that these should be nearly identical, they can then be processed (by the path delay estimation block) to produce a highly reliable (and, as we shall see, consistent) estimate of path locations across the two terminals. The path delay estimation block works as follows. It partitions the time line into small segments and counts the number of path delay outputs of OGA1 in each segment. The outputs of this block are set as the middle of those time segments, which contain locally maximum numbers and are above a minimum threshold.

Simulations show that the path searcher composed of OGA1 and path delay estimation works reasonably well for most 3GPP standard, reference multipath channels. Table II shows the number of paths detected by the path searcher for these channels. As the table makes clear, the path searcher finds all the underlying paths for WG4 Case 1/2/3 channels and most of the underlying paths for ITU PB3 and ITU VA30 channels. The small relative path delays in the ITU PA3 channel cause the difficulty in distinguishing the paths.

Note that the above path searcher may be implemented in alternate ways, e.g., by utilizing the tap information gained from a rake receiver or equalizer.

The OGA2 block uses the refined path delays and stored sampled CIR as input to “finish” the basic OGA algorithm as follows:

Step 1: For a given path delay τ_l , determine its corresponding path amplitude $\alpha_l = m_l e^{j\phi_l}$ according to (10) and (11).

Step 2: Set $H_{l+1}(f) = H_l(f) - \alpha_l P(f) e^{-2\pi f \tau_l}$.

Step 3: Repeat for the entire given path delays (τ_1, \dots, τ_L) .

Then, output $(\alpha_1, \dots, \alpha_L)$.

The outputs of OGA2, i.e., the path amplitudes, are independent Gaussian random variables, which are normalized based on their estimated variances.

Another advantage of the double pass scheme is that the exchange of path delay information is not needed. The consistency of the path delay estimation block is sufficient to minimize the impact of path delay error events. Finally, we note again that the proposed scheme may require significant memory to store the CIRs for processing in OGA2. In slow fading channels, however, the CIRs change quite slowly, which means that only a few CIRs may need to be stored (the rest would be completely correlated to these).

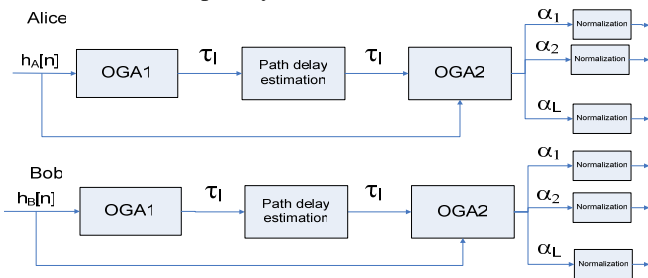


Figure 4: Implementing the CIR post-process block by the double pass scheme

Table II: THE NUMBER OF PATHS DETECTED BY THE PATH SEARCHER (TESTED FOR SNR [10, 35] dB)

Channel model	WG4 Case 1	WG4 Case 2	WG4 Case 3	ITU PA3	ITU PB3	ITU VA30
# detected paths	2	3	4	1(SNR<22dB) 2(SNR>22dB)	5	4(SNR<22dB) 5(SNR>22dB)
# underlying paths	2	3	4	4	6	6

E. Data Processing Scheme

As seen from Figures 3 and 4, at the end of CIR post-processing, each terminal can get several sequences of i.i.d. normalized Gaussian random variables, one sequence per estimated path. After concatenating these into one sequence, we are ready to use the existing techniques [4] to construct a secret key. This technique of concatenating sequences is referred to as a *mixed scheme*. Specifically, by the *mixed scheme*, all the normalized Gaussian random variables from different paths are quantized at the same level (i.e., the same number of bits per random variable), and the quantization level is determined by the SNR for the reference path.

Due to the difference in average path powers, each path in a multipath fading channel corresponds to an individual SNR. This SNR is likely to be different from the SNR for the reference path. Hence, the correlation between two terminals' corresponding Gaussian sequences from one path may be different from the correlation from another path. This implies that separate processing on the i.i.d. Gaussian sequence for each path may result in a higher secret key rate. Such a scheme of separately using the techniques of [5] on each detected path is referred to as *per path scheme*. By the *per path scheme*, the Gaussian random variables from a path are quantized at a level determined by the SNR for that path.

