# Construction of a Shared Secret Key Using Continuous Variables (Full Text Report)

Jean Cardinal
Gilles Van Assche
Université Libre de Bruxelles
B-1050 Brussels, Belgium

*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 the mutual information $I(X_A;X_B)$ is nonzero. They wish to convert their values into a shared secret digital information while leaking as little information as possible to Eve. We show that the problem can be decomposed in two subproblems that are known in other contexts. The first is the design of a *quantizer* that maximizes a mutual information criterion, the second is known as *lossless coding with side information*.**

## I. Introduction

### A Key distribution with continuous quantum states

The work presented in this paper 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 two parties, usually called Alice and Bob, to share a secret key that can be used for encrypting messages using a classical cipher, e.g., the one-time pad [25]. The main interest of such a key distribution scheme is that eavesdropping is detectable, as the laws of quantum mechanics imply that measuring a quantum state generally disturbs it. Actually, 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, or vice-versa [13], on the quantum channel. These states carry some information that Bob will determine the best he can. 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. They measure some appropriate disturbance metric, from which they can 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 $S$ 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, about which we now give some details.

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 sends[1] Bob some *correction* information $f(X_A)$ over the public authenticated channel in such a way that he can recover $X_A$ knowing $X_B$. Since $f(X_A)$ 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 [4, 19, 3], where the tapped information is wiped out at the cost of a reduction in the key length.

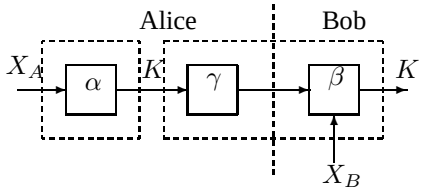


Figure 1: Block diagram of the proposed system

Privacy amplification (PA) is not covered in this paper, since the currently developed protocols can readily be used. It is however relevant to our problem, as the reduction in key length during PA is roughly equal to the number of bits known to an eavesdropper [3, 20], both from tapping the quantum channel and from listening to the public channel. It should thus now appear clearly that the reconciliation information $f(X_A)$ should not give more information than necessary on $X_A$, otherwise resulting in a penalty in the key length. Ideally, only $H(X_A|X_B)$ bits should be given, resulting in a secret key length of $H(X_A) - H(X_A|X_B) = I(X_A;X_B)$ bits, assuming an untapped perfect quantum channel.

Unlike binary QKD protocols, some recent protocols [8, 12, 13] 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.

### B Proposed Scheme

We propose a three-phase approach (excluding privacy amplification) to the problem of constructing a shared secret key

[1] Actually, this may involve an interactive correction rather than a one-way communication, but this aspect will be detailed later in Sec. III.

$K$. In a first phase, Alice maps her value $X_A$ to an integer $K = \alpha(X_A)$ using a predefined function $\alpha$. 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. In the source coding terminology, $\alpha$ is a *quantizer* [11], and the pair $(\gamma, \beta)$ is a *lossless code with side information at the receiver* [27, 31]. We have therefore split our problem in two main parts: 1) design a good quantizer $\alpha$, 2) design a good lossless code $(\gamma, \beta)$.

We set $X_A, X_B \in \mathbb{R}^d, K \in \mathcal{K} \subseteq \mathbb{N}$. The functions involved are

$$\alpha : \mathbb{R}^d \to \mathcal{K} \quad (1)$$
$$\gamma : \mathcal{K} \to \{0,1\}^* \quad (2)$$
$$\beta : \{0,1\}^* \times \mathbb{R}^d \to \mathcal{K}. \quad (3)$$

A summary of the scheme is provided in Fig. 1. The operations $\alpha$ and $\gamma$ are made on Alice's side, while the decoding $\beta$ 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:

$$R = E[|\gamma(K)|]. \quad (4)$$

A lower bound to the correction rate is the lowest achievable rate for a lossless code with side information, which is known [27] to be equal to the conditional entropy of the message with respect to the side information:

$$R \geq H(K \mid X_B). \quad (5)$$

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, to which we subtract the number of correction bits $R$. From Eqn. (5), an upper bound to this quantity is $H(K) - H(K \mid X_B) = I(K; X_B)$. We call $H(K \mid X_B)$ the *ideal correction rate*.

## 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 the granularity of the quantizer $\alpha$ tends to infinity, we have

$$I(K; X_B) \to I(X_A; X_B) \quad (6)$$
$$H(K \mid X_B) \to +\infty. \quad (7)$$

The first limit is well known [9], 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 $\alpha$ is to maximize $I(K; X_B)$ while keeping $H(K \mid X_B)$ bounded.

