Construction of a Shared Secret Key Using Continuous Variables

Jean Cardinal, Gilles Van Assche
Université Libre de Bruxelles
B-1050 Brussels, Belgium
{**jcardin, gvanassc**}**@ulb.ac.be**

Abstract — **Motivated by recent advances in quantum cryptography with continuous variables, we study the problem of extracting a shared digital secret key from two correlated real values. Alice has access to a real value X_A, and Bob to another value X_B such that $I(X_A; X_B) > 0$. They wish to convert their values into a shared secret digital information while leaking as little information as possible to Eve. We show how the problem can be decomposed in two subproblems known in other contexts. The first is the design of a *quantizer* that maximizes a mutual information criterion, the second is known as *coding with side information*.**

I. Introduction

The work presented in this paper[1] is motivated by some recent quantum key distribution (QKD) protocols that make use of continuous quantum states instead of discrete ones.

Quantum key distribution (also called quantum cryptography) allows Alice and Bob to share a secret key that can be used for encrypting messages. Eavesdropping is detectable in such key distribution schemes, as the laws of quantum mechanics imply that measuring a quantum state generally disturbs it. Quantum cryptography uses two channels: a quantum channel (e.g., a fiber in which single photons are sent) and a classical public authenticated channel. To share a secret key, a few steps must be performed. First, quantum states are sent from Alice to Bob on the quantum channel. This process gives the two parties correlated random variables, X_A and X_B. Then, Alice and Bob compare a sample of the transmitted information over the public channel. By measuring some appropriate disturbance metric, they determine an upper bound on the amount of information a possible eavesdropper was able to get, thanks to the laws of quantum mechanics. Finally, they extract a common secret key out of X_A and X_B. The last step of QKD, namely the construction of a common secret key out of correlated random variables is a non-trivial operation. In many QKD schemes such as BB84 [2], X_A and X_B are simply balanced binary random variables, connected by some error probability $\epsilon = \Pr[X_A \neq X_B]$. In this case, the secret key distillation usually involves two steps. First, Alice and Bob use a correction protocol over the public authenticated channel in such a way that they get identical keys. Since the correction information is sent over a public channel, it is considered known to an eavesdropper. Therefore, the second step of key distillation consists in applying a *privacy amplification* protocol [3], where the tapped information is wiped out at the cost of a reduction in the key length. Privacy amplification is not covered in this paper, since the currently developed protocols can readily be used. Unlike binary QKD protocols, some recent protocols [7] use a continuous modulation of quantum states, thus producing continuous random variables $X_A, X_B \in \mathbb{R}^d$. The extraction of a common secret key works like for their binary counterparts, although the reconciliation step will extract common *discrete* variables out of continuous ones. We thus wish Alice and Bob to be able to agree on a discrete key from X_A and X_B while leaking as little information as possible on the public channel.

We propose a three-phase approach to the problem of constructing a shared secret key K. In a first phase, Alice maps her value X_A to an integer $K = \alpha(X_A)$ using a predefined function α. Then she sends a *correction* information $\gamma(K)$ on the authenticated channel to Bob. Finally, using this information and his continuous value X_B, Bob is able to determine $K = \beta(\gamma(K), X_B)$ with high probability. The subject of this paper is the design of mappings α, γ and β that maximize the amount of shared secret information. In the source coding terminology, α is a *quantizer*, and the pair (γ, β) is a *lossless code with side information at the receiver* [10]. We have therefore split our problem in two main parts: 1) design a good quantizer α, 2) design a good lossless code (γ, β). We set $X_A, X_B \in \mathbb{R}^d, K \in \mathcal{K} \subseteq \mathbb{N}$. The functions involved are $\alpha : \mathbb{R}^d \to \mathcal{K}$, $\gamma : \mathcal{K} \to \{0,1\}^*$ and $\beta : \{0,1\}^* \times \mathbb{R}^d \to \mathcal{K}$. A diagram of the scheme is provided in Fig. 1. The operations α and γ are made on Alice's side, while the decoding β takes place on Bob's side.