IV. SIMULATION RESULTS

In this section, we examine the simulated secret key rate resulting from different OGA application schemes (i.e., the single pass scheme or the double pass scheme) and different data processing schemes (i.e., the *mixed scheme* or the *per path scheme*), followed by the techniques developed in [5] for 3GPP standard, reference multipath channels.

The techniques developed in [5] are given a brief overview below. Alice uses scalar quantization and Gray coding to encode her Gaussian samples to a bit string. The syndrome of this bit string (in terms of a given LDPC code) is transmitted to Bob. Bob then tries to decode that bit string by applying a modified belief-propagation algorithm, in which the log-likelihood ratio is softly encoded. Finally, both terminals hash out the publicly revealed information (i.e., the syndrome) from that bit string, leaving purely secret bits.

The same LDPC code as in [5] is also used in the simulations, i.e., the irregular LDPC code with rate=1/2, block size=4800 bits, and degree distribution pair as

$$\lambda(x) = 0.071428 x + 0.230118 x^2 + 0.079596 x^9 + 0.147043 x^{10} + 0.073821 x^{48} + 0.397994 x^{49},$$

$$\rho(x) = x^{27}.$$

Thirty iterations of the belief-propagation algorithm are

allowed. A target secret key bit error rate of 10^{-4} is achieved in all the simulations.

Figures 5 and 6 respectively show the secret key rates for WG4 Case 1 and WG4 Case 2 channels achieved by using the single pass, *mixed schemes* and the double pass, *mixed schemes*. For comparison, we also plot the upper bound (6) in these figures. It is seen from Figure 6 that the secret key rates achieved by using the single pass scheme are much lower than those achieved by using the double pass scheme. This is due to the missing paths in the single pass scheme.

For the WG4 Case 1 channel, the double pass scheme outperforms the single pass scheme only at high SNR. This is because the WG4 Case 1 channel has 2 paths, and the second path has very low average path power. At low SNR, the data collected from the second path make few contributions to the

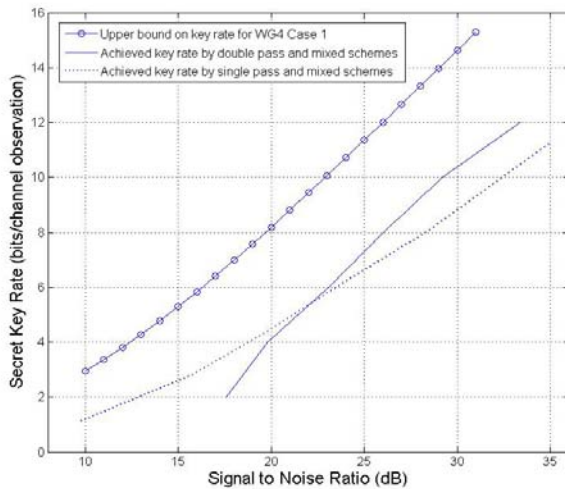


Figure 5: Secret key rates for WG4 Case 1 channel

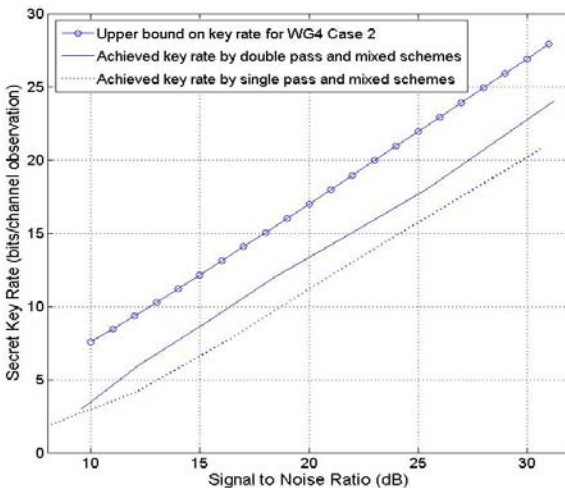


Figure 6: Secret key rates for WG4 Case 2 channel resulting secret key. Furthermore, mixing the data from the second path with the data from the first path significantly decorrelates the data between two terminals, causing decoding failures with the given LDPC code. Hence, at low SNR the single pass scheme, which is mainly based on the data from the first path, performs better than the double pass scheme,

which is based on the data from both paths. In other words, the detection of the second path actually worsens the situation. This problem can be solved by the double pass, per path schemes.