We first mention several previous studies related to this problem. In Sec. B, we propose a general algorithm. Next, we discuss extensions and practical implementation of the general algorithm.

# A Previous works

Traditional quantization aims at minimizing a distortion measure defined in the signal space, such as the mean squared error [11]. 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. [32], a maximal mutual information quantizer is utilized to classify context vectors in data compression applications. They wish to predict the distribution of a variable $X$ given a large context $Y$. They describe how to map $Y$ optimally onto the set $\{1, 2, \ldots, N\}$ with a quantizer $\alpha$ so that the conditional entropy $H(X \mid \alpha(Y))$ is minimal. This is strictly equivalent to maximizing the mutual information $I(\alpha(Y); X)$. They exhibit an exact polynomial algorithm for the binary case $X \in \{0, 1\}$ and mention the Lloyd approach presented next. In the so-called *information bottleneck method* from Tishby, Pereira and Bialek [29], a stochastic map $p(k \mid x_A)$ plays the role of the quantizer $\alpha$, and maximizes $I(K; X_B)$ subject to a constraint on the value of $I(K; X_A)$. Tishby et al. describe an algorithm for computing these maps based on classical developments in rate-distortion theory and similar to the Blahut-Arimoto algorithm [5]. In a subsequent development of the method [28], they describe a heuristic agglomerative algorithm for designing "hard clusters", i.e., a deterministic quantizer, optimizing the same criteria. Note that in that case the constraint on $I(K; X_A)$ reduces to a constraint on $H(K)$, since $H(K \mid X_A) = 0$. The authors argue that this method provides a practical solution to numerous problems in prediction, neural coding and signal processing. Let us also notice several recent contributions in neural information processing based on the same ideas, such as [26].

# B A general algorithm

We propose a method that follows the developments provided in [32] and inspired from the Lloyd optimality conditions for vector quantizers [11]. 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 $\alpha$ is a solution of

$$\begin{aligned}
&\arg\max_\alpha I(K; X_B) \\
&= \arg\max_\alpha h(X_B) - h(X_B \mid K) \\
&= \arg\min_\alpha h(X_B \mid K) \\
&= \arg\min_\alpha h(X_B \mid K) - h(X_B \mid X_A) \\
&= \arg\min_\alpha -\langle P_{X_A}, \langle P_{X_B|X_A}, \log P_{X_B|K} \rangle \rangle \\
&\quad + \langle P_{X_A}, \langle P_{X_B|X_A}, \log P_{X_B|X_A} \rangle \rangle \\
&= \arg\min_\alpha \langle P_{X_A}, \langle P_{X_B|X_A}, \log \frac{P_{X_B|X_A}}{P_{X_B|K}} \rangle \rangle \\
&= \arg\min_\alpha E_{X_A}[D(P_{X_B|X_A} \parallel P_{X_B|K})].
\end{aligned} \quad (8)$$

The function $D(p \parallel q)$ is called the *Kullback-Leibler divergence* or the *relative entropy* of $p$ with respect to $q$ [9].

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 $\alpha$ to the key $\alpha(x_A)$ such that

$$\alpha(x_A) = \arg\min_{k=1}^{N} D(P_{X_B|X_A=x_A} \parallel P_{X_B|K=k}), \quad (9)$$

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. (9), however, is tail-biting: the mapping $\alpha$ is defined through the distributions $P_{X_B|K}$, which in turn depend on $\alpha$. 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 $\alpha$ is completely defined by the partition $\{\mathcal{Q}_k\}_{k=1}^N$. Algorithm 1 is applied, starting with any initial quantizer $\alpha$. A suitable tie-breaking rule is used in the update step for $\mathcal{Q}_k$.

**Algorithm 1** A general alternate optimization algorithm

**repeat**
  **for** $k = 1, 2, \ldots, N$ **do**
    $f_k \leftarrow E[P_{X_B|X_A} \mid X_A \in \mathcal{Q}_k]$
  **end for**
  **for** $k = 1, 2, \ldots, N$ **do**
    $\mathcal{Q}_k \leftarrow \{x_A \mid \forall j \neq k \; D(P_{X_B|X_A=x_A} \parallel f_j) > D(P_{X_B|X_A=x_A} \parallel f_k)\}$
  **end for**
**until** variation of $E_{X_A}[D(P_{X_B|X_A} \parallel P_{X_B|K})]$ becomes negligible

While this algorithm is an adaptation of the well-known generalized Lloyd algorithm, we can consider that the agglomerative information bottleneck technique [28] is an adaptation of the Pairwise Nearest Neighbor algorithm [10] for vector quantizer design.