We define the *correction rate* R as the average length of the correction message that Alice sends to Bob: $R = E[|\gamma(K)|]$. A lower bound to the correction rate is the lowest achievable rate for a lossless code with side information, which is known [10] to be equal to the conditional entropy of the message with respect to the side information: $R \geq H(K \mid X_B)$. The amount of information that is shared by Alice and Bob is therefore equal to the entropy $H(K)$ of the key generated by Alice, minus the number of correction bits R. An upper bound to this quantity is $H(K) - H(K \mid X_B) = I(K; X_B)$.

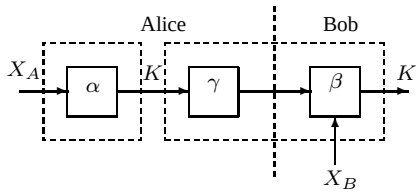

Figure 1: Block diagram of the proposed system

II. Quantization

We have seen that $I(K; X_B)$ is an upper bound on the amount of information shared by Alice and Bob. Actually, when

[1] A longer version is available as technical report [5]

the granularity of the quantizer α tends to infinity, we have $I(K;X_B) \to I(X_A;X_B)$, and $H(K \mid X_B) \to +\infty$. The first limit is well known, while the second comes from the fact that the discrete entropy of a continuous variable is infinite. Hence the price to pay to get $I(K;X_B)$ closer to the ultimate upper bound $I(X_A;X_B)$ is an increase in the average size $H(K \mid X_B)$ of the correction message assuming an ideal lossless coder. Our goal in designing the quantizer α is to maximize $I(K;X_B)$ while keeping $H(K \mid X_B)$ bounded. We show that such quantizers actually have the structure of vector quantizers for probability distributions with the Kullback-Leibler divergence as distortion measure. Traditional quantization aims at minimizing a distortion measure defined in the signal space, such as the mean squared error [6]. There is however already some literature on quantization for maximal mutual information. This idea has actually emerged recently in rather different contexts. In a recent contribution from Wu et al. [14], a maximal mutual information quantizer is utilized to classify context vectors in data compression applications. They use a minimal conditional entropy criterion which turns out to be strictly equivalent to maximizing the mutual information. They mention the Lloyd approach presented next. A similar maximum mutual information optimization is presented in the *information bottleneck method* from Tishby, Pereira and Bialek [12].

We propose a method that follows the developments provided in [14] and inspired from the Lloyd optimality conditions for vector quantizers. We first assume that K belongs to the set $\mathcal{K} = \{1, 2, \ldots, N\}$. Then clearly $H(K \mid X_B)$ is bounded by $\log N$. We use the notation $\langle f, g \rangle = \int f(x)g(x)dx$, and $h(.)$ for the differential entropy. Then α is a solution of

$$\begin{aligned}
&\arg\max_{\alpha} I(K;X_B) \\
&= \arg\max_{\alpha} h(X_B) - h(X_B \mid K) \\
&= \arg\min_{\alpha} h(X_B \mid K) \\
&= \arg\min_{\alpha} h(X_B \mid K) - h(X_B \mid X_A) \\
&= \arg\min_{\alpha} \langle P_{X_A}, \langle P_{X_B|X_A}, \log \frac{P_{X_B|X_A}}{P_{X_B|K}} \rangle \rangle \\
&= \arg\min_{\alpha} E_{X_A}[D(P_{X_B|X_A} \parallel P_{X_B|K})].
\end{aligned} \tag{1}$$

The function $D(p \parallel q)$ is called the *Kullback-Leibler (K-L) divergence* or the *relative entropy* of p with respect to q.

From the previous developments, we see that a realization x_A of the continuous value X_A on Alice's side should be mapped by α to the key $\alpha(x_A)$ such that

$$\alpha(x_A) = \arg\min_{k=1}^{N} D(P_{X_B|X_A=x_A} \parallel P_{X_B|K=k}), \tag{2}$$

that is, to the key k whose associated distribution $P_{X_B|K=k}$ is the nearest neighbor of $P_{X_B|X_A=x_A}$ in terms of the K-L divergence. This is equivalent to the first Lloyd's optimality condition in classical vector quantization. The nearest neighbor condition in Eqn. (2), however, is tail-biting: the mapping α is defined through the distributions $P_{X_B|K}$, which in turn depend on α. This observation suggests an algorithm in which the mapping and the conditional distributions are updated alternately. Let us define $\{f_k\}_{k=1}^N$ the *codebook* of probability distributions for X_B and the *quantization cells* $\mathcal{Q}_k = \{x_A \mid \alpha(x_A) = k\}$, i.e., the subsets of $\mathbb{R}^d$ whose elements are mapped to the same quantization index k. The quantizer α is completely defined by the partition $\{\mathcal{Q}_k\}_{k=1}^N$. The following algorithm is applied, starting with any initial quantizer α:

