Secret Key Generation over Correlated Wireless Fading Channels using Vector Quantization

Hou-Tung Li and Y.-W. Peter Hong
Institute of Communications Engineering
National Tsing Hua University
Hsinchu, Taiwan 30013
E-mail: htli@erdos.ee.nthu.edu.tw and ywhong@ee.nthu.edu.tw

***Abstract*—Vector quantization schemes are proposed to extract secret keys from correlated wireless fading channels. By assuming that the channel between two terminals are reciprocal, its estimates can be used as the common randomness for generating secret keys at the two terminals. Most schemes in the literature assume that channels are independent over time and utilize scalar quantization on each element of the estimated channel vector to generate secret key bits. These schemes are simple to implement but yield high key disagreement probability (KDP) at low SNR and low key entropy when channels are highly correlated. In this work, two vector quantization schemes, namely, the minimum key disagreement probability (MKDP) and the minimum quadratic distortion (MQD) secret key generation schemes, are proposed to effectively extract secret keys from correlated channel estimates. The vector quantizers are derived using KDP and QD as the respective distortion measures. To further reduce KDP, each channel vector is first pre-multiplied by an appropriately chosen unitary matrix to rotate the vector away from quantization cell boundaries. The MKDP scheme achieves the lowest KDP but requires high complexity whereas the MQD scheme yields lower complexity but at the cost of slightly increased KDP. Computer simulations are provided to demonstrate the effectiveness of the proposed vector quantization schemes.**

I. Introduction

Due to the broadcast nature of the wireless medium, communication between terminals in a wireless environment are often susceptible to eavesdropping, message modifying, and node impersonation. Secret keys have thus been used to protect the confidentiality, integrity, and authenticity of messages and nodes. The generation and agreement of secret keys at the communicating terminals have traditionally been done in the network or application layers using, e.g., the Diffie-Hellman key agreement protocol [1]. However, these schemes rely on message exchanges between terminals, which could also be intercepted, and assume certain computational constraints at the adversaries. Alternatively, physical-layer approaches that do not require such assumptions have been examined, e.g., in [2]–[11], utilizing in particular the common randomness in the channel between legitimate users to generate secret keys.

Channel-based secret key generation schemes [5]–[11] allow terminals to extract common randomness from their locally estimated channels and utilize it as the random seed to locally generate secret keys at the terminals. When the channel shared by two terminals is reciprocal, their local estimates will be similar and, thus, the generated secret keys will agree with high probability. An eavesdropper located more than half a wavelength away will, on the other hand, experience independent fading [12] and, thus, will not be able to infer any information about the secret key from its local estimate of the channel. However, the channel estimates obtained at different terminals are often subject to discrepancies caused by noise and may lead to large key disagreement if their continuous values are used as the random seed for secret key generation. Hence, quantized versions of the channel estimates must be used instead in order to combat the effect of noise. These key generation schemes are designed with the goal of ensuring low key disagreement probability (KDP) among communicating terminals and also high key entropy so that the generated keys cannot be easily inferred by the eavesdropper.

The use of common randomness to generate secret keys at different terminals has first been studied in [2]–[4]. More recently, many works in the literature, e.g., in [5]–[11], have exploited specifically the common randomness in the channel to achieve this task. These schemes utilize quantization of the amplitude and/or phase of the channel to mitigate the effect of noise and to determine the common index of the secret key at the two terminals. In particular, the use of channel phase as the common randomness was considered in [5] and was applied to OFDM multipath channels in [6]. The use of channel amplitude was considered in [7]–[9]. These methods have also been extended to UWB systems in [10], [11]. However, these works assume that the channels observed at the users must be independent over time, which limits the secrecy generation rate in slow fading channels. The issue of channel correlation has been addressed in [13] by first decorrelating the channel observations and, then, allocating different quantization bits to each decorrelated dimension depending on its effective signal-to-noise ratio. However, the proposed bit allocation scheme is less systematic and requires integer number of bits to be allocated to each dimension, which is rather restrictive.

The main contribution of this work is to propose vector quantization schemes to efficiently extract secret keys from correlated wireless fading channels. Two vector quantization schemes are proposed: the minimum key disagreement probability (MKDP) and the minimum quadratic distortion (MQD) secret key generation schemes. These schemes are designed using Lloyd-Max-like algorithms with KDP and quadratic distortion (QD) as their respective distortion measures. The MKDP scheme achieves low KDP but requires

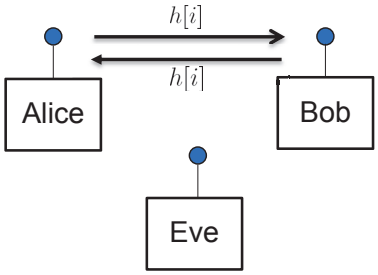

Fig. 1. Two legitimate users and an eavesdropper at the third location.

high complexity whereas the MQD yields low complexity but slightly higher KDP. Notice that, different from conventional source coding literature, quantization in secret key generation applications is used to reduce the discrepancy at the two terminals. The reconstruction or representation of the vectors in each quantization cell is not a concern. However, in this application, two users with a given quantizer may yield large KDP if their channel estimation vector lies at the boundary of the quantization cells since, in this case, small discrepancy between the two users' estimates may result in completely different quantizer outputs. Hence, in this work, we propose the use of appropriately chosen unitary transformations to rotate the channel vectors away from the cell boundaries before quantization The chosen unitary transformation can then be communicated between the two terminals without providing the eavesdropper with any information regarding the channel vector. Computer simulations are provided to demonstrate the effectiveness of the proposed vector quantization schemes.