## C Practical algorithm

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 $\mathcal{T}$ of instances of $X_A$ and applying the nearest neighbor rule (9) for each element of the set. The algorithm becomes as described in Algorithm 2.

One has to take care of the following points. First, for evaluating the K-L divergence, it is probably simpler to quantize $X_B$ as well (e.g., with a high resolution quantizer or as suggested in Sec. D), so that $f_k$ becomes a vector and the integral reduces to a discrete sum. Then, the estimations of $P_{X_B|X_A=x_A}$ and $P_{X_B|K=k}$ can be either numerical or analytical, depending on the knowledge we have about the joint behavior of $X_A$ and $X_B$. It may occur, in particular, that only empirical data is available, for instance in the form of joint training sets. This can lead to serious precision problems when evaluating the K-L divergence. A good solution in that case might be to model the distribution using simple assumptions.

**Algorithm 2** A practical alternate optimization algorithm

**repeat**
  **for** $k = 1, 2, \ldots, N$ **do**
    $\mathcal{T}_k \leftarrow \{x_A \in \mathcal{T} \mid \alpha(x_A) = k\}$
    $f_k \leftarrow (\sum_{x_A \in \mathcal{T}_k} P_{X_B|X_A=x_A})/|\mathcal{T}_k|$
  **end for**
  **for** each $x_A \in \mathcal{T}$ **do**
    $\alpha(x_A) \leftarrow \arg\min_{k=1}^{N} D(P_{X_B|X_A=x_A} \parallel f_k)$
  **end for**
**until** variation of $\sum_{x_A \in \mathcal{T}} D(P_{X_B|X_A=x_A} \parallel f_{\alpha(x_A)})/|\mathcal{T}|$ becomes negligible

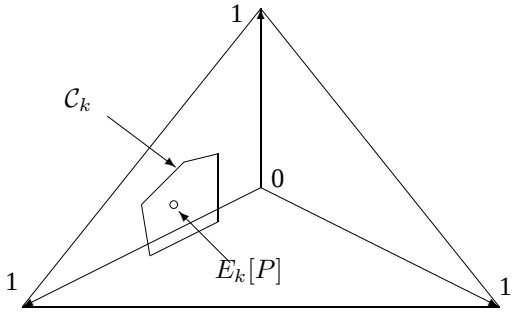


Figure 2: A cell $\mathcal{C}_k$ on the probability simplex for $|\mathcal{X}_B| = 3$

## D Properties

For simplicity, we temporarily assume in this subsection 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 $k$: $\mathcal{C}_k = \{P_{X_B|X_A=x_A} \mid \alpha(x_A) = k\}$. An illustration is given on Fig. 2. These cells are connected and bounded by $(|\mathcal{X}_B| - 2)$-dimensional hyperplanes. We first 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. In the following, $g(.)$ is the probability density function of the distribution $P = (P_1, P_2, \ldots, P_{|\mathcal{X}_B|})$ of $X_B$ within the cell $\mathcal{C}_k$ and the expectation $E_k[.]$ denotes the expected value within that same cell. Hence $E_k[P]$ is the $|\mathcal{X}_B|$-dimensional vector $(\int_{\mathcal{C}_k} P_1 g(P)dP, \int_{\mathcal{C}_k} P_2 g(P)dP, \ldots, \int_{\mathcal{C}_k} P_{|\mathcal{X}_B|} g(P)dP)$. Note that if $X_A$ is a discrete random variable, the probability density function $g(.)$ is actually a probability mass function. The following discussion is without loss of generality.

We wish to show that $f_k = E_k[P]$ is the solution of

$$\min \int_{\mathcal{C}_k} \langle P, \log \frac{P}{f_k} \rangle g(P)dP \quad (10)$$

subject to $\langle 1, f_k \rangle = 1$. This is easily achieved by writing the Lagrangian cost

$$J = \lambda \langle 1, f_k \rangle + \int_{\mathcal{C}_k} \langle P, \log \frac{P}{f_k} \rangle g(P) dP. \tag{11}$$

Taking the derivative for each component $j$ leads to

$$\frac{\delta J}{\delta f_{k,j}} = \lambda - \frac{1}{f_{k,j}} \int_{\mathcal{C}_k} P_j g(P) dP = 0 \tag{12}$$

by identification, we find $\lambda = 1$ and $f_{k,j} = \int_{\mathcal{C}_k} P_j g(P) dP = E_k[P_j]$, hence $f_k = E_k[P]$.

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.

