shown that the product of redundancy and square root of LFSW of maxentropic dc-free sequences is approximately constant when the bound of $RDS^{(1)}$ values is large.

ACKNOWLEDGMENT

The authors wish to thank Dr. K. A. S. Immink for helpful discussion regarding the reproduction of power spectra of previously published dc^2-free codes, and the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

[1] K. A. S. Immink, *Codes for Mass Data Storage Systems*. Amsterdam, The Netherlands: Shannon Foundation, 1999.
[2] ——, "Spectrum shaping with binary dc^2-constrained channel codes," *Phillips J. Res.*, vol. 40, pp. 40–53, 1985.
[3] R. Karabed and P. H. Siegel, "Matched spectral-null codes for partial-response channels," *IEEE Trans. Inform. Theory*, vol. 37, pp. 818–855, May 1991.
[4] E. Eleftheriou and R. D. Cideciyan, "On codes satisfying Mth-order running digital sum constraints," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1294–1313, Sept. 1991.
[5] C. Monti and G. L. Pierobon, "Codes with a multiple spectral null at zero frequency," *IEEE Trans. Inform. Theory*, vol. 35, pp. 463–472, Mar. 1989.
[6] J. Justesen, "Information rate and power spectra of digital codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 457–472, May 1982.
[7] J. N. Franklin and J. R. Pierce, "Spectra and efficiency of binary codes without dc," *IEEE Trans. Commun.*, vol. COM-20, pp. 1182–1184, Dec. 1972.
[8] V. Braun and A. J. E. M. Janssen, "On the low-frequency suppression performance of dc-free runlength-limited modulation codes," *IEEE Trans. Consumer Electron.*, vol. 42, pp. 939–945, Nov. 1996.
[9] G. L. Cariolaro, G. L. Pierobon, and G. P. Tronca, "Analysis of codes and spectra calculations," *Int. J. Electronics*, vol. 55, pp. 35–79, July 1983.
[10] K. A. S. Immink and G. Beenker, "Binary transmission codes with high order spectral zeros at zero frequency," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 452–454, May 1987.
[11] R. M. Roth, P. H. Siegel, and A. Vardy, "High-order spectral-null codes—Constructions and bounds," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1826–1840, Nov. 1994.
[12] G. L. Pierobon, "Codes for zero spectral density at zero frequency," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 435–439, Mar. 1984.
[13] G. Bilardi, R. Padovani, and G. L. Pierobon, "Spectral analysis of functions of Markov chains with applications," *IEEE Trans. Commun.*, vol. COM-31, pp. 853–861, July 1983.
[14] J. W. Harris and H. Stocker, *Handbook of Mathematics and Computational Science*. New York: Springer-Verlag, 1998.
[15] T. M. Chien, "Upper bound on the efficiency of dc-constrained codes," *Bell Syst. Tech. J.*, vol. 49, pp. 2267–2287, Nov. 1970.
[16] K. J. Kerpez, "The power spectral density of maximum entropy charge constrained sequences," *IEEE Trans. Inform. Theory*, vol. 35, pp. 692–695, May 1989.
[17] Y. Xin, "High-order spectral-null multimode codes," Ph. D. dissertation, Dept. Elec. Comput. Eng., Univ. Alberta, Edmonton, AB, Canada, 2002.

Reconciliation of a Quantum-Distributed Gaussian Key

Gilles Van Assche, Jean Cardinal, *Member, IEEE*, and Nicolas J. Cerf, *Member, IEEE*

***Abstract*—Two parties, Alice and Bob, wish to distill a binary secret key out of a list of correlated variables that they share after running a quantum key distribution (QKD) protocol based on continuous-spectrum quantum carriers. We present a novel construction that allows the legitimate parties to get equal bit strings out of correlated variables by using a classical channel, with as little leaked information as possible. This opens the way to securely correcting nonbinary key elements. In particular, the construction is refined to the case of Gaussian variables as it applies directly to recent continuous-variable protocols for QKD.**

***Index Terms*—Cryptography, privacy amplification, quantum secret key distribution, secret-key agreement.**

I. INTRODUCTION

With the advent of quantum key distribution (QKD), sometimes also called quantum cryptography, it is possible for two remote parties, Alice and Bob, to securely agree on secret information that shall later be used as a key for encrypting messages [1]–[4]. Although most QKD schemes make use of discrete modulation of quantum states, such as BB84 [1], some recent protocols [5], [6] use a continuous modulation of quantum states, thus producing continuous random variables. In particular, in [7], a QKD scheme based on the Gaussian modulation of quantum coherent states is demonstrated, which generates correlated Gaussian variables at Alice's and Bob's sides. The construction of a common secret key from discrete variables partly known to an adversary has been a long studied problem [8]–[11]. However, in order to bring the intrinsically continuous QKD experiments up to getting a usable secret key, such key construction techniques needed to be adapted to Gaussian variables.

In QKD, the quantum channel that Alice and Bob use to create a secret key is not deemed to be perfect. Noise will necessarily make Alice's and Bob's values different. Furthermore, the laws of quantum mechanics imply that eavesdropping also causes extra discrepancies, making the eavesdropper detectable. To overcome this, one can correct errors by using some *reconciliation* protocol, carried out over a public authenticated channel [9], [10]. Yet, this does not entirely solve the problem as an eavesdropper can gain some information about the key while Alice and Bob exchange their public reconciliation messages. Fortunately, such gained information can then be wiped out, at the cost of a reduction in the secret key length, using another protocol called *privacy amplification* [8], [11].