The remainder of this paper is organized as follows. In Section II, the system model is described. In Section III, the general concept of quantization-based secret key generation is introduced. In Sections IV and V, two Lloyd-Max-like vector quantization methods are proposed and are later extended to the case with entropy constraints in Section **??**. Finally, simulations are provided in Section VI to demonstrate the effectiveness of our proposed schemes and a brief conclusion is given in Section VII.

II. System Model

Consider a system that consists of two users, Alice and Bob, transmitting confidential messages to each other using a common secret key, as shown in Fig. 1. All terminals are assumed to have only a single antenna element. The secret key is generated at each terminal based on its local estimate of the channel. Suppose that training signals are transmitted from Alice to Bob and from Bob to Alice in consecutive time slots at the beginning of each block, as shown in Figure 2. The channels observed by the two users are assumed to be constant in each block but may vary from block to block. The received signals at Alice and Bob (namely, nodes A and B in the remaining of the paper) in block i are given respectively by

$$y_a[i] = \sqrt{P}h[i] + n_a[i] \tag{1}$$

and

$$y_b[i] = \sqrt{P}h[i] + n_b[i] \tag{2}$$

where P is the power of the training signal and $n_a[i]$ and $n_b[i]$ are independent complex Gaussian random variables with mean 0 and variance σ_n^2, i.e., $n_a[i], n_b[i] \sim \mathcal{CN}(0, \sigma_n^2)$.

The channel estimates obtained over L blocks at Alice and Bob are represented as

$$\hat{\mathbf{h}}_a = [\hat{h}_a[1], \ldots, \hat{h}_a[L]]^T$$

and

$$\hat{\mathbf{h}}_b = [\hat{h}_b[1], \ldots, \hat{h}_b[L]]^T$$

where $\hat{h}_a[i] = y_a[i]/\sqrt{P}$ and $\hat{h}_b[i] = y_b[i]/\sqrt{P}$, for $i = 1, \ldots, L$. The covariance matrices $\mathbf{C}_{\hat{\mathbf{h}}_a}$ and $\mathbf{C}_{\hat{\mathbf{h}}_b}$ of $\hat{\mathbf{h}}_a$ and $\hat{\mathbf{h}}_b$ are given by

$$\mathbf{C}_{\hat{\mathbf{h}}_a} = \mathbf{C}_{\hat{\mathbf{h}}_b} = \mathbf{C}_{\mathbf{h}} + \frac{\sigma_n^2}{P}\mathbf{I} = \mathbf{U}\left(\mathbf{\Lambda} + \frac{\sigma_n^2}{P}\mathbf{I}\right)\mathbf{U}^H, \tag{3}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_L)$ is a diagonal matrix consisting of the eigenvalues of $\mathbf{C}_{\mathbf{h}}$ on its diagonal and $\mathbf{U}$ is a unitary matrix formed by the eigenvectors of $\mathbf{C}_{\mathbf{h}}$. Please note that the channel vector $\mathbf{h}$ can also be viewed as vectorization of the channels received over multiple antenna elements in MIMO systems and, thus, the following operations extend readily to the MIMO case as well.

By letting

$$\mathbf{D} = \mathbf{U}\left(\mathbf{\Lambda} + \frac{\sigma_n^2}{P}\mathbf{I}\right)^{\frac{1}{2}}$$

as the decorrelating matrix, we can obtain uncorrelated equivalent channels

$$\mathbf{g}_a \triangleq \mathbf{D}^{-1}\hat{\mathbf{h}}_a \tag{4}$$

with covariance $\mathbf{C}_{\mathbf{g}_a} = \mathbf{D}^{-1}\mathbf{C}_{\hat{\mathbf{h}}_a}\mathbf{D}^{-H} = \mathbf{I}$ and, similarly, $\mathbf{g}_b \triangleq \mathbf{D}^{-1}\hat{\mathbf{h}}_b$ with $\mathbf{C}_{\mathbf{g}_b} = \mathbf{I}$. The covariance of $\mathbf{g}_a$ (and also $\mathbf{g}_b$) can also be written as

$$\begin{aligned}\mathbf{C}_{\mathbf{g}_a} &= \mathbf{D}^{-1}\mathbf{C}_{\mathbf{h}}\mathbf{D}^{-H} + \frac{\sigma_n^2}{P}\mathbf{D}^{-1}\mathbf{D}^{-H} \\ &= \mathrm{diag}\left(\frac{\lambda_1}{\lambda_1 + \sigma_n^2/P}, \ldots, \frac{\lambda_L}{\lambda_L + \sigma_n^2/P}\right) \\ &\quad + \mathrm{diag}\left(\frac{\sigma_n^2/P}{\lambda_1 + \sigma_n^2/P}, \ldots, \frac{\sigma_n^2/P}{\lambda_L + \sigma_n^2/P}\right),\end{aligned} \tag{5}$$

where the first term corresponds to the covariance of the signal component and the second term corresponds to that of the noise component. That is, even though the entries in $\mathbf{g}_a$ are independent and identically distributed (i.i.d.), their signal to noise ratios (SNRs) are different and are given by $\lambda_1 P/\sigma_n^2$, …, $\lambda_L P/\sigma_n^2$, respectively.