Let us now compute the exact average K-L divergence $D_k$ within the cell $\mathcal{C}_k$:

$$\begin{aligned} D_k &= \int_{\mathcal{C}_k} \langle P, \log \frac{P}{f_k} \rangle g(P) dP \\ &= \int_{\mathcal{C}_k} \left( \langle P, \log P \rangle - \langle P, \log f_k \rangle \right) g(P) dP \\ &= -E_k[H(P)] - \langle E_k[P], \log f_k \rangle \end{aligned} \tag{13}$$

but since $f_k = E_k[P]$ we obtain

$$D_k = H(E_k[P]) - E_k[H(P)]. \tag{14}$$

When $g(.)$ is actually a probability mass function, this expression is known as the generalized Jensen-Shannon divergence [17]. We conclude that minimization of the average K-L divergence or the average Jensen-Shannon divergence within a cluster are similar problems.

## E Quantization with ideal correction rate constraint

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

$$\max_{\alpha} I(K; X_B) - \lambda H(K \mid X_B). \tag{15}$$

Applying previous developments in Eqn. (8), this problem reduces to

$$\begin{aligned} \min_{\alpha} E_{X_A}[D(P_{X_B|X_A} \parallel P_{X_B|K})] \\ + \lambda H(K \mid X_B), \end{aligned} \tag{16}$$

from which we derive the following *modified nearest neighbor rule* for $\alpha$: for any $x_A$, $\alpha(x_A)$ is such that

$$\begin{aligned} \alpha(x_A) = \arg \min_{k=1}^{N} \big( & D(P_{X_B|X_A=x_A} \parallel P_{X_B|K=k}) \\ & - \lambda E[\log P[K = k \mid X_B] \mid X_A = x_A] \big). \end{aligned}$$

The Lagrangian multiplier $\lambda$ 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. When $\lambda$ tends to 0, the granularity of $\alpha$ becomes infinitely high and the behavior of $I(K; X_B)$ and $H(K \mid X_B)$ are as in Eqn. (6–7). When $\lambda \to +\infty$, both values tend to zero. Each intermediate value corresponds to a constraint $R^*$ on the ideal correction rate.

We can plug these expressions in the previous algorithm and derive a modified method that finds a locally optimal quantizer satisfying the correction rate constraint. This modified criterion requires a bigger computational effort than the previous one, for we have to evaluate not only $P_{X_B|X_A=x_A}$ for each $x_A$ but also $E[\log P[K = k \mid X_B] \mid X_A = x_A]$ for each $x_A$ and each $k$.

## F Trivial case $X_B = X_A$

A trivial case arises when Alice and Bob share identical continuous values. The problem degenerates as follows:

$$\begin{aligned} & \arg\max_{\alpha} I(K; X_B) \\ = & \arg\max_{\alpha} H(K) - H(K \mid X_B) \\ = & \arg\max_{\alpha} H(K) - H(K \mid X_A) \\ = & \arg\max_{\alpha} H(K), \end{aligned} \tag{17}$$

which amounts to finding any quantizer with maximal entropy, i.e., whose cells are equiprobable. Note that this is a property of minimum mean squared error quantizers [11].

If we only bound the correction rate, the shared information can be made arbitrarily close to the upper bound $I(X_A; X_B)$, since the correction rate is always zero.

## G Gaussian modeling

Let us assume that the conditional distributions $P_{X_B|X_A}$ are Gaussian, or in some sense close to Gaussian. This might be a useful assumption in the quantum cryptography application. Then 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 can be computed as follows:

$$\begin{aligned} D(P \parallel f_k) = & \langle P, \log \frac{P}{f_k} \rangle \\ = & \langle P, \log \frac{P}{\tilde{f}_k} \frac{\tilde{f}_k}{f_k} \rangle \\ = & D(P \parallel \tilde{f}_k) + \langle P, \log \frac{\tilde{f}_k}{f_k} \rangle. \end{aligned} \tag{18}$$

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. It can be computed to give an indication of the quality of the approximation.

Let $f_1$ and $f_2$ be two Gaussian pdf with respective means $\mu_1$ and $\mu_2$ and standard deviations $\sigma_1$ and $\sigma_2$. It is straightforward to show that

$$D(f_1 \parallel f_2) = \ln \frac{\sigma_2}{\sigma_1} - \frac{1}{2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} \text{ nats.} \quad (19)$$

If $X_B|X_A$ is multivariate Gaussian, a similar approximation of the K-L divergence can be obtained by modeling $f_k$ as a multivariate Gaussian with estimated covariance matrices. Simple formulas for the K-L divergence are still applicable. Even finer approximations have been recently proposed by Lin, Saito and Levine in [18], using higher order statistics. All these ideas straightforwardly apply to our method. Properties of the Voronoi diagram induced by the K-L divergence in a Gaussian parametric space are described in [21].