Manuscript received March 20, 2001; revised August 14, 2003. This work was supported by the Communauté Française de Belgique under Grant ARC 00/05-251 and by the IUAP program of the Belgian Government under Grant V-18. The material in this correspondence was presented in part at the ESF Workshop on Continuous Variable Quantum Information Processing, Brussels, Belgium, April 2002.

G. Van Assche and N. J. Cerf are with the Centre for Quantum Information and Communication, Ecole Polytechnique, CP 165, Université Libre de Bruxelles, 1050 Brussels, Belgium (e-mail: gvanassc@ulb.ac.be; ncerf@ulb.ac.be).

J. Cardinal is with the Computer Science Department, Faculté des Sciences, CP 212, Université Libre de Bruxelles, 1050 Brussels, Belgium (e-mail: jcardin@ulb.ac.be).

Communicated by P. W. Shor, Associate Editor for Quantum Information Theory.

Digital Object Identifier 10.1109/TIT.2003.822618

Current reconciliation and privacy amplification protocols are aimed at correcting and distilling strings of bits. However, the recently developed continuous-variable QKD schemes cannot be efficiently used with such discrete protocols. This correspondence proposes an extention of these protocols in the case of (discrete or continuous) nonbinary key elements, with a special treatment of the Gaussian case.

II. Quantum Distribution of a Gaussian Key

In QKD, Alice and Bob use a quantum channel in order to share secret random data (a secret key) that can then be used for exchanging encrypted information. By using the shared secret key with one-time pad encryption, the security of this cryptosystem depends on the ability of Alice and Bob to minimize the amount of information an eavesdropper (Eve) can acquire on this key. QKD offers this without any computational assumption, in contrast to classical cryptography.

Since its inception, QKD has traditionally been developed with discrete quantum carriers, especially quantum bits (implemented e.g., as the polarization state of single photons). Yet, it has been shown recently that the use of continuous quantum carriers is advantageous in some situations, namely, because high secret key bit rates can be attained [6]. The postprocessing of the raw data produced by such continuous-variable protocols therefore deserves further investigation.

As we shall see, the security of QKD fundamentally relies on the fact that the measurement of *incompatible* variables inevitably affects the state of a quantum system. With the information encoded in such incompatible variables, eavesdropping becomes thus measurable. In a scheme such as BB84, Alice sends random key elements (e.g., key bits) to Bob using either one of two conjugate sets of quantum information carriers. Alice randomly chooses one of the two sets of carriers, encodes a random key element using this set, and sends it to Bob. On his side, Bob measures the received quantum state assuming either set was used at random. The two sets of quantum information carriers are designed in such a way that measuring the wrong set yields random uncorrelated results (i.e., the two sets are conjugate). Therefore, Bob will measure correctly only half of the key elements Alice sent him, not knowing which ones are wrong. After the process, Alice reveals which set of carriers she chose for each key element, and Bob is then able to discard all the wrong measurements, the remaining data making the key.

An eavesdropper (Eve) can, of course, intercept the quantum carriers and try to measure them. However, like Bob, Eve does not know in advance which set of carriers Alice chose for each key element. A measurement will yield irrelevant results about half of the time, and thereby disturb the state of the carrier. Not knowing if she has a correct value, Eve can decide to retransmit or not a quantum carrier with the key element she obtained. Discarding a key element is useless for Eve since this sample will not be used by Alice and Bob to make the key. Then, if she does retransmit the state (even though it is wrong half of the time), Alice and Bob will detect her presence by an unusually high error rate between their key elements. Stated otherwise, quantum information cannot be cloned, thereby making noticeable any information gained by an eavesdropper. QKD works because Bob has the advantage, over Eve, of being able to talk to Alice over a classical authenticated channel in order to select a common key and discard Eve's partial knowledge on it.

The continuous-variable QKD protocols described in [5], [6] take advantage of a pair of canonically conjugate continuous variables such as the two quadratures X_1 and X_2 of the amplitude of a mode of the electromagnetic field, which behave just like position x and momentum p [12]. The uncertainty relation $\Delta X_1 \, \Delta X_2 \geq 1/4$ then states that it is impossible to measure with full accuracy *both* quadratures of a single mode, X_1 and X_2. This can be exploited by associating the two sets of quantum information carriers with X_1 and X_2, respectively. For example, in the protocol [5], these two sets of carriers essentially behave like two–dimensional (2-D) Gaussian distributions in the (X_1, X_2) plane (Wigner function, see, e.g., [12]). In Set 1, the carriers are squeezed states, shaped as $N(x, \sigma_1) \times N(0, 1/4\sigma_1)$, with $\sigma_1 < 1/2$ corresponding to the squeezing of X_1. (So, for instance, the measure of X_2 of this state gives a random result distributed as $N(0, 1/4\sigma_1)$, even with a perfect apparatus.) Here, x is the key element Alice wishes to send, and is itself distributed as a Gaussian: $x \sim N(0, \Sigma_1)$. In Set 2, the carriers are similar but X_1 and X_2 are interchanged, that is, $N(0, 1/4\sigma_2) \times N(x, \sigma_2)$, with $\sigma_2 < 1/2$. The raw key information is thus encoded sometimes in X_1 and sometimes in X_2, and the protocol resembles a continuous version of BB84. In contrast, in [6], two Gaussian raw key elements x_1 and x_2 are simultaneously encoded in a coherent state shaped as $N(x_1, 1/2) \times N(x_2, 1/2)$ in the (X_1, X_2) plane. Bob can, however, only measure one of them, not both, so that only one Gaussian value $x = x_{1 \text{ or } 2}$ is really transmitted. Eve, not knowing which one Bob will measure, necessarily disturbs x_1 when attempting to infer x_2 and *vice versa*, and she in general disturbs both to some extent whatever the tradeoff between acquired knowledge and induced disturbance she chooses.