Figures 7-12 show the respective secret key rates for WG4 Case 1, WG4 Case 2, WG4 Case3, ITU PA3, ITU PB3 and ITU VA30 channels, achieved by using the double pass, *mixed schemes* and double pass, *per path schemes*.

It is seen from Figures 8 and 9 that, for WG4 Case 2 and WG4 Case 3 channels, there is not much difference between the secret key rates resulting from the *per path scheme* and from the *mixed scheme*. Recall from Table I that all 3 underlying paths in WG4 Case 2 channel have the same average power, and that the 4 underlying paths in WG4 Case 3 channel have similar average path powers. Hence, the replacement of individual SNRs by the reference SNR will not cause recognizable performance loss. For other channels, the difference in the average path powers leads to a large performance loss by using the *mixed scheme*.

In these figures, steep increases (observed as kinks in the performance curve) can be observed in all secret key rate curves for the *per path scheme*. The reason is that, when the SNR passes a certain value, the given LDPC code is able to correct all the errors between Alice and Bob's CIR post-processed samples from a path. At this point, the resulting secret key bits from the additional detected path begin contributing, resulting in a steep secret key rate increase.

Since all the underlying paths for WG4 Case 1/2/3 channels can be detected in the path searcher (cf. Table II), the achieved secret key rates have almost the same slopes as the upper bounds. For these channels, the gaps between the achieved secret key rates and the corresponding upper bounds are mainly migrated from the gap reported in [4]. Similar arguments hold for ITU PB3 and ITU VA30 channels.

A large gap between the achieved secret key rate and the upper bound for ITU PA3 channel is observed in Figure 10. An obvious reason is that only 1 or 2 of the 4 underlying paths are detected by the path searcher. The achieved secret key rate is based on the detected paths, while the upper bound (6) is derived from all 4 underlying paths. To justify the above assertions, we also plot a new "upper bound" based on the first two paths of the ITU PA3 channel in Figure 10. It is seen that the achieved secret key rate has almost the same slope as this new "upper bound."

The achieved secret key rates for WG4 and ITU channels at an SNR of 30 dB are listed in Table III. The table shows that the WG4 Case 2 channel has the largest key rates when normalized to the number of paths. This is due to the high average path powers in the WG4 Case 2 channel.