## 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.

# A Definitions and choices

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 $\beta$ 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 symbols to have identical codewords, even if confusable. This otherwise makes $\gamma$ bijective, in which case the rate $R \geq H(K)$ is of course unacceptably high, and which in the particular case of QKD completely discloses the key.

The probability of confusion is defined as

$$P_c = \Pr\left[\beta(\gamma(K), X_B) \neq K\right], \quad (20)$$

which is thus to be minimized together with the rate $R$, defined in Eqn. (4).

The code $\gamma$ 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 $\beta$ 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.

Zhao and Effros [34] use *partition trees* to capture the requirements on which key elements can have equal and/or prefix codewords. In our case, using UI implies that the partition tree is flat.

Also note that the partition tree must be converted into a binary stream by using either Huffman or arithmetic coding [34]. In the latter case, the definition of the code $\gamma$ should be understood for a block of key elements $k_{1\ldots m}$, and the correction rate to minimize is $H(\gamma(K))$.

# B Previous works

We will now overview some constructions of code in previous research. They will be briefly discussed in the light of our goal, which is to design an UI code with discrete side information with minimum rate under a constraint on the probability of confusion. Note that a discrete side information is usually assumed, unlike $X_B$ in our case, but this aspect will be covered in Sec. D, allowing the reader to assume a discrete $X_B$ in the current section.

We divide explicit constructions of codes in three main families. First there are constructions for zero-error codes [31, 33, 14, 15, 16, 34], some of which are based on graph coloring, and their near-zero-error variants [34]. Then, some are based on sending a syndrome of an error-correcting code [22, 23, 24, 30]. Finally, in the context of QKD, a mention of interactive codes [6, 7] will be made.

## B.1 Zero-error codes

Zero-error codes are aimed at allowing the decoder to unambiguously determine the transmitted symbol without any error, with the help of the side information, while minimizing the rate of the transmission. These constructions make explicit use of zero entries in the joint probability distribution.

With $K \in \mathcal{K} \subseteq \mathbb{N}$ the set of key elements and $X_B \in \mathcal{X}_B$ the side information known at Bob's side, let us define:

- the *characteristic bipartite graph* $G = (\mathcal{K} \cup \mathcal{X}_B, E)$, where for $k \in \mathcal{K}$ and $x_B \in \mathcal{X}_B$, $\{k, x_B\} \in E$ iff $P_{K,X_B}(k, x_B) > 0$;
- the *confusability graph* $G_K = (\mathcal{K}, E)$, where $\{k, k'\} \in E$ iff $k$ and $k'$ are confusable.

While Witsenhausen [31] relate zero-error codes to the chromatic number of the confusability graph, Alon et al. [1] show that the best coloring does not necessarily imply the best rate. Koulgi et al. [15] show an exponential-time optimal design algorithm for UI and RI codes based on confusability graphs and mention a fast approximation algorithm to design UI codes. A construction called MASC [34] produce optimal RI codes, generalizing either Huffman-type or arithmetic-type codes.

Further properties can be found in [33] where necessary and/or sufficient conditions on codeword lengths for small side

information alphabet sizes are provided and in [16] on theoretical properties of the achievable rate region using characteristic bipartite graphs.

Clearly, our problem involves joint probabilities that have no zero entries. Zero-error corrections thus cannot be used as such and a possible modification is examined next.

### B.2 Near-zero-error codes

Zero-error code constructions can be used to design near-zero-error codes. This may be done by applying a zero-error construction on a modified joint probability distribution, where small entries are set to zero and the remaining entries are renormalized.

This is done explicitly in [34]. First all the subsets of entries in the joint probability distribution that satisfy the given constrain on probability of error are listed. Then, for each of these subsets, a lossless MASC is designed with the modified joint probability distribution as input. Finally, the encoder with the minimum rate is selected.

Although this results in an optimal code for the required maximum probability of confusion, such a construction is not practical. Heuristics may be used to speed up the search, at the cost of an increase in $R$ and/or $P_c$.

### B.3 Syndrome-based codes

A 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. Suppose that Bob chooses an a priori decoding function $\beta_0(x_B) \in GF(q)^n$ such that $P_{K,X_B}(k, x_B)$ decreases as a function of $k$ iff $d_H(k, \beta_0(x_B))$ increases. Call Bob's syndrome $s_B = H\beta_0(x_B)$. Since $H(k - \beta_0(x_B)) = s_A - s_B$, we look for the coset leader $a$ of the syndrome $s_A - s_B$, which we add to $\beta_0(x_B)$, giving $\beta(s_A, x_B) = \beta_0(x_B) + a$ so that $H(k - \beta(s_A, x_B)) = 0$ and the probability that $\beta(s_A, x_B) = k$ is maximized.