In all these continuous-variable protocols, the vacuum noise fluctuations of the transmitted states are such that Bob's measurement will not give him the exact value x chosen by Alice, even in the absence of eavesdropping and with a perfect measurement apparatus. The noise is Gaussian and additive, allowing us to model the transmission as a Gaussian channel. The amplitude of the noise can be estimated by Alice and Bob when they compare a subset of their exchanged values. Any noise level beyond the intrinsic fluctuations must be attributed to Eve, giving an estimate on the amount of information $I(X; E)$ that she was able to infer in the worst case [5]–[7]. This information, along with the information Eve gains by monitoring the reconciliation protocol, must then be eliminated via privacy amplification.

For the above protocols, the choice of Gaussian states and modulations results from both physical and practical reasons. First, Gaussian states saturate the uncertainty relation, i.e., $\Delta X_1 \, \Delta X_2 = 1/4$, which make their study quite natural. Second, the particular class of coherent states are fairly easy to produce, as chosen in [7]. Third, the Gaussian modulation makes such QKD protocols easier to analyze in a Gaussian formalism. Finally, the Gaussian modulation is easy to implement if one accepts a cutoff of far-aside tails.

Finally, note that Alice must strictly respect $x \sim N(0, \Sigma_{1 \text{ or } 2})$ or $(x_1, x_2) \sim N(x_1, 1/2) \times N(x_2, 1/2)$. She may not choose a codebook $x(k)$ from some discrete alphabet to $\mathbb{R}$ that displays the same variance. The resulting distribution would not be Gaussian, and Eve would be able to take advantage of this situation. For example, in [5], measuring the correct or the wrong set must yield statistically indistinguishable results. If this were not the case, Eve would be able to infer whether she measured the correct set of carriers and adapt her strategy to this knowledge.

III. Problem Description

A. Problem Statement

The two parties each have access to a distinct random variable, namely, X for Alice and X' for Bob, with nonzero mutual information $I(X; X') > 0$. This models the quantum modulation and measurement of a QKD scheme, but other sources of common randomness could as well be used. When running the same QKD protocol several times, the instances of X (resp., X') are denoted $X_1 \ldots X_l$ (resp., $X'_1 \ldots X'_l$) for the time slots $1 \ldots l$, and are assumed independent for different time slots. The outcomes are denoted with the corresponding lower case letters. An eavesdropper Eve also has access to a random

variable E, resulting from tapping the quantum channel. It is also considered independent for different time slots, hence assuming individual attacks [4]. Note that we will focus in Section VI on the case of (X, X') being joint Gaussian variables, as they naturally arise in the QKD protocols described above.

The goal of the legitimate parties is to distill a secret key, i.e., to end up with a shared binary string that is unknown to Eve. We assume, as a convention, that Alice's outcomes of X will determine the shared key $K(X)$. It is, of course, not a problem if the roles of Alice and Bob are reversed, as required in [7]. The function $K(X)$ is chosen to be discrete, even if X is continuous in nature, and this aspect is discussed in what follows.

In principle, secret key distillation does not require separate reconciliation and privacy amplification procedures, but it is much easier to use such a two-step approach.

First, reconciliation consists in exchanging reconciliation messages over the public authenticated classical channel, collectively denoted C, so that Bob can recover $K(X_{1\dots l})$ from C and $X'_{1\dots l}$. By compressing $K(X_{1\dots l})$, Alice and Bob can obtain about $lH(K(X))$ common random bits.

Then, privacy amplification can be achieved by universal hashing [11], [13]. Starting from $K(X_{1\dots l})$, the decrease in key length is roughly equal to $lI(K(X);E) + |C|$, as shown in [11], [14], [15], where $|C|$ is the number of bits exchanged and where $I(K(X);E)$ is determined from the disturbance measured during the QKD procedure. Privacy amplification therefore does not need special adaptations in our case, as the existing protocols can readily be used.

Maximizing the net secret key rate

$$H(K(X)) - I(K(X);E) - l^{-1}|C|$$

involves taking all possible eavesdropping strategies into account during the optimization, which is very difficult in general. Instead, we notice that $I(K(X);E) \leq I(X;E)$, the latter being independent of the reconciliation procedure. Thus, we wish to devise a procedure that produces a large number of fully secret equal bits, hence to maximize $H(K(X)) - l^{-1}|C|$.