In the following, we assume that $\mathbf{g}_a$ and $\mathbf{g}_b$ are utilized as the random seed to generate secret keys at nodes A and B, respectively. In the noiseless case, we have $\mathbf{g}_a = \mathbf{D}^{-1}\mathbf{h} = \mathbf{g}_b$ and, thus, the secret keys generated by the two terminals will be identical. However, in practice, $\mathbf{g}_a$ will be equal to $\mathbf{g}_b$ plus noise. In this case, quantized information would be more robust since two nodes will observe the same random seed as

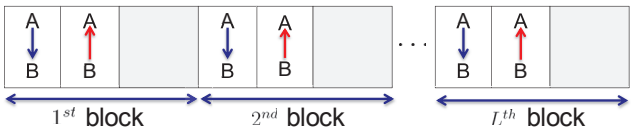

Fig. 2. Relationship between channel training and coherence block in the half-duplex communication system.

long as $\mathbf{g}_a$ and $\mathbf{g}_b$ fall in the same quantization cell. In [13], secret keys are generated by performing scalar quantization separately on each entry of $\mathbf{g}_a$ and the heterogeneity of the SNRs are taken into consideration by heuristically allocating different number of quantization bits to each entry of $\mathbf{g}_a$. However, the resulting quantizer in this case is suboptimal. In this work, optimized vector quantization schemes are used to generate secret keys from the channel vectors $\mathbf{g}_a$ and $\mathbf{g}_b$, respectively, at nodes A and B. A general description of the vector quantization based secret key generation scheme is given in the following section.

III. Secret Key Generation Using Vector Quantization: General Concept

Let us define a vector quantizer $Q : \mathbb{C}^L \to \{1, \ldots, K\}$ as a mapping from an L-dimensional input vector $\mathbf{g}$ to an integer from 1 to K, where K is the number of secret keys. Since L channel observations are used to generate K secret keys, the secret key generation rate is $\log_2 K/L$ bits per observation. The quantizer divides the L-dimensional Euclidean space into K partitions (or quantization cells) $\mathcal{R}_1, \cdots, \mathcal{R}_K$, where

$$\mathcal{R}_k = \{\mathbf{g} \in \mathbb{C}^L : Q(\mathbf{g}) = k\} \tag{6}$$

for $k = 1, \ldots, K$. The same quantizer Q is used at both nodes A and B to generate their secret keys. However, if the decorrelated channel vectors $\mathbf{g}_a$ or $\mathbf{g}_b$ are used directly as the input of the quantizer Q, the two nodes will likely yield different quantizer outputs when the signal component $\mathbf{D}^{-1}\mathbf{h}$ lies close to the boundary of a quantization cell. This is because, if $\mathbf{D}^{-1}\mathbf{h}$ lies at the boundary of a quantization cell, the noise component in $\mathbf{g}_a$ and $\mathbf{g}_b$ will lead to discrepancies between the two vectors and, thus, cause them to lie in two different quantization cells or partitions. This effect often dominates the key disagreement probability (KDP) in quantization-based secret key generation schemes.

To address this issue, we propose to pre-multiply each channel vector $\mathbf{g}_a$ and $\mathbf{g}_b$ by a rotation matrix $\mathbf{U}$ to move it away from the quantization cell boundaries before quantization. The matrix $\mathbf{U}$ is chosen from a predetermined set

$$\mathcal{U} = \{\mathbf{U}_1, \ldots, \mathbf{U}_N\}, \tag{7}$$

where

$$\mathbf{U}_n = \text{diag}\left(e^{j\theta_{n1}}, \ldots, e^{j\theta_{nL}}\right) \tag{8}$$

and $\theta_{n\ell} \in [0, 2\pi]$, for $n = 1, \ldots, N$ and $\ell = 1, \ldots, L$. During each secret key generation process, Alice first performs quantization on the vectors $\mathbf{U}_1\mathbf{g}_a$, …, $\mathbf{U}_N\mathbf{g}_a$, and selects the matrix $\mathbf{U}$ among the set $\mathcal{U}$ that yields the minimum KDP based on its statistical knowledge of $\mathbf{g}_b$, i.e.,

$$\mathbf{U} = \arg\min_{\mathbf{U}\in\mathcal{U}} \Pr\left(Q(\mathbf{U}\mathbf{g}_a) \neq Q(\mathbf{U}\mathbf{g}_b) | \, \mathbf{g}_a\right). \tag{9}$$

The index of $\mathbf{U}$ is then sent to Bob, who then pre-multiplies its observation $\mathbf{g}_b$ with the same rotation matrix $\mathbf{U}$. Notice that, by choosing $\mathbf{U}$ to take on the form in (8), the statistics of $\mathbf{g}_a$ (or $\mathbf{g}_b$) do not change and, thus, the optimality of the quantizer Q, derived in later sections, is not violated. It is interesting to remark that $Q(\mathbf{U}_1\mathbf{g}_a)$, …, $Q(\mathbf{U}_N\mathbf{g}_a)$ can be viewed equivalently as the output of N different quantizers $Q_1(\mathbf{g}_a)$, …, $Q_N(\mathbf{g}_a)$ and, indeed, the boundary effects can be addressed in more generality by considering N different quantizers instead of using the N different rotation matrices mentioned above. However, the complexity of generating a quantizer is much higher than that of generating the rotation matrices in (8) and, thus, the latter is employed in this work.

It is interesting to note that, even with well designed quantizers, the nodes will always be subject to key disagreement due to noise. In practice, the key discrepancies can be overcome through information reconciliation (i.e., error correction) [5], [7] or by simply regenerating the keys at the two terminals [6]. In this work, these operations are assumed to be inherent in the system and we focus on designing quantizers to minimize the rate of key disagreement.