This idea is implemented in the DISCUS framework [22, 23, 24]. However, the focus there is set on a rate-distortion version of the problem, in which (lattice) quantization and (trellis-based) side-information are combined. Still, good syndromes may be of help in the scope of secret key construction, allowing fast decoding procedures.

A mention should also be made to the rather theoretical construction in [30], where a universal two-step deterministic encoding is used. First, the input is divided into blocks and each block is encoded using a different linear code. Then, a syndrome of a class of linear error correcting code is appended. Provided that the rate of the code is inside the achievable region, the probability of error tends exponentially to zero as the block length goes to infinity.

### B.4 Interactive error correction

Interactive protocols are often used for QKD purposes. Cascade [6] 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, and so on.

The protocol Winnow [7] is another binary IEC protocol and works in a similar way. When a parity differs, however, Alice and Bob exchange the syndrome of a Hamming code instead of performing a bisection (notice the similarity with Sec. B.3). It also includes a privacy amplification step interleaved with the error correction, but this aspect will not be discussed further here.

Let us briefly analyze the cost of IEC. Let $A, B \in GF(2)^n$ be respectively Alice's and Bob's binary string of size $n$. After running Cascade, Alice and Bob disclosed $RA$ and $RB$ for some matrix $R$ of size $l \times n$. They thus communicated the parities calculated over identical subsets of bit positions. The matrix $R$ and the number $l$ of disclosed parities are not known beforehand but are the result of the interactive protocol and of the number and positions of the diverging parities encountered.

In the context of QKD, it is essential to minimize of the number $l$ of disclosed parities. For Cascade, $l \approx n(1+\xi)h(\epsilon)$, where $\epsilon = \Pr[A_i \neq B_i]$ is the bit error rate, $h(\epsilon) = -\epsilon \log \epsilon - (1-\epsilon)\log(1-\epsilon)$ and $\xi$ is some small overhead factor $\xi \ll 1$.

Assuming that Alice's bits $A$ will be used as a key, let us evaluate the amount of information on $A$ that is disclosed to an eavesdropper through $RA$ and $RB$. Let us also assume that Alice's and Bob's bits are balanced and are connected by a symmetric probability of error: $P[A_i = 0] = P[A_i = 1] = P[B_i = 0] = P[B_i = 1] = \frac{1}{2}$, $P[A_i = 0 \wedge B_i = 1] = P[A_i = 1 \wedge B_i = 0] = \epsilon$. In this case, the parities $RA$ give Eve $l$ bits of information on $A$, but $RB$ does not give any extra information since it is merely a noisy version of $RA$. Stated otherwise, $A \rightarrow RA \rightarrow RB$ is a Markov chain, hence $I(RA, RB; A) - I(RA; A) = I(RB; A|RA) = 0$ so that $I(RA, RB; A) \leq l \approx n(1+\xi)h(\epsilon)$ is not far away from the ideal $nh(\epsilon)$.

However, in the more general case where Eve gathered in $E$ some information on $A$ and $B$ by some other means (i.e., on the quantum channel in the scope of QKD), $A|E \rightarrow RA|E \rightarrow RB|E$ does not necessarily form a Markov chain. Instead, $I(RA, RB; A|E)$ must be explicitly evaluated or if this is not possible, it must be upper bounded by the number of bits disclosed by both parties as if they were independent, $I(RA, RB; A|E) \leq 2l \approx 2n(1+\xi)h(\epsilon)$, making the rate less

attractive than in the previous paragraph.

The interactivity offers overwhelmingly small probability of errors at the end of the protocol. It allows the communicating parties to spot errors without spending too many resources elsewhere. Even after an IEC was run, an interactive check procedure can be conducted, which for instance requires Alice and Bob to exchange the parity of random subsets of their bits. Conditionally on the fact that these tests succeeded, the residual probability of errors decreases exponentially with the number of such exchanged parities.

The low residual probability of errors can be exploited by considering an IEC as a complement to a non-interactive code. First, a non-interactive code gives $K$ to Bob with a small (but not small enough) probability of error. Then, an IEC is used to further reduce the probability of error down to a satisfactory limit.