Note that this problem is not equivalent to known transmission schemes, namely, quantization and coded modulation.

In a quantization system, a random input variable X is transmitted over a noiseless discrete channel using the index of the closest code-vector in a given codebook. More precisely, X is encoded as the discrete value $\alpha(X)$, which is then transmitted noiselessly. From this, the decoder decodes $\hat{X} = \beta(\alpha(X))$, and the functions α and β are chosen so as to minimize some average distortion measure d (e.g., the Euclidean distance) between the input and the decoded vector $E[d(X, \beta(\alpha(X))]$. The codebook design issue has been extensively studied in the literature [16]. In our problem, we do not have reproduction vectors since we are not interested in reproducing the continuous code but rather extracting common discrete information between two random variables X and X'. Furthermore, the quantities to optimize are not the same, namely, the average distortion to minimize for quantization and the amount of secret bits to maximize for our problem. Techniques inspired from quantization can be used to find $K(X)$ that maximizes $I(K(X);X')$, which is discussed in [17] and the references therein. Yet, this still must be completed with appropriate reconciliation to extract common information between $K(X)$ and X'.

In a coded modulation system, a binary key k is sent over a continuous noisy channel using a vector X belonging to a codebook in a Euclidean space. Trellis-coded modulation and lattice-based coded modulation are instances of this scheme. In this case, the information sent on the channel is chosen by Alice in a codebook, which is not true in our case.

B. Discrete Versus Continuous Variables

It is shown in [5]–[7] that working with continuous quantum states as carriers of information naturally leads to expressing information in a continuous form. It is therefore natural to devise an all-continuous cryptographic processing. Nevertheless, we found more advantageous to distill a discrete secret key than a continuous one, and these aspects are now discussed.

First, a continuous secret key would need to be used along with a continuous version of the one-time pad, which is possible [18], but this would be difficult to make noise resistant. It is much more convenient to rely on the equality of Alice's and Bob's values in the discrete case, rather than dealing with bounded errors on real numbers. The resulting secret key is thus chosen to be discrete.

Second, the reconciliation messages can either be continuous or discrete. Unless the public authenticated classical channel has infinite capacity, exchanged reconciliation messages are either discrete or noisy continuous values. The latter case introduces additional uncertainties into the protocol, which quite goes against our purposes. Furthermore, a noisy continuous reconciliation message would less efficiently benefit from the authentication feature of the reconciliation channel. Hence, discrete reconciliation messages are preferred.

Third, the choice of a discrete final key also induces discrete effects in the protocols, which makes natural the choice of a continuous-to-discrete conversion during reconciliation. The process of reconciliation and privacy amplification can be summarized as functions $k = f_A(x, c)$ and $k = f_B(x', c)$ to produce the key k, where c indicate the exchanged messages. As both k and c are to be taken in some finite set, these two functions define each a finite family of subsets of values that give the same result $S_{kc} = \{x : f_A(x, c) = k\}$ and $S'_{kc} = \{x' : f_B(x', c) = k\}$. The identification of the subset in which x (or x') lies is the only data of interest—and can be expressed using discrete variables—whereas the value within that subset does not affect the result and can merely be considered as noise.

Finally, the discrete conversion does not put a fundamental limit on the resulting efficiency. It is possible (see Section IV) to bring $|C|$ as close as desired to $lH(K(X)|X')$, giving almost $I(K(X);X')$ secret bits per raw key element (excluding quantum eavesdropping). Also, one can define $K(X)$ as a fine-grained quantizer so that $I(K(X);X')$ can be made arbitrarily close to $I(X;X')$ [19]. On the other hand, no continuous protocol can expect Alice and Bob to share more secret information than what they initially share $I(X;X')$.

For all the reasons stated in the preceding discussion, our reconciliation protocol mainly consists of exchanging discrete information between the two communicating parties so that they can deduce the same discrete representation from the real values they initially share.

IV. Sliced Error Correction

Sliced error correction (SEC) is a generic reconciliation protocol that corrects strings of nonbinary elements. It gives, with high probability, two communicating parties, Alice and Bob, equal binary digits from a list of correlated values. Just like other error-correction protocols, it makes use of a public authenticated channel. The underlying idea is to convert Alice's and Bob's values into strings of bits, apply a binary correction protocol (BCP) as a primitive and take advantage of all available information to minimize the number of exchanged reconciliation messages.

The key feature of this generic protocol is that it enables Alice and Bob to correct errors that are not modeled as a binary symmetric channel (BSC), although using a BCP that is optimized for a BSC.

To remain general, Alice and Bob can process multidimensional key values and group them into d-dimensional vectors. In the sequel, X and X' denote d-dimensional variables, taking values in what is defined

as the raw key space, i.e., $\mathbb{R}^d$ for Gaussian variables. When explicitly needed by the discussion, the dimension of the variables is noted with a $\cdot^{(d)}$ superscript.

To define the protocol, we must first define the slice functions. A slice $S(x)$ is a function from Alice's raw key space to GF(2). A vector of slices $S_{1,\dots,m}(x) = (S_1(x), \dots, S_m(x))$ is chosen so as to map Alice's raw key elements to a discrete alphabet of size at most 2^m. A vector of slices will convert Alice's raw key elements into binary digits, that is, $K(x) = S_{1\dots m}(x)$.