In the following sections, two Lloyd-Max-like quantizers are derived for secret key generation, utilizing KDP and QD as the distortion measures.

IV. Minimum Key Disagreement Probability (MKDP) Vector Quantization Scheme

The main objective of quantization-based secret key generation schemes is to minimize the probability that the two terminals employing the same quantizer yield different quantizer outputs, i.e., the KDP. In this section, a Lloyd-Max like vector quantizer is derived by using the KDP as the distortion criterion.

Specifically, by the choice of $\mathbf{U}$ in (9), the average KDP can be computed as

$$\begin{aligned}&\text{KDP}(Q)\\ &\triangleq \int_{\mathbf{g}_a} \left[\min_{\mathbf{U}\in\mathcal{U}} \Pr(Q(\mathbf{U}\mathbf{g}_a) \neq Q(\mathbf{U}\mathbf{g}_b) \mid \mathbf{g}_a)\right] \cdot f_{\mathbf{g}_a}(\mathbf{g}_a) d\mathbf{g}_a\end{aligned} \tag{10}$$

and the minimum KDP (MKDP) quantizer is chosen as

$$Q_{\text{MKDP}} = \arg\min_{Q} \text{KDP}(Q). \tag{11}$$

Following the Lloyd-Max philosophy [14], the MKDP quantizer can be found using the following iterative approach. However, notice that, in conventional quantization schemes, both an encoder and a decoder must be found. In the case of secret key generation, there is no explicit need to find decoders (or representation levels for the quantization cells). Theoretically, the iteration can be given as follows:

1) **Initialization:** Set $t = 0$. Initialize partition sets $\mathcal{R}_1^{(t)}$, ..., $\mathcal{R}_K^{(t)}$ and let

$$Q^{(t)}(\mathbf{g}_a) = k, \quad \text{if } \mathbf{g}_a \in \mathcal{R}_k^{(t)}$$

for $k = 1, \ldots, K$.

2) **Update:** Update the partition sets such that

$$\mathcal{R}_k^{(t+1)} = \{\mathbf{U}\mathbf{g}_a : \Pr(k \neq Q^{(t)}(\mathbf{U}\mathbf{g}_b) \mid \mathbf{g}_a) \leq \Pr(k' \neq Q^{(t)}(\mathbf{U}'\mathbf{g}_b) \mid \mathbf{g}_a), \forall k', \mathbf{U}'\}$$

for $k = 1, \ldots, K$.

3) Continue update until the difference between the sets $\{\mathcal{R}_1^{(t)}, \ldots, \mathcal{R}_K^{(t)}\}$ and $\{\mathcal{R}_1^{(t+1)}, \ldots, \mathcal{R}_K^{(t+1)}\}$ are negligible, e.g.,

$$\sum_{k=1}^{K} \left| \mathcal{R}_k^{(t)} \cup \mathcal{R}_k^{(t+1)} - \mathcal{R}_k^{(t)} \cap \mathcal{R}_k^{(t+1)} \right| / \sum_{k=1}^{K} \left| \mathcal{R}_k^{(t)} \right| \leq \delta.$$

In the update step, the vector $\mathbf{U}\mathbf{g}_a$ is included in the set $\mathcal{R}_k^{(t+1)}$ if the probability that $Q(\mathbf{U}\mathbf{g}_b) \neq k$ is the smallest among all k and $\mathbf{U}$. Note that, by choosing the best rotation matrix $\mathbf{U}$, each observation vector (e.g., $\mathbf{g}_a$) is rotated away from neighboring quantization cells before it is actually quantized. Hence, the vectors that are actually quantized (i.e., $\mathbf{U}\mathbf{g}_a$) are concentrated in smaller regions and the distance between the regions $\mathcal{R}_1^{(\infty)}, \ldots, \mathcal{R}_K^{(\infty)}$ will be larger. Therefore, the union of the partitions, i.e., $\cup_{k=1}^K \mathcal{R}_k^{(\infty)}$, may not cover entirely the space $\mathbb{C}^L$. Vectors $\mathbf{U}\mathbf{g}_a$ that fall outside of the set $\cup_{k=1}^K \mathcal{R}_k^{(\infty)}$ correspond to suboptimal choices of $\mathbf{U}$ and, thus, will never be the vectors that are actually quantized. Hence, these vectors can be assigned arbitrarily to any of the partitions since they do not affect the KDP performance.

Due to the irregularity of the partitions $\mathcal{R}_1$, ..., $\mathcal{R}_K$ generated from the above algorithm, the probabilities required in the update step will be difficult to compute explicitly. However, in practice, the partitions can be found using training vectors generated based on the statistics of $\mathbf{g}_a$ and $\mathbf{g}_b$. The more practical procedure is described below.

1) **Initialize:** Set $t = 0$. Initialize partition sets $\mathcal{R}_1^{(t)}$, ..., $\mathcal{R}_K^{(t)}$ and let

$$Q^{(t)}(\mathbf{g}_a) = k, \quad \text{if } \mathbf{g}_a \in \mathcal{R}_k^{(t)}$$

for $k = 1, \ldots, K$.

2) **Update:** Generate the set of M_a training vectors $\mathcal{G}_a = \{\mathbf{g}_{a,1}, \cdots, \mathbf{g}_{a,M_a}\}$ randomly according to the pdf $f_{\mathbf{g}_a}$.