1. $\forall\, k = 1, 2, \ldots, N : f_k \leftarrow E[P_{X_B|X_A} \mid X_A \in \mathcal{Q}_k]$
2. $\forall\, k = 1, 2, \ldots, N : \mathcal{Q}_k \leftarrow \{x_A \mid \forall j \neq k\ D(P_{X_B|X_A=x_A} \parallel f_j) > D(P_{X_B|X_A=x_A} \parallel f_k)\}$
3. Repeat the previous steps until convergence.

While this algorithm is an adaptation of the well-known generalized Lloyd algorithm [6], we can consider that the agglomerative information bottleneck technique [11] is an adaptation of a family of algorithm for vector quantizer design known as Pairwise Nearest Neighbor algorithms

The previous description of the local optimization algorithm is rather general and not directly implementable. First, probability distributions are generally estimated up to a certain precision. Then, the design of the improved quantizer is not straightforward either. It can be carried out using a *training set* of realizations of X_A and applying the nearest neighbor rule (2) for each element of the set.

We now discuss some properties of these quantizers. For simplicity, we temporarily assume that X_B is a discrete random variable in the finite set $\mathcal{X}_B$. Quantization cells $\mathcal{Q}_k$ have no special structure. It is not necessary, in particular, that values of X_A that are close to each other lead to similar distributions for X_B. On the other hand, there exist quantization cells $\mathcal{C}_k$ on the probability simplex, the set of vectors of size $|\mathcal{X}_B|$ with positive components summing to one. These cells contain all probability mass functions for X_B corresponding to a given quantization index k: $\mathcal{C}_k = \{P_{X_B|X_A=x_A} \mid \alpha(x_A) = k\}$. These cells are connected and bounded by $(|\mathcal{X}_B| - 2)$-dimensional hyperplanes. It is not too difficult to show that the optimal value of f_k within a cell is the average probability mass function in that cell. In other words, vector quantizers minimizing the K-L divergence obey the centroid rule. This centroid rule is important because it proves that the alternate optimization algorithm converges: Each of the two steps decreases the K-L divergence, and since this quantity is always positive, the algorithm must converge to a quantizer that is locally optimal with respect to both the nearest neighbor and the centroid rule. Furthermore, we can show that the exact average K-L divergence D_k within the cell $\mathcal{C}_k$ is known as the generalized Jensen-Shannon divergence when $g(.)$ is actually a probability mass function.

Instead of fixing $K \in \mathcal{K} = \{1, 2, \ldots, N\}$, we can simply let $\mathcal{K} = \mathbb{N}$ and solve the constrained problem of maximizing $I(K;X_B)$ subject to $H(K \mid X_B) \leq R^*$ for a certain bound R^* on the ideal correction rate. Introducing a Lagrangian multiplier $\lambda \in \mathbb{R}^+$, we seek

$$\max_{\alpha} I(K;X_B) - \lambda H(K \mid X_B). \tag{3}$$

The Lagrangian multiplier λ controls the tradeoff between the fraction of the maximal mutual information $I(X_A; X_B)$ that is actually shared and the ideal correction rate $H(K \mid X_B)$ on the authenticated channel.

The assumption that the conditional distributions $P_{X_B|X_A}$ are Gaussian, or in some sense close to Gaussian might also be interesting in applications. A reasonable approximation of the K-L divergence $D(P_{X_B|X_A=x_A} \parallel f_k)$ that we wish to minimize can be obtained by modeling f_k by a Gaussian pdf $\tilde{f}_k$ with the same mean and variance. The error due to this approximation is $D(P \parallel \tilde{f}_k) + \langle P, \log \frac{\tilde{f}_k}{f_k} \rangle$, where the additional term $\langle P, \log \frac{\tilde{f}_k}{f_k} \rangle$, the "distance" between f_k and its approximation, averaged with respect to P, should be minimized. Let f_1 and f_2 be two Gaussian pdf with respective means μ_1 and μ_2 and standard deviations σ_1 and σ_2. It is straightforward to show that

$$D(f_1 \parallel f_2) = \ln \frac{\sigma_2}{\sigma_1} - \frac{1}{2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} \text{ nats.} \qquad (4)$$

III. Lossless Coding with Side Information

As the previous section described the design of the quantizer, let us now discuss the design of the lossless code with side information at the receiver. Alice wishes to send $\gamma(K)$ with a rate R as little as possible such that Bob is able to recover K with a high probability.