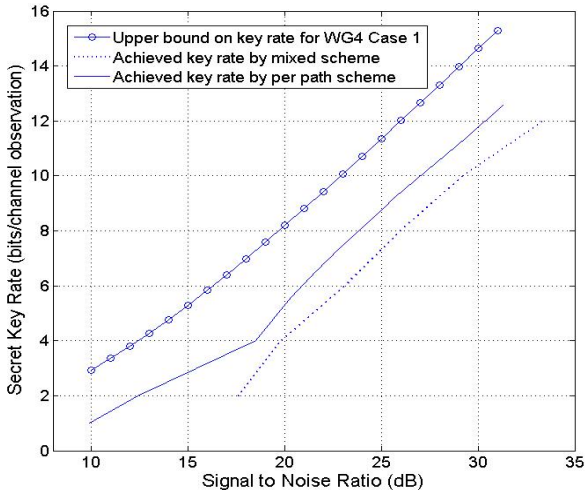


Figure 7: Secret key rate for WG4 Case 1 channel

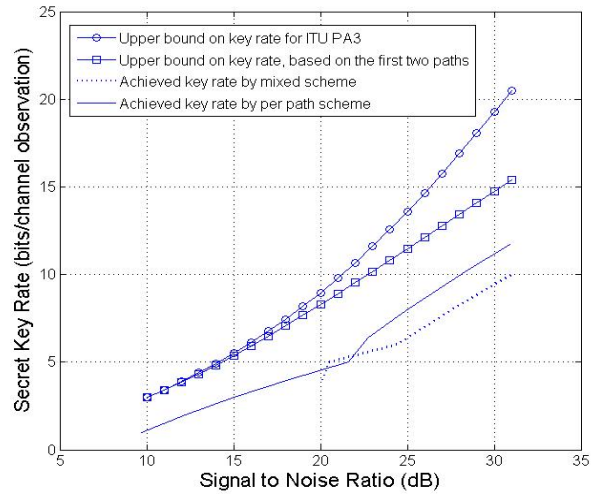


Figure 10: Secret key rate for ITU PA3 channel

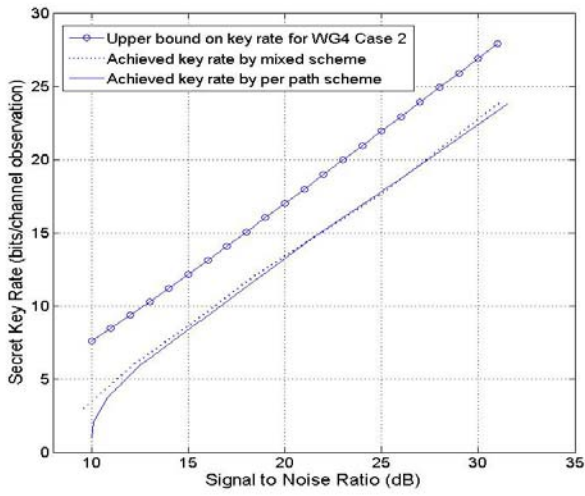


Figure 8: Secret key rate for WG4 Case 2 channel

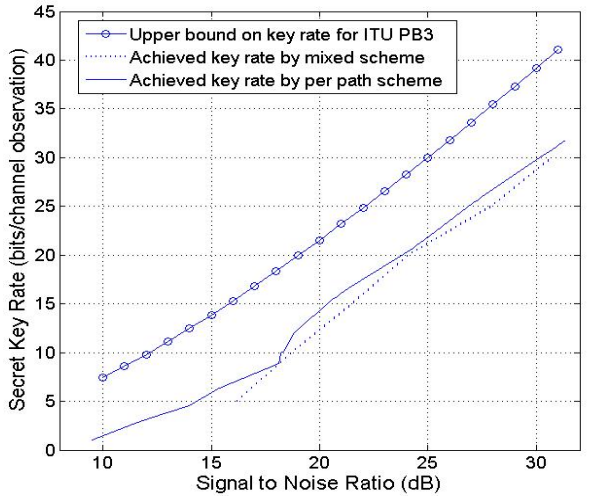


Figure 11: Secret key rate for ITU PB3 channel

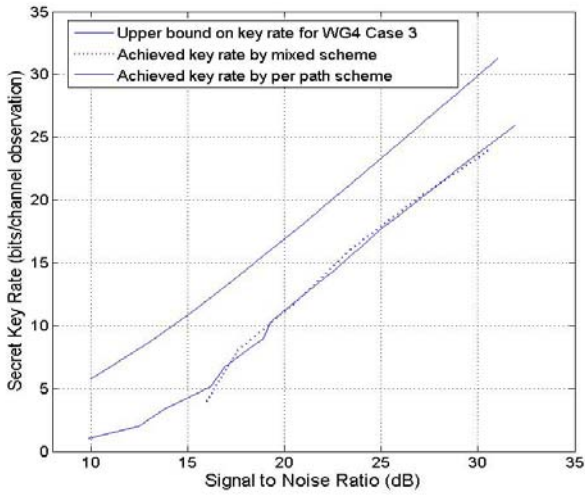


Figure 9: Secret key rate for WG4 Case 3 channel

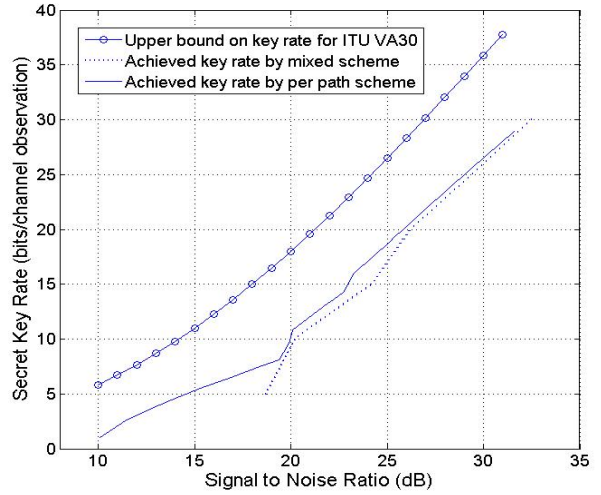


Figure 12: Secret key rate for ITU VA30 channel

Table III: ACHIEVED SECRET KEY RATES AT SNR=30 DB

Channel model	WG4 Case 1	WG4 Case 2	WG4 Case 3	ITU PA3	ITU PB3	ITU VA30
Overall secret key rates (bits/channel observation)	11.8	22.8	23.7	11.2	29.8	26.5
# underlying paths	2	3	4	4	6	6
Normalized secret key rates (bits/channel observation/path)	5.9	7.6	5.9	2.8	5.0	4.4

V. DISCUSSION AND FUTURE WORK

In this paper, we presented an OGA-based algorithm used for channel decomposition and then combined with techniques in [5] for secrecy generation from multipath fading channels in the non-UWB scenarios. While we observe that our technique generally achieves the same slope as the secrecy upper bound, a significant gap to the upper bound still exists. The main reason for the gap is that we apply the techniques developed in [5] in our secret key generation system. The 1-bit gap observed by [5] translates to a 1-bit gap *per* path in our simulation results. Hence, a good approach to increasing the achieved secret key rates for multipath fading channels is to close the gap for a single path Rayleigh fading channel.

This work has focused on extracting secrecy bits from independent observations. An issue not addressed here is the time correlation of multipath channels to create a stream of bits. One simple approach would be to estimate the coherence time of the channel and align the CIR sampling to that coherence time. However, this would probably introduce losses, and an approach capable of addressing time dependent path-vector observations would still need to be provided.