3) For each $\mathbf{g}_{a,m} \in \mathcal{G}_a$, generate a set of M_b training vectors $\mathcal{G}_{b|\mathbf{g}_{a,m}} = \{\mathbf{g}_{b,1|\mathbf{g}_{a,m}}, \cdots, \mathbf{g}_{b,M_b|\mathbf{g}_{a,m}}\}$ according to the pdf $f_{\mathbf{g}_b|\mathbf{g}_{a,m}}$.

4) Construct the sets

$$\mathcal{S}_k^{(t+1)} = \left\{ \mathbf{U}\mathbf{g}_{a,m} : \frac{1}{M_b} \sum_{i=1}^{M_b} \mathbf{1}_{\{\mathbf{U}\mathbf{g}_{b,i|\mathbf{g}_{a,m}} \notin \mathcal{R}_k^{(t)}\}} \leq \frac{1}{M_b} \sum_{i=1}^{M_b} \mathbf{1}_{\{\mathbf{U}'\mathbf{g}_{b,i|\mathbf{g}_{a,m}} \notin \mathcal{R}_{k'}^{(t)}\}}, \ \forall k', \mathbf{U}' \right\}$$

5) Construct the partitions

$$\mathcal{R}_k^{(t+1)} = \{\mathbf{g} : \exists \mathbf{g}' \in \mathcal{S}_k^{(t+1)} \text{ such that } \|\mathbf{g} - \mathbf{g}'\| \leq \|\mathbf{g} - \widetilde{\mathbf{g}}\|, \ \forall \widetilde{\mathbf{g}} \in \cup_{i=1}^K \mathcal{S}_i^{(t+1)}\}$$

for $k = 1, \ldots, K$.

6) Continue update until the difference between the sets $\{\mathcal{S}_1^{(t)}, \ldots, \mathcal{S}_K^{(t)}\}$ and $\{\mathcal{S}_1^{(t+1)}, \ldots, \mathcal{S}_K^{(t+1)}\}$ are negligible, e.g.,

$$\sum_{k=1}^{K} \left| \mathcal{S}_k^{(t)} \cup \mathcal{S}_k^{(t+1)} - \mathcal{S}_k^{(t)} \cap \mathcal{S}_k^{(t+1)} \right| / \sum_{k=1}^{K} \left| \mathcal{S}_k^{(t)} \right| \leq \delta.$$

In Step 4 of the algorithm, $\frac{1}{M_b} \sum_{i=1}^{M_b} \mathbf{1}_{\{\mathbf{U}\mathbf{g}_{b,i|\mathbf{g}_{a,m}} \notin \mathcal{R}_k^{(t)}\}}$ is used as an approximation of the probability $\Pr(k \neq Q^{(t)}(\mathbf{U}\mathbf{g}_b)|\mathbf{g}_a)$. In this step, the training vectors are associated to the set $\mathcal{S}_k^{(t+1)}$ that yields the minimum approximate KDP. The partition $\mathcal{R}_k^{(t+1)}$ is then chosen to include all vectors $\mathbf{g}$ that is closest to one of the vectors in $\mathcal{S}_k^{(t+1)}$. However, when the number of training vectors is large, the computation of the partitions $\mathcal{R}_1^{(t+1)}$, ..., $\mathcal{R}_K^{(t+1)}$ according to the procedures in Step 5 require large memory. To reduce the complexity, these partitions can be further approximated using the centroid approximation, i.e.,

$$\mathcal{R}_k^{(t+1)} \approx \left\{ \mathbf{g} : \left\| \mathbf{g} - \bar{\mathbf{g}}_k^{(t+1)} \right\| \leq \left\| \mathbf{g} - \bar{\mathbf{g}}_{k'}^{(t+1)} \right\|, \ \forall k' \right\}, \quad (12)$$

where $\bar{\mathbf{g}}_k^{(t+1)} = \sum_{\mathbf{g} \in \mathcal{S}_k^{(t+1)}} \frac{\mathbf{g}}{|\mathcal{S}_k^{(t+1)}|}$ is the centroid of the partition $\mathcal{R}_k^{t+1}$. Namely, instead of searching for the minimum distance among all the vectors in $\cup_{i=1}^K \mathcal{S}_i^{(t+1)}$, the centroid approximation reduces the complexity by searching for the minimum distance among K centroids $\{\bar{\mathbf{g}}_1^{(t+1)}, \ldots, \bar{\mathbf{g}}_K^{(t+1)}\}$.

The performance of the MKDP scheme is demonstrated through simulations in Section VI. However, even though the MKDP yields the minimum KDP, the complexity is relatively high due to the large number of training vectors needed for both $\mathbf{g}_a$ and $\mathbf{g}_b$. By compromising slightly the MKDP, we show in the following section that an algorithm with lower complexity can be obtained, utilizing the quadratic distortion as the performance measure.

V. Minimum Quadratic Distortion (MQD) Vector Quantization Scheme

In this section, we reduce the complexity of the quantizer design by considering instead the quadratic distortion measure and decouple the quantizer design and the rotation selection problems. The key idea is to map each channel observation vector $\mathbf{g}_a$ at node A (or $\mathbf{g}_b$ at node B) to a vector $\mathbf{x}$ in a finite set $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \subset \mathbb{C}^L$ such that this vector is closest to the true vector $\mathbf{g} = \mathbf{D}^{-1}\mathbf{h}$. Since both $\mathbf{g}_a$ and $\mathbf{g}_b$ are noisy versions of $\mathbf{g}$, they will both associate to the same vector $\mathbf{x}$ with high probability and, thus, reduce the KDP.