Symbols k and k' are said to be *confusable* if $\exists\ x_B$ such that $P_{K,X_B}(k, x_B) > 0 \wedge P_{K,X_B}(k', x_B) > 0$. If such k and k' are associated with the same codeword, the decoder β will not be able to tell which one is correct. For many interesting cases, such as joint Gaussian variables, the joint probability function P_{K,X_B} will in general always be strictly positive. All symbols are thus confusable. This means that a non-zero probability of error at the decoder side must be tolerated, allowing some confusable symbols to have identical codewords, otherwise making γ bijective. The probability of confusion is defined as

$$P_c = \Pr\left[\beta(\gamma(K), X_B) \neq K\right], \qquad (5)$$

which is thus to be minimized together with the rate R. The code γ can be either [1]: a *restricted inputs* (RI) code, where $\gamma(k)$ is not a prefix of $\gamma(k')$ whenever k and k' are confusable, or an *unrestricted inputs* (UI) code, where $\gamma(k) \neq \gamma(k')$ whenever k and k' are confusable and $\gamma(k)$ can never be a prefix of $\gamma(k')$ (even if k and k' are not confusable).

In general, the codes of consecutive inputs will be concatenated to make a binary stream. This means that, in addition to outputting an incorrect k, the decoder β may as well desynchronize if the code associated to a symbol k is a proper prefix of the code of a distinct confusable symbol k'. This problem should thus be circumvented by using an UI code, making the stream instantaneously decodable even without the side information. Confusion can still happen, but desynchronization cannot. We will now overview some constructions of code in previous research.

Zero-error codes are aimed at allowing the decoder to unambiguously determine the transmitted symbol without error. They make explicit use of zero entries in the joint probability distribution, and generally refer to the notion of *confusability graph*, an undirected graph whose nodes are symbols and in which edges connect all pairs of confusable symbols. Witsenhausen [13] relates zero-error codes to the chromatic number of the confusability graph. Further work along these lines is found in Koulgi et al. [8]. A construction called MASC [15] produces optimal RI codes. Clearly, our problem involves joint probabilities that have no zero entries. Zero-error corrections thus cannot be used as such. A possible modification is examined below.

Another way for Alice to give Bob information about $K = \alpha(X_A)$ is to send him the syndrome of a linear error correcting code $\gamma(K) = HK$, with K expressed in some vector space $GF(q)^n$ and H the parity check matrix of the code. Upon receiving $s_A = Hk$ for an outcome k of K, Bob looks for the most probable $\tilde{k}$ conditionally on $X_B = x_B$ such that $H\tilde{k} = s_A$. Standard decoding techniques can be used as soon as choosing the most probable symbol reduces to minimizing the Hamming distance between Bob's a priori (without HK) and a posteriori (with HK) guesses. This idea is implemented in the DISCUS framework [9]. However, the focus there is set on a rate-distortion version of the problem, in which lattice quantization and trellis-coded side-information are combined. Still, good syndromes may be of help in the scope of secret key construction, allowing fast decoding procedures.

Interactive protocols are often used for QKD purposes. Cascade [4] for instance is a binary interactive error correction (IEC) protocol. It works on a long binary string and requires Alice and Bob to exchange parities of subsets of their bits. When the parity of a subset differs, they know for sure that they have an odd number of wrong bits in this subset, hence at least one. They can perform a bisection and repeatedly exchange the parity of half the current subset until one bit is isolated and corrected (flipped). Cascade keeps track of all investigated subsets and takes advantage of this information: When an error is isolated and corrected, it updates the parity of all previously processed subsets to which the corrected bit belongs. This may then imply that the parity of some updated subset now differs between Alice and Bob, causing a new bisection to start, until the error is found and corrected. The interactivity of such IEC protocols has some drawbacks in the scope of QKD, as information leaks from both sides, an aspect detailed in [5]. It however offers overwhelmingly small probability of errors at the end of the protocol, making IEC fruitful when combined with a source code with side information, to further reduce the number of residual errors.