## C Working with UI codes

Given an encoder $\gamma$, the decoder that minimizes the probability of confusion simply returns the most probable symbol conditionally on the side information, along with some suitable tie-breaking rule:

$$\beta(\phi, x_B) = \arg \max_{k \in \gamma^{-1}(\phi)} P_{K,X_B}(k, x_B), \tag{21}$$

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

$$P_c = 1 - \int dx_B \sum_{k\,:\,\exists \phi\ k=\beta(\phi,x_B)} P_{K,X_B}(k, x_B). \tag{22}$$

Since UI codes allow only different prefix-free or equal codewords, we can without loss of generality define $\gamma$ as the composition of an index assignment (IA) function $\delta$ and of a bijective code assignment function $\gamma_0$:

$$\delta : \mathcal{K} \to \mathcal{K}, \tag{23}$$

$$\gamma_0 : \mathcal{K} \to \{0,1\}^*, \tag{24}$$

$$\gamma = \gamma_0 \circ \delta. \tag{25}$$

The IA function thus represents the partition of $K$ into subsets with equal codes, such as a flat partition tree [34] or as a graph coloring [31]. The function $\gamma_0$ can be for instance Huffman or arithmetic coding.

For a given IA function $\delta$, the decoder (21) becomes

$$\beta^{(\delta)}(\gamma_0(i), x_B) = \arg \max_{k \in \delta^{-1}(i)} P_{K,X_B}(k, x_B), \tag{26}$$

and by defining

$$P^{(\delta)}(i, x_B) = \begin{cases} 0 \quad \text{if } \delta^{-1}(i) = \emptyset, \\ \max\limits_{k \in \delta^{-1}(i)} P_{K,X_B}(k, x_B) \text{ otherwise,} \end{cases} \tag{27}$$

we get

$$P_c^{(\delta)} = 1 - \int dx_B \sum_i P^{(\delta)}(i, x_B). \tag{28}$$

## D Quantization of $X_B$

We notice from Eqn. (26) 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 $\delta$ 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 $\delta$ 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

$$\pi^{(\delta)}(x_B) = \left(\beta^{(\delta)}(\gamma_0(i), x_B)\right)_{i\,:\,\delta^{-1}(i)\neq\emptyset}, \tag{29}$$

and consider $\beta$ as a function of the received codewords and of the quantized $\pi^{(\delta)}(X_B)$ without increasing $P_c$.

If $\delta$ is not known when quantizing $X_B$, a procedure that works for any choice of $\delta$ is to use the full relative order of the conditional probability of the $k$'s. Define $\pi(x_B) = (\pi_1, \pi_2, \ldots, \pi_{|\mathcal{K}|})$ with $\pi_l \in \mathcal{K}$ and

$$\begin{aligned}(&P_{K,X_B}(\pi_l, x_B) > P_{K,X_B}(\pi_m, x_B) \\ &\quad \vee P_{K,X_B}(\pi_l, x_B) = P_{K,X_B}(\pi_m, x_B) \wedge \pi_l < \pi_m) \\ &\qquad\qquad \Leftrightarrow l < m.\end{aligned} \tag{30}$$

We can thus replace the random variable $X_B$ by a 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.

## E A simple agglomerative algorithm

We now present a simple heuristic algorithm to design UI codes. The $\gamma_0$ 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 $\gamma$ at the cost of an increase in $P_c$. Merging two indexes $i_1, i_2 \in R(\delta)$ consists in creating a new IA function $\delta'$ identical to $\delta$ except that it now returns $i_1$ whenever $i_2$ was returned:

$$\delta'(k) = \begin{cases} i_1 & \text{if } k \in \delta^{-1}(i_2), \\ \delta(k) & \text{otherwise.} \end{cases} \tag{31}$$

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 $i_2$, we have from Eqn. (27) $P^{(\delta')}(i_1, x_B) = \max\{P^{(\delta)}(i_1, x_B), P^{(\delta)}(i_2, x_B)\}$ and $P^{(\delta')}(i_2, x_B) = 0$. The increase in probability of confusion is thus

$$\Delta P_c = \sum_{x_B} \min\{P^{(\delta)}(i_1, x_B), P^{(\delta)}(i_2, x_B)\}, \qquad (32)$$

and the decrease in rate

$$\Delta R = f(P_{\delta(K)}(i_1), P_{\delta(K)}(i_2)), \qquad (33)$$

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.

A table in memory can hold $P^{(\delta)}$, which is updated after each merge. No more than $|\mathcal{K}| - 1$ merges can occur. The evaluation of $\lambda$ takes $|\mathcal{X}_B|$ additions for each of the $O(|\mathcal{K}|^2)$ index pairs, and the update of $P^{(\delta)}$ takes $|\mathcal{X}_B|$ arithmetic operations. Thus the algorithm takes $O(|\mathcal{K}|^3|\mathcal{X}_B|)$.