Specifically, let us define $X : \{1, \ldots, K\} \to \mathbb{C}^L$ as a mapping that associates each output of the quantizer Q to a vector in $\mathbb{C}^L$. The vectors $X(k) = \mathbf{x}_k$, for $k = 1, \ldots, K$, can be viewed as the representation level or the decoder

in conventional quantization literature [14]. However, in the MQD scheme, the quadratic distortion is defined between the observation vector $\mathbf{g}_a$ at node A (or $\mathbf{g}_b$ at node B) and the noiseless vector $\mathbf{g} \triangleq \mathbf{D}^{-1}\mathbf{h}$. It is defined as

$$\mathrm{QD}(Q, X) = \mathbf{E}[\|X(Q(\mathbf{g}_a)) - \mathbf{g}\|^2] \tag{13}$$

and the MQD quantizer is chosen as

$$Q_{\mathrm{MQD}} = \arg\min_{Q} \left\{ \min_{X} \mathrm{QD}(Q, X) \right\}. \tag{14}$$

Notice that the search for the optimal Q and X above is similar to the problem of finding the optimal encoder and decoder in conventional Lloyd-Max iterative quantization design procedure. Motivated by this approach, we propose in the following an iterative approach where Q and X are optimized interchangeably while the other is fixed.

A. Optimization of Q for given X

In this subsection, we first optimize Q for a given choice of X (i.e., the set $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$. Notice that, in this case, the QD in (13) can be computed as

$$\begin{aligned}
&\mathbb{E}[\|X(Q(\mathbf{g}_a)) - \mathbf{g}\|^2] \\
&= \sum_{k=1}^{K} \int_{\mathbf{g}'_a \in \mathcal{R}_k} \mathbb{E}[\|\mathbf{x}_k - \mathbf{g}\|^2 \mid \mathbf{g}'_a] f_{\mathbf{g}_a}(\mathbf{g}'_a) d\mathbf{g}'_a \\
&= \sum_{k=1}^{K} \int_{\mathbf{g}'_a \in \mathcal{R}_k} \left(\int_{\mathbf{g}'} \|\mathbf{x}_k - \mathbf{g}'\|^2 f_{\mathbf{g}|\mathbf{g}_a}(\mathbf{g}'|\mathbf{g}'_a) d\mathbf{g}' \right) f_{\mathbf{g}_a}(\mathbf{g}'_a) d\mathbf{g}'_a \\
&= \sum_{k=1}^{K} \int_{\mathbf{g}'_a \in \mathcal{R}_k} D_e(\mathbf{g}'_a, \mathbf{x}_k) f_{\mathbf{g}_a}(\mathbf{g}'_a) d\mathbf{g}'_a
\end{aligned} \tag{15}$$

where

$$D_e(\mathbf{g}_a, \mathbf{x}_k) = \int_{\mathbf{g}} \|\mathbf{x}_k - \mathbf{g}\|^2 f_{\mathbf{g}|\mathbf{g}_a}(\mathbf{g}|\mathbf{g}_a) d\mathbf{g} \tag{16}$$

is the average distortion between $\mathbf{g}$ and $\mathbf{x}_k$ when conditioned on $\mathbf{g}_a$. From (15), we can see that, to minimize the overall average distortion, the observation vector $\mathbf{g}_a$ should be associated with the representation vector $\mathbf{x}_k$ that yields the minimum value of $D_e(\mathbf{g}_a, \mathbf{x}_k)$. The optimal quantizer for given X is then given by $Q_{MQD}(\mathbf{g}_a) = k$ if $\mathbf{g}_a \in \mathcal{R}_k$, where

$$\mathcal{R}_k = \{\mathbf{g}_a \mid D_e(\mathbf{g}_a, \mathbf{x}_k) \le D_e(\mathbf{g}_a, \mathbf{x}_\ell),\ \forall \ell\} \tag{17}$$

for $k = 1, \ldots, K$. Note that the conditionally average distortion can be computed explicitly as

$$D_e(\mathbf{g}_a, \mathbf{x}_k) = \sum_{\ell=1}^{L} \left| x_k[\ell] - \frac{\lambda_\ell g_a[\ell]}{\lambda_\ell + \sigma_n^2/P} \right|^2 + \sum_{\ell=1}^{L} \frac{\lambda_\ell \sigma_n^2/P}{(\lambda_\ell + \sigma_n^2/P)^2},$$

where $x_k[\ell]$ and $g_a[\ell]$, for $\ell = 1, \ldots, L$, are the L-th entries of $\mathbf{x}_k$ and $\mathbf{g}_a$, respectively. Since the last term is a constant, the optimal quantizer can be found as

$$Q(\mathbf{g}_a) = \arg\min_{k \in \{1, \cdots, K\}} \sum_{\ell=1}^{L} \left| x_k[\ell] - \frac{\lambda_\ell g_a[\ell]}{\lambda_\ell + \sigma_n^2/P} \right|^2, \tag{18}$$

which is easily computable.

B. Optimization of X for given Q

In this subsection, we optimize X for a given choice of Q (i.e., the partitions $\mathcal{R}_1, \ldots, \mathcal{R}_K$). Notice that, in the source coding literature [14], the function X represents the decoding or reconstruction of the quantized variable but, in the case of secret key generation, it is only an auxiliary operation used to facilitate our search of Q since only the index of the secret key is needed in the latter case.