Finally, the analysis presented in this paper is based on the computer-simulated CIR samples using industry-accepted CIR models. While these are designed to be representative of common channel conditions they are not exact and may be a poor representation of conditions encountered in certain environments. More importantly, these are designed to stress data transmission capabilities of communication systems, not their secrecy generation capabilities. Practical CIR measurements are needed to establish true secrecy-generation capabilities of wireless channels.

REFERENCES

- [1] W. C. Jakes, Ed., *Microwave Mobile Communications*, IEEE Press, Piscataway, NJ, 1974.
- [2] R. Wilson, D. Tse and R. Scholtz, "Channel identification: Secret sharing using reciprocity in UWB channels," submitted to *IEEE Trans. Inform. Forensics. and Security*. (available at <http://www.eecs.berkeley.edu/~dtse>)
- [3] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography — part I: Secret sharing," *IEEE Trans. Inf. Theory*, IT-39:1121–1132, 1993.
- [4] U. Maurer, "Secret key agreement by public discussion," *IEEE Trans. Inf. Theory*, IT-39:733–742, 1993.
- [5] C. Ye, A. Reznik and Y. Shah, "Extracting secrecy from jointly Gaussian random variables," *Proceedings ISIT 2006*, July 2006.
- [6] 3GPP TS 25.101 User Equipment (UE) radio transmission and reception (FDD) (Release 6), rev. 6.7.0, 12/2005.
- [7] Q. Wang, S. R. Kulkarni and S. Verdú, "Divergence estimation for continuous distributions based on data-dependent partitions," *IEEE Trans. Inf. Theory* IT-51, 2005.
- [8] Q. Wang, S.R. Kulkarni, S. Verdú, "A nearest-neighbor approach to estimating divergence between continuous random vectors," *Proceedings ISIT 2006*, July 2006.
- [9] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, IT-49, pp. 3320–3325, 2003.
- [10] D. L. Donoho, M. Elad and V. L. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, IT-52, pp.6–18, 2006.
- [11] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inf. Theory*, IT-47, pp. 2845–2862, 2001.
- [12] D. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization," *Proc. Nat. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, Mar. 2003.
- [13] M. Elad and A. Bruckstein, "A generalized uncertainty principle and sparse representations in pairs of bases," *IEEE Trans. Inf. Theory*, IT-48, pp. 2558–2567, 2002.
- [14] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, IT-50, pp. 2231–2242, 2004.