Each of the slice estimators

$$\tilde{S}_1(x'),\ \tilde{S}_2(x', S_1(x)), \dots, \tilde{S}_m(x', S_1(x), \dots, S_{m-1}(x))$$

defines a mapping from Bob's raw key space and from Alice's slices of lower indexes to GF(2). These will be used by Bob to guess $S_i(X)$ the best he can given his knowledge of X' and of the slice bits previously corrected.

The construction of the slices $S_i(X)$ and their estimators depends on the nature and distribution of the raw key elements. These aspects are covered in Section V, where we apply the SEC to our Gaussian key elements.

Let us now describe our generic protocol, which assumes that the legitimate parties defined and agreed on the functions S_i and $\tilde{S}_i$. Alice (resp., Bob) processes l key elements x_j (resp., x'_j), $j = 1 \dots l$—be reminded that the corresponding random variables X_j (resp., X'_j) are independent for different time slots j.

- For $i = 1$ to m, successively, Alice and Bob perform the following steps:

— Alice prepares the string of bits $(S_i(x_1), \dots, S_i(x_l))$.

— Bob prepares the string of bits

$$(\tilde{S}_i(x'_1, S_{1\dots i-1}(x_1)), \dots, \tilde{S}_i(x'_l, S_{1\dots i-1}(x_l)))$$

where is $S_{1\dots i-1}(x_1)$ is known to Bob, with high probability, from the previous $i-1$ steps.

— Alice and Bob make use of a chosen BCP so that Bob acquires the knowledge of Alice's bits $(S_i(x_1), \dots, S_i(x_l))$.

The goal of SEC is for Alice and Bob to gather common bits (i.e., $l \times m$ bits $K(x_j) = S_{1\dots m}(x_j)$, $j = 1 \dots l$) by disclosing as little information as possible on them. However, one does not expect a protocol running with strings of finite length and using finite computing resources to achieve the Shannon bound $I(X; X')$ exactly. Yet, it is easy to show that SEC is indeed asymptotically efficient, that is, it reaches the Shannon bound in terms of leaked information when the number of dimensions d (i.e., the input alphabet size) goes to infinity.

A famous theorem by Slepian and Wolf [20] shows the achievability rate regions for encoding correlated sources. In the context of SEC, this means that, with d sufficiently large, there exist slice functions such that disclosing the first $r = \lfloor dH(K(X^{(1)})|X'^{(1)}) + 1 \rfloor$ slices $S_{1\dots r}(X^{(d)})$ is enough for Bob to recover the $m-r$ remaining ones and reconstruct $S_{1,\dots,m}(X^{(d)})$ with arbitrarily low probability of error.

For continuous variables X', it is necessary here to quantize X', as Slepian and Wolf's theorem assumes discrete variables. As shown in [19], X' can be approximated as accurately as necessary by a discrete variable $\hat{X}'$, with $H(K(X)|\hat{X}') \to H(K(X)|X')$.

V. Analysis of Sliced Error Correction

Let us now analyze the amount of information leaked on the public channel during SEC. Clearly, this will depend on the primitive BCP chosen. This aspect will be further duscussed in Section VI.

If not using SEC, one can in theory use encoding of correlated information [20] to achieve, when $l \to \infty$

$$l^{-1}|C| = I_0 \triangleq H(S_{1\dots m}(X)|X'). \tag{1}$$

When using slices, however, the BCP blindly processes the bits calculated by Alice $S_i(X)$ on one side and the bits calculated by Bob $\tilde{S}_i(X', S_{1\dots i-1}(X))$ on the other side. The l bits produced by the slices are of course independent from one time slot to another. Assuming a perfect BCP

$$l^{-1}|C| = I_s \triangleq \sum_{i=1}^{m} H(S_i(X)|\tilde{S}_i(X', S_{1\dots i-1}(X))) \geq I_0. \tag{2}$$

The inequality follows from the fact that

$$H(S_{1\dots m}(X)|X') = \sum_i H(S_i(X)|X', S_{1\dots i-1}(X))$$

and that the term in the sum cannot decrease if replaced by $H(S_i(X)|\tilde{S}_i(X', S_{1\dots i-1}(X)))$. The primitive BCP can be optimized to work on a binary symmetric channel (BSC-BCP), thus processing the bits as if balanced both on Alice's and Bob's sides. This is, of course, suboptimal for unbalanced bit strings as the actual redundancies cannot be exploited. Assuming a perfect BSC-BCP

$$l^{-1}|C| = I_e \triangleq \sum_{i=1}^{m} h(e_i) \geq I_s \tag{3}$$

with

$$h(e) = -e \log e - (1-e)\log(1-e)$$

and

$$e_i = \Pr[S_i(X) \neq \tilde{S}_i(X', S_{1\dots i-1}(X))].$$

The inequality follows from Fano's inequality [19] applied to a binary alphabet. In practice, a BSC-BCP is expected to disclose a number of bits that is approximately proportional to $h(e)$, i.e., $(1+\xi)h(e)$ for some overhead constant ξ, see Section V-B.

Note that in the case of asymptotically large block sizes, $d \to \infty$, the quantities I_0, I_s, and I_e tend to the same limit $dH(K(X^{(1)})|X'^{(1)})$ since the first slices can be completely disclosed, determining the remaining ones with arbitrarily small error probabilities, as shown in Section IV.