To find the optimal $\mathbf{x}_1, \ldots, \mathbf{x}_K$ to minimize the QD, we can take the derivative of the QD with respect to each $\mathbf{x}_k$ and set it to zero. Specifically, by taking the derivative of the QD given in (15), the optimal values of $\mathbf{x}_1, \ldots, \mathbf{x}_K$ can be computed as

$$\mathbf{x}_k = \mathbb{E}\left[\mathbf{g} \mid \mathbf{g}_a \in \mathcal{R}_k\right] \tag{19}$$

for $k = 1, \ldots, K$.

By the optimization procedures given in the above subsections, an iterative algorithm can be proposed to find the optimal Q_{MQD} by optimizing over Q and X interchangeably while fixing the other. The iterative procedure can be described as follows.

1) **Initialize:** Set $t = 0$ and set initial values for $\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_K^{(0)}$.
2) Find $Q_{\mathrm{MQD}}^{(0)}$ (or, equivalently, $\mathcal{R}_1^{(0)}, \ldots, \mathcal{R}_K^{(0)}$) based on $\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_K^{(0)}$ and (17).
3) **Update:** Compute $\mathbf{x}_1^{(t+1)}, \ldots, \mathbf{x}_K^{(t+1)}$ by
$$\mathbf{x}_k^{(t+1)} = \mathbb{E}\left[\mathbf{g} \mid \mathbf{g}_a \in \mathcal{R}_k^{(t)}\right]$$
for $k = 1, \ldots, K$.
4) Find $Q_{\mathrm{MQD}}^{(t+1)}$ (or, equivalently, $\mathcal{R}_1^{(t+1)}, \ldots, \mathcal{R}_K^{(t+1)}$) based on $\mathbf{x}_1^{(t+1)}, \ldots, \mathbf{x}_K^{(t+1)}$ and (17).
5) Repeat update until
$$|\mathrm{QD}(Q_{\mathrm{MQD}}^{(t)}, X^{(t)}) - \mathrm{QD}(Q_{\mathrm{MQD}}^{(t+1)}, X^{(t+1)})| \le \delta.$$

After obtaining the quantizer Q_{MQD}, the rotation matrices are also employed to reduce the KDP between the Alice and Bob. The rotation matrix is chosen such that the rotated vector $\mathbf{U}\mathbf{g}_a$ is closest to its representation level $X(Q(\mathbf{U}\mathbf{g}_a))$, i.e.,

$$\mathbf{U} = \arg\min_{\mathbf{U} \in \mathcal{U}} \|X(Q(\mathbf{U}\mathbf{g}_a)) - \mathbf{U}\mathbf{g}_a\|^2. \tag{20}$$

This ensures that the vector to be quantized does not lie at the boundary of the quantization cells. The performance of the MQD scheme is also demonstrated through simulations in Section VI.

VI. Simulation Results

In this section, the effectiveness of the proposed vector-quantization-based secret key generation schemes are demonstrated through computer simulations. The performance of these schemes are compared with conventional schemes proposed in [13] that perform scalar quantization over entries in $\mathbf{g}_a$ (or $\mathbf{g}_b$). In our experiments, channel correlation is modeled using a Gauss-Markov process, where

$$h[i+1] = \alpha h[i] + \sqrt{1-\alpha^2} w[i], \quad 0 \le \alpha \le 1, \tag{21}$$

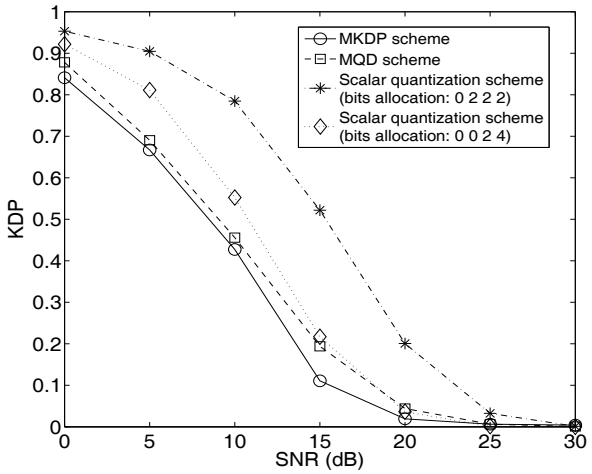

Fig. 3. KDP of the MKDP, MQD and the scalar quantization schemes when $\alpha = 0.9$ and key rate 1.5 bits per measurement.

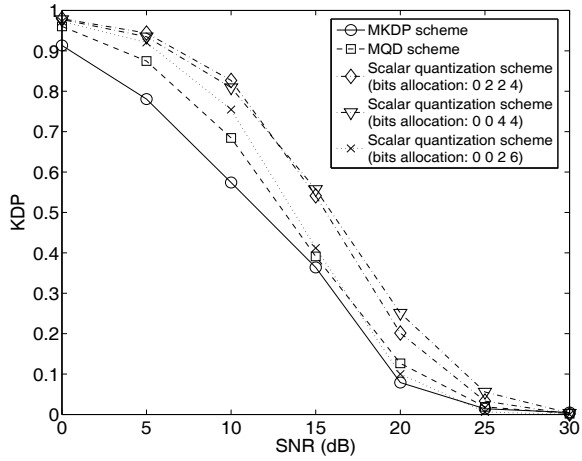

Fig. 4. KDP of the MKDP, MQD and the scalar quantization schemes when $\alpha = 0.9$ and key rate 2 bits per measurement.

where $h[i]$ represents the channel in the i-th block and $\{w[i]\}_{i=1}^{L}$ is i.i.d. over time. Both $h[i]$ and $w[i]$ are assumed to complex Gaussian with mean 0 and variance σ_h^2. Here, we set $\alpha = 0.9$ and $L = 4$. We consider key generation rates: $\log_2 K/L = 1.5$ and 2 bits per measurement (i.e., $K = 2^6$ and 2^8). Moreover, we consider a set $\mathcal{U}$ with $N = 16$ rotation matrices. The phases in each diagonal element of the rotation matrices are chosen randomly according to the uniform distribution between $[0, 2\pi]$.