Given an encoder γ, the decoder that minimizes the probability of confusion simply returns the most probable symbol:

$$\beta(\phi, x_B) = \arg \max_{k \in \gamma^{-1}(\phi)} P_{K,X_B}(k, x_B), \qquad (6)$$

where $\phi \in \{0,1\}^*$ and $\gamma^{-1}(\phi) = \{k : \gamma(k) = \phi\}$. With such a decoder, the confusion probability is the probability mass that the decoder cannot reach,

$$P_c = 1 - \int dx_B \sum_{k\,:\,\exists \phi\ k = \beta(\phi, x_B)} P_{K,X_B}(k, x_B). \qquad (7)$$

Since UI codes allow only different prefix-free or equal codewords, we can w.l.o.g. define γ as the composition of an

index assignment (IA) function δ and of a bijective code assignment function γ_0: $\gamma = \gamma_0 \circ \delta$, with $\delta : \mathcal{K} \to \mathcal{K}$ and $\gamma_0 : \mathcal{K} \to \{0,1\}^*$. The IA function thus represents the partition of K into subsets with equal codes, such as a flat partition tree [15] or as a graph coloring [13]. The function γ_0 can be for instance Huffman or arithmetic coding. For a given IA function δ, the decoder (6) becomes $\beta^{(\delta)}(\gamma_0(i), x_B) = \arg\max_{k \in \delta^{-1}(i)} P_{K,X_B}(k, x_B)$, and by defining $P^{(\delta)}(i, x_B) = \max_{k \in \delta^{-1}(i)} P_{K,X_B}(k, x_B)$ (or 0 if $\delta^{-1}(i) = \emptyset$), we get $P_c^{(\delta)} = 1 - \int dx_B \sum_i P^{(\delta)}(i, x_B)$.

Note that the only relevant information extracted from X_B is the symbol k of highest conditional probability for each index i such that $\delta^{-1}(i) \neq \emptyset$. When δ is the identity, K is transmitted losslessly without taking the side information into account, making X_B irrelevant to the decoder. On the other hand, if δ is a constant, the decoder has no information on K except via X_B. Since there is only one set $\delta^{-1}(i) = \mathcal{K}$, the only relevant information extracted by the decoder is the symbol k of highest conditional probability for each x_B. More general cases lie between these two extreme cases. This enables us to quantize X_B in a way that does not alter the performance of the decoder. Instead of working with X_B as such, one can define the vector $\pi^{(\delta)}(x_B) = \big(\beta^{(\delta)}(\gamma_0(i), x_B)\big)_{i\,:\,\delta^{-1}(i)\neq\emptyset}$ and consider β as a function of the received codewords and of the quantized $\pi^{(\delta)}(X_B)$ without increasing P_c.

If δ is not known when quantizing X_B, a procedure that works for any choice of δ is to use the full relative order of the conditional probability of the k's. Hence, $\pi(x_B)$ maps to a permutation of $\mathcal{K}$. We can thus replace the random variable X_B by the discrete variable $\pi(X_B)$, an effect that results directly from the discrete nature of K. Note that this may not be efficient, as the size of the resulting alphabet may grow as $O(n!)$ if $|\mathcal{K}| \leq n$. However, this can be reduced in practice if one neglects the relative order of key symbols that have low conditional probabilities, or if one limits the density of the resulting cells in $\mathbb{R}^d$. In the sequel, $X_B \in \mathcal{X}_B$ will denote the quantized version, unless stated otherwise.