An explicit construction of slice estimators applying the expression of I_e in (3) is examined next.

A. Maximum-Likelihood Slice Estimators

Maximizing the global efficiency of the SEC protocol for a given pair of variables X and X' is not a simple task because the number of key bits produced and leaked with slice i recursively depends on the design of the previous slices $1 \dots i-1$. For this reason, our goal in this section is simply to minimize I_e by acting on each slice estimator $\tilde{S}_i$ independently. More precisely, we will minimize each e_i, of which $h(e_i)$ is an increasing function for $0 \leq e_i < \frac{1}{2}$, so as to locally minimize the number of leaked bits $l^{-1}|C|$ without changing the number of produced bits $H(K(X))$. This approach has the advantage of also allowing the optimization of $l^{-1}|C| = (1+\xi)I_e$ for a nonideal BSP-BCP, and results in an explicit expression for $\tilde{S}_i(x', S_{1\dots i-1}(x))$, see (5).

The following equations assume continuous variables X and X' with probability density function (pdf) $p(x, x')$. (Note that the same result also applies to discrete variables, with integrals over x' replaced by sums.) The error probability in slice i is the probability that Bob's slice estimator yields a result different from Alice's slice

$$e_i = \Pr[S_i(X) \neq \tilde{S}_i(X', S_{1\dots i-1}(X))]$$

which can be expanded as

$$e_i = \int dx' \sum_{\beta \in GF(2)^{i-1}} \Pr[S_i(X) \neq \tilde{S}_i(x', \beta) \wedge S_{1\ldots i-1}(X) = \beta \wedge X' = x']. \quad (4)$$

Each term of the right-hand side of (4) integrates $p(x, x')$ over nonoverlapping areas of the (x, x') plane, namely,

$$\{(x, x') : S_{1\ldots i-1}(x) = \beta\}.$$

So, each of them can be minimized independently of the others, and thus to minimize e_i, $\tilde{S}_i$ must satisfy

$$\begin{aligned}\tilde{S}_i(x', \beta) &= \arg\min_{\tilde{s}} \Pr[S_i(X) \neq \tilde{s} \wedge S_{1\ldots i-1}(X) = \beta \wedge X' = x'] \\ &= \arg\max_{\tilde{s}} \Pr[S_i(X) = \tilde{s} | S_{1\ldots i-1}(X) = \beta, X' = x'] \quad (5)\end{aligned}$$

with an appropriate tie-breaking rule.

Since the slice estimators are now determined by the slice functions S_i and the pdf $p(x, x')$, the bit error probability e_i can be evaluated as

$$e_i = \int dx' \sum_{\beta \in GF(2)^{i-1}} \min_a \Pr[S_i(X) = a \wedge S_{1\ldots i-1}(X) = \beta \wedge X' = x']. \quad (6)$$

Following intuition, the error probability is minimal when the variables x' and $\beta_{1\ldots i-1}$ determine $S_i(x)$ without ambiguity. It now remains to optimize only the functions S_i, which is done for Gaussian variables in Section VI.

B. Binary Correction Protocols

To be able to use SEC, it is necessary to chose a suitable BCP. There are two trivial protocols that are worth noting. The first consists in disclosing the slice entirely, while the second does not disclose anything. These are at least of theoretical interest with the asymptotical optimality of SEC: it is sufficient for Alice to transmit entirely the first $r = \lfloor dH(K(X^{(1)})|X'^{(1)}) + 1 \rfloor$ slices and not transmit the remaining $m - r$ ones.

A BCP can consist in sending syndromes of error-correcting codes, see, e.g., [21]. In binary QKD protocols, however, an interactive reconciliation protocol is often used, such as Cascade [9], [22]–[24] or Winnow [25]. In practice, interactivity offers overwhelmingly small probability of errors at the end of the protocol, which is valuable for producing a usable secret key.

Let us briefly analyze the cost of Cascade, which consists in exchanging parities of various subsets of bits [9]. Let $A, B \in \mathrm{GF}(2)^l$ be respectively Alice's and Bob's binary string of size l constructed from some slice S_i and its estimator $\tilde{S}_i$. After running Cascade, Alice (resp., Bob) disclosed RA (resp., RB) for some matrix R of size $n \times l$. They thus communicated the parities calculated over identical subsets of bit positions. The matrix R and the number n 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. The expected value of n is $n \approx l(1+\xi)h(e)$, where $e = \Pr[A_j \neq B_j]$ is the bit-error rate, and ξ is some small overhead factor.

If A and B are balanced and are connected by a BSC, the parities RA give Eve n 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 only $n \approx l(1 + \xi)h(e)$ bits are disclosed, which is not far away from the ideal $lh(e)$.

However, in the more general case, where Eve gathered in E some information on A and B by tapping the quantum channel, $A|E \rightarrow RA|E \rightarrow RB|E$ does not necessarily form a Markov chain. Instead, the cost must be upper-bounded by the number of bits disclosed by both parties as if they were independent, $|C| = 2n \approx 2l(1 + \xi)h(e)$.