In Figs. 3 and 4, the KDP of the MKDP, the MQD, and conventional scalar quantization schemes are compared for key generation rates of 1.5 and 2 bits per measurement, respectively. In the case with 1.5 bits per measurement (i.e., Fig. 3), two bit allocation policies are considered: the case with $(0, 2, 2, 2)$ bits allocated to the $L = 4$ entries of $\mathbf{g}_a$ with SNR from low to high and the case $(0, 0, 2, 4)$ bits allocated correspondingly. In the case with 2 bits per measurement (i.e., Fig. 4), three bit allocation policies are considered: the case with $(0, 2, 2, 4)$ bits, the case with $(0, 0, 4, 4)$ bits, and the case with $(0, 0, 2, 6)$ bits. In both figures, we can see that the MKDP scheme always achieves the minimum KDP among all schemes. The MQD scheme performs well in most cases but has a slight degradation at high SNR. This is due to the fact that, without noise, the distribution of $\mathbf{g}_a$ is less smooth and the centroid computed in (19) may be close to the boundary of the cell, resulting in undesired KDP. Moreover, we can also see that the bit allocation policies that can be adopted in the scalar quantization scheme are limited and generally achieve suboptimal performance.

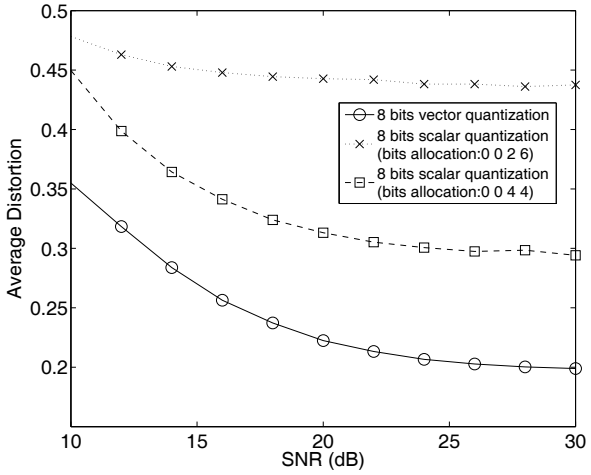

Fig. 5. Average distortion between the MQD scheme and the scalar quantization scheme when $\alpha = 0.9$ and key rate 2 bits per measurement.

In Fig. 5, we show the quadratic distortion between the quantized $\mathbf{g}_a$ and the noiseless channel $\mathbf{g}$. We can see that the MQD scheme indeed achieves the lowest QD compared to scalar quantization schemes. Utilizing the QD reduces the design complexity but in general does not always lead to minimum KDP as can be observed in Figs. 3 and 4.

VII. Conclusions

In this work, vector quantization schemes were proposed for secret key generation over correlated channels. Two quantization schemes were proposed, namely, the minimum quadratic distortion (MQD) and the minimum key disagreement probability (MKDP) secret key generation schemes. These schemes were derived by using Lloyd-Max-like algorithms with KDP and QD as the respective distortion measures. The MKDP scheme achieves the smallest KDP among all schemes but requires high complexity whereas the MQD yields low complexity but may have a slightly higher KDP than conventional schemes at high SNR. Moreover, to further reduce KDP, rotation matrices were imposed on the channel vectors before quantization to move the channel vectors away from the quantization cell boundaries. The rotation matrix does not change the statistics of the channel and thus does not affect

the optimality of the quantization schemes. The effectiveness of the proposed schemes over other schemes in the literature were demonstrated through computer simulations.

REFERENCES

[1] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, Nov./Dce. 1999.

[2] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, Oct. 1949.

[3] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inform. Theory*, vol. 39, pp. 733–742, May 1993.

[4] U. Maurer and S. Wolf, "Secret-key agreement over unauthenticated public channels," *IEEE Trans. Inform. Theory*, vol. 49, pp. 822–838, Apr. 2003.

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

[6] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Las Vegas, Nevada, March 31 2008-April 4 2008, pp. 3013–3016.

[7] B. Azimi-Sadjadi, A. Mercado, A. Kiayias, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *Proc. ACM Comput. Commun. Security*, Oct. 2007, pp. 401–410.

[8] Z. Li, W. Xu, R. Miller, and W. Trappe, "Securing wireless systems via lower layer enforcements," in *Proc. Fifth ACM Workshop Wireless Security*, 2006.

[9] S. Jana, S. N. Premnath, M. Clark, S. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *International Conference on Mobile Computing and Networking*. Beijing, China: ACM, 2009.

[10] R. Wilson, D. Tse, and R. A. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Trans. Information Forensics and Security*, vol. 2, pp. 364–375, 2007.

[11] M. G. Madiseh, M. L. McGuire, S. S. Neville, L. Cai, and M. Horie, "Secret key generation and agreement in UWB communication channels," in *Proc. IEEE GLOBECOM*, 2008.

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

[13] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Trans. Mobile Computing*, vol. 9, no. 1, pp. 17–30, Jan. 2009.

[14] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer, Nov. 1991.

[15] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 37, pp. 31–42, Jan. 1989.