We now present a simple heuristic algorithm to design UI codes. The γ_0 function is assumed to be arithmetic coding, implying to minimize $R = H(\delta(K))$. We start with a bijective IA function $\delta(k) = k$, hence giving $P_c = 0$ and $R = H(K)$, and then merge some key symbols so as to reduce the rate of γ at the cost of an increase in P_c. Merging two indices $i_1, i_2 \in R(\delta)$ consists in creating a new IA function δ' identical to δ except that it now returns i_1 whenever i_2 was returned:

$$\delta'(k) = \begin{cases} i_1 & \text{if } k \in \delta^{-1}(i_2), \\ \delta(k) & \text{otherwise.} \end{cases} \tag{8}$$

We thus get $R(\delta') = R(\delta) \setminus \{i_2\}$ and $\delta'^{-1}(i_1) = \delta^{-1}(i_1) \cup \delta^{-1}(i_2)$, so that the key elements that were assigned to either index i_1 or i_2 are now assigned to the same codeword.

Upon merging i_1 and i_2, this gives $P^{(\delta')}(i_1, x_B) = \max\{P^{(\delta)}(i_1, x_B), P^{(\delta)}(i_2, x_B)\}$ and $P^{(\delta')}(i_2, x_B) = 0$. The increase in confusion probability is thus $\Delta P_c = \sum_{x_B} \min\{P^{(\delta)}(i_1, x_B), P^{(\delta)}(i_2, x_B)\}$, and the decrease in rate $\Delta R = f(P_{\delta(K)}(i_1), P_{\delta(K)}(i_2))$, with $f(p_1, p_2) = -(p_1 + p_2)\log(p_1 + p_2) + p_1 \log p_1 + p_2 \log p_2$. At each step, we choose the pair (i_1, i_2) such that the ratio $\lambda(i_1, i_2) = -\Delta R / \Delta P_c$ is maximized and merge i_1 and i_2, until no more merging is possible or if the maximum tolerated probability of confusion has been reached.

Although it does not necessarily give the optimal solution, this algorithm has the advantage of giving many possible codes with many associated (R, P_c) pairs in polynomial time.

IV. Conclusions

We presented a new secret key construction scheme and motivated it as a tool for some recent protocols of quantum key distribution. This problem was shown to divide into two other subproblems that are met in other contexts. First, we showed how to quantize a continuous secret key source in order to maximize an information-theoretic criterion. Then, we made a survey of existing codes with side information and listed the required features of such codes for the scope of our problem. We showed how unrestricted input codes can be used in this context and proposed a simple heuristic algorithm to construct such codes.

References

[1] N. Alon and A. Orlitsky. Source coding and graph entropies. *IEEE Trans. Inform. Theory*, 42(5):1329–1339, 1996.

[2] C. H. Bennett and G. Brassard. Public-key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, India*, pages 175–179, New York, 1984. IEEE.

[3] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer. Generalized privacy amplifi cation. *IEEE Trans. Inform. Theory*, 41(6):1915–1923, November 1995.

[4] G. Brassard and L. Salvail. Secret-key reconcialiation by public discussion. In T. Helleseth, editor, *Advances in Cryptology – Eurocrypt'93*, pages 411–423. Lecture Notes in Computer Science – Springer-Verlag, 1993.

[5] J. Cardinal and G. Van Assche. Construction of a shared secret key using continuous variables. Technical Report 497, Computer Science Dept., 2003. `http://www.ulb.ac.be/di/publications/`.

[6] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. Inform. Theory*, 44, 1998.

[7] F. Grosshans, G. Van Assche, J. Wenger, R. Brouri, N. J. Cerf, and P. Grangier. Quantum key distribution using gaussian-modulated coherent states. *Nature*, 421:238–241, 2003.

[8] P. Koulgi, E. Tuncel, S. Regunathan, and K. Rose. Minimum redundancy zerro-error source coding with side information. In *Proc. Int. Symposium on Information Theory*, June 2001.

[9] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. In *Proc. IEEE Data Compression Conf.*, pages 158–167, March 1999.

[10] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, 19:471–480, July 1973.

[11] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *Proc. of NIPS-12*, pages 617–623. MIT Press, 2000.

[12] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[13] H. S. Witsenhausen. The zero-error side information problem and chromatic numbers. *IEEE Trans. Inform. Theory*, 22(5):592–593, 1976.

[14] X. Wu, P. A. Chou, and X. Xue. Minimum conditional entropy context quantization. In *Proc. Int. Symposium on Information Theory*, 2000.

[15] Q. Zhao and M. Effros. Optimal code design for lossless and near lossless source coding in multiple access networks. In *Proc. IEEE Data Compression Conf.*, pages 263–272, 2001.