Such a penalty is the result of interactivity, as both Alice and Bob disclose some information. This can, however, be reduced by noticing that RA and RB can also be equivalently expressed by RA and $R(A+B)$. The first term RA gives information directly on Alice's bits $A = S_i(X_{1\ldots l})$ for some slice number i, which are used as a part of the key. The second term $R(A + B)$, however, contains mostly noise and does not contribute much to Eve's knowledge on the key. This must however be explicitly evaluated with all the details of the QKD protocol in hand [7].

With SEC, it is not required to use the same protocol for all slices. Noninteractive and interactive BCPs can be combined. In the particular case of slices with large e_i, disclosing the entire slice may cost less than interactively correcting it. Overall, the number of bits revealed is

$$|C| = \sum_i |C_i|, \qquad \text{with } |C_i| = \min(l, f_i(l, e_i)) \quad (7)$$

and $f_i(l, e_i)$ the expected number of bits disclosed by the BCP assigned to slice i working on l bits with a bit-error rate equal to e_i.

As d grows and it becomes sufficient to only disclose the first r slices so as to leave an acceptable residual error, using a practical BCP comes closer to the bound $l^{-1}|C| \geq H(K(X)|X')$. This follows from the obvious fact that

$$l^{-1} \sum_{i=1}^{r} |C_i| \leq r$$

while the last slices can be ignored $f_i = 0, i > r$.

VI. Correction of Gaussian Key Elements

A. Design

We must now deal with the reconciliation of information from Gaussian variables $X \sim N(0, \Sigma)$ and $X' = X + \epsilon$, $\epsilon \sim N(0, \sigma)$. In this section, we present how we designed slices and slice estimators for extracting a common key from such raw key elements. We assume $d = 1$, that is, Alice and Bob use Gaussian key elements individually. The idea is to divide the set of real numbers into intervals and to assign slice values to each of these intervals. The slice estimators are then derived as maximum-likelihood estimators as explained earlier.

For simplicity, the design of the slices was divided into two smaller independent problems. First, we cut the set of real numbers (Alice's raw key space) into a chosen number of intervals—call this process $T(X)$. For the chosen number of intervals, we try to maximize $I(T(X); X')$. Second, we assign m binary values to these intervals in such a way that slices can be corrected with as little leaked information as possible.

If the reconciliation is optimal, it produces $H(T(X))$ common bits and discloses I_0 bits, thus, from (1), giving a net result of

$$H(T(X)) - H(T(X)|X') = I(T(X); X') \text{ bits.}$$

Note that $S_{1\ldots m}(X)$ will be an invertible function of $T(X)$. However, optimizing $I(T(X); X')$ does not depend on the bit assignment, so this is not yet relevant.

The process $T(X)$ of dividing the real numbers into t intervals is defined by $t - 1$ variables $\tau_1, \ldots, \tau_{t-1}$. The interval a with $1 \leq a \leq t$ is then defined by the set $\{x : \tau_{a-1} \leq x < \tau_a\}$ where $\tau_0 = -\infty$ and $\tau_t = +\infty$. The function $I(T(X); X')$ was numerically maximized under the symmetry constrains $\tau_a = \tau_{t-a}$ to reduce the number of variables to process.

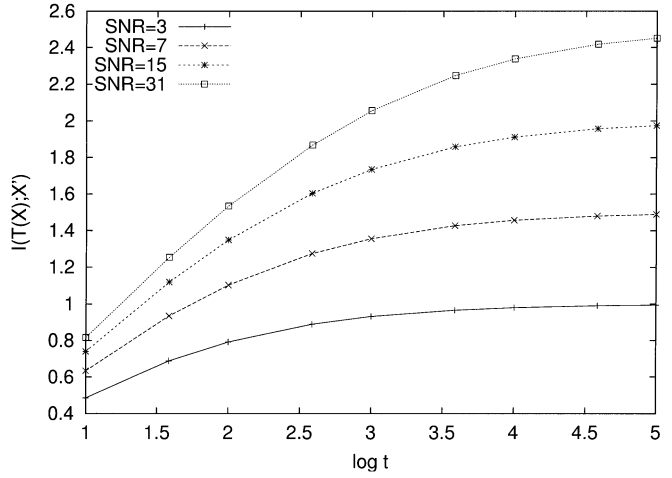

Fig. 1. Optimized $I(T(X);X')$ as a function of $\log t$ for various signal-to-noise ratios, with t the number of intervals.

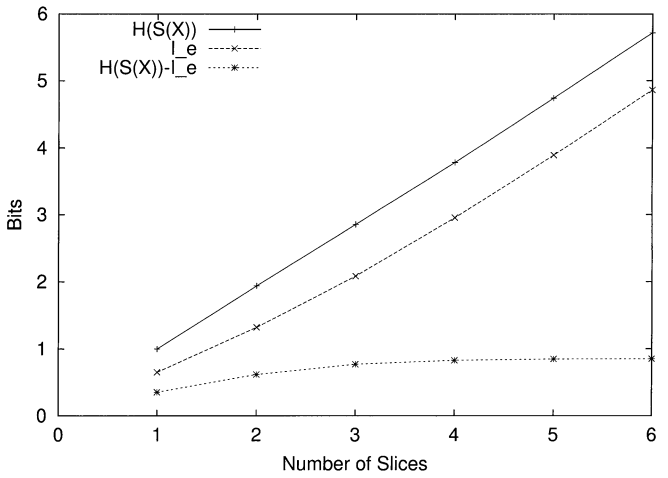

Fig. 2. $H(S_{1,\ldots,m}(X))$, I_e and their difference as a function of the number of slices m when $\Sigma^2/\sigma^2 = 3$.

The results are displayed in Fig. 1. $I(T(X);X')$ is bounded from above by $\log t$ and goes to $\frac{1}{2}\log(1+\text{SNR})$ as $t \to \infty$. (All the logarithms are in base 2.)

From the above procedure, we get intervals that are bounded by the thresholds τ_a. The next step is to construct m slices that return binary values for each of these intervals. Let us restrict ourselves to the case where t is a power of two, namely, $t = 2^m$. We investigated several assignment methods, and it turned out that the best bit assignment method consists of assigning the least significant bit of the binary representation of $a-1$ ($0 \le a-1 \le 2^m-1$) to the first slice $S_1(x)$ when $\tau_{a-1} \le x < \tau_a$. Then, each bit of $a-1$ is subsequently assigned up to the most significant bit, which is assigned to the last slice $S_m(x)$. More explicitly

$$S_i(x) = \begin{cases} 0, & \text{if } \tau_{2^i n} \le x < \tau_{2^i n + 2^i - 1} \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

This ensures that the first slices containing noisy values help Bob narrow down his guess as quickly as possible.

B. Numerical Results

Let us now give some numerical examples in the case of a BCP optimized for a BSC, as this is the most frequent case in practice. To make the discussion independent of the chosen BCP, we evaluated $H(S_{1,\ldots,m}(X))$ and $I_e = \sum_i h(e_i)$ for several $(m, \Sigma/\sigma)$ pairs, thus assuming a perfect BSC-BCP. (Note that, in practice, one can make use of the properties of the practical BCP chosen so as to optimize the practical net secret key rate [7].)

Assume that the Gaussian channel has a signal-to-noise ratio Σ^2/σ^2 of 3. According to Shannon's formula, a maximum of 1 bit can thus be transmitted over such a channel. Various values of m are plotted in Fig. 2. First, consider the case $m = 1$, that is only 1 bit is extracted and corrected per Gaussian value. From our construction in (8), the slice reduces to the sign of x: $S_1(x) = 1$ when $x \ge 0$ and $S_1(x) = 0$ otherwise. Accordingly, Bob's maximum-likelihood estimator (5) is equivalent to Alice's slice, $\hat{S}_1(x') = S_1(x')$. In this case, the probability that Alice's and Bob's values differ in sign is $e_1 \approx 0.167$ and, hence, $I_e = h(e_1) \approx 0.65$ bit. The net amount of information is thus approximately $1 - 0.65 = 0.35$ bit per raw key element.

TABLE I
SYMMETRIC INTERVAL BOUNDARIES THAT MAXIMIZE $I(T(X);X')$, WITH $\Sigma = 1$ AND $\sigma = 1/\sqrt{3}$

τ_8	0	$\tau_{12} = -\tau_4$	1.081
$\tau_9 = -\tau_7$	0.254	$\tau_{13} = -\tau_3$	1.411
$\tau_{10} = -\tau_6$	0.514	$\tau_{14} = -\tau_2$	1.808
$\tau_{11} = -\tau_5$	0.768	$\tau_{15} = -\tau_1$	2.347

Let us now investigate the case of $m = 4$ slices, still with a signal-to-noise ratio of 3. The division of the raw key space into intervals that maximizes $I(T(X);X')$ is given in Table I. Note that the generated intervals blend evenly distributed intervals and equal-width intervals. Evenly distributed intervals maximize entropy, whereas equal-width intervals best deal with additive Gaussian noise.

Alice's slices follow (8), and Bob's slice estimators are defined as usual using (5). The correction of the first two slices (i.e., the least two significant bits of the interval number) have an error rate that makes them almost uncorrelated, namely, $e_1 \approx 0.496$ and $e_2 \approx 0.468$. Then comes $e_3 \approx 0.25$ and $e_4 \approx 0.02$. Note that slice 4 gives the sign of x, just like the only slice when $m = 1$. The error rate is much lower here because correcting slice 4 in this case benefits from the correction of the first three slices. Indeed, for $m = 4$, the net amount of information is about $3.78 - 2.95 = 0.83$ bit per raw key element.

We also investigated other signal-to-noise ratios. When $\Sigma^2/\sigma^2 = 15$, Alice and Bob can share up to 2 bits per raw key element. With $m = 5$, this gives a net amount of information of about 1.81 bits per raw key element.

As one can notice, the first few error rates (e.g., e_1 and e_2) are high and then the subsequent ones fall dramatically. The first slices are used to narrow down the search among the most likely possibilities Bob can infer, and then the last slices compose the shared secret information. Also, slices with high error rates play the role of sketching a hypothetical codebook to which Alice's value belongs. After revealing the first few slices, Bob knows that her value lies in a certain number of narrow intervals with wide spaces between them. If Alice had the possibility of choosing a codebook, she would pick up a value from a discrete list of values—a situation similar to the one just mentioned except for the interval width. Using more slices $m > 4$ would simply make these codebook-like intervals narrower.