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 an essential tool for some recent protocols of quantum key distribution. This problem was shown to divide into two other subproblems that are used 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.

We believe that showing this problem to the information theory community may raise interest and that it will lead to new problems and methods. In particular, both parts of our key construction scheme need further separate developments.

## V. *

References


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

[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, New York, 1984, IEEE, pp. 175–179.

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

[4] C. H. Bennett, G. Brassard, and J.-M. Robert, *Privacy amplifi cation by public discussion*, SIAM Journal on Computing, 17 (1988), pp. 210–229.

[5] R. E. Blahut, *Computation of channel capacity and rate-distortion functions*, IEEE Trans. Inform. Theory, 18 (1972), pp. 460–473.

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

[7] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, and C. G. Peterson, *Fast, effi cient error reconciliation for quantum cryptography*. arXiv e-print `quant-ph/0203096`, 2002.

[8] N. J. Cerf, M. Lévy, and G. Van Assche, *Quantum distribution of Gaussian keys using squeezed states*, Phys. Rev. A, 63 (2001), p. 052311.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, John Wiley & Sons, New York, NY, USA, 1991.

[10] W. H. Equitz, *A new vector quantization clustering algorithm*, IEEE Trans. Acoust., Speech, Signal Processing, 37 (1989), pp. 1568–1575.

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

[12] F. Grosshans and P. Grangier, *Continuous variable quantum cryptography using coherent states*, Phys. Rev. Lett., 88 (2002), p. 057902.

[13] 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 (2003), pp. 238–241.

[14] P. Koulgi, E. Tuncel, S. Regunathan, and K. Rose, *Graph-entropic characterization of optimal zero-error coding with side information*, in Seventh Canadian Workshop on Information Theory, June 2001.

[15] ———, *Minimum redundancy zerro-error source coding with side information*, in Proc. Int. Symposium on Information Theory, June 2001.

[16] P. Koulgi, E. Tuncel, and K. Rose, *On zero-error coding of correlated sources*, in Proc. Int. Symposium on Information Theory, July 2002, p. 62.

[17] J. Lin, *Divergence measures based on the shannon entropy*, IEEE Trans. Inform. Theory, 37 (1991), pp. 145–151.

[18] J.-J. Lin, N. Saito, and R. A. Levine, *Edgeworth approximations of the Kullback-Leibler distance towards problems in image analysis*. submitted for publication, 2001.

[19] U. M. Maurer, *Secret key agreement by public discussion from common information*, IEEE Trans. Inform. Theory, 39 (1993), pp. 733–742.

[20] U. M. Maurer and S. Wolf, *Information-theoretic key agreement: From weak to strong secrecy for free*, in Advances in Cryptology – Eurocrypt 2000, B. Preneel, ed., Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 351–368.

[21] K. Onishi and H. Imai, *Voronoi diagram in statistical parametric space by Kullback-Leibler divergence*, in Proceedings of the 13th ACM Symposium on Computational Geometry, 1997, pp. 463–465.

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

[23] ———, *Distributed source coding: Symmetric rates and applications to sensor networks*, in Proc. IEEE Data Compression Conf., Mar. 2000, pp. 363–372.

[24] ———, *Group-theoretic construction and analysis of generalized coset codes for symmetric/asymmetric distributed source coding*, in Proc. Conf. Information Science and Systems (CISS), March 2000.

[25] C. E. Shannon, *Communication theory of secrecy systems*, Bell Syst. Tech. J., 28 (1949), pp. 656–715.

[26] J. Sinkkonen and S. Kaski, *Clustering based on conditional distributions in an auxiliary space*, Neural Computation, 14 (2002), pp. 217–239.

[27] D. Slepian and J. K. Wolf, *Noiseless coding of correlated information sources*, IEEE Trans. Inform. Theory, 19 (1973), pp. 471–480.

[28] N. Slonim and N. Tishby, *Agglomerative information bottleneck*, in Proc. of NIPS-12, MIT Press, 2000, pp. 617–623.

[29] 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, 1999, pp. 368–377.

[30] T. UYEMATSU, *An algebraic construction of codes for Slepian-Wolf source networks*, IEEE Trans. Inform. Theory, 47 (2001), pp. 3082–3088.

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

[32] X. WU, P. A. CHOU, AND X. XUE, *Minimum conditional entropy context quantization*, in Proc. Int. Symposium on Information Theory, 2000.

[33] Y. YAN AND T. BERGER, *On instantaneous codes for zero-error coding of two correlated sources*, in Proc. Int. Symposium on Information Theory, June 2000.

[34] 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., 2001, pp. 263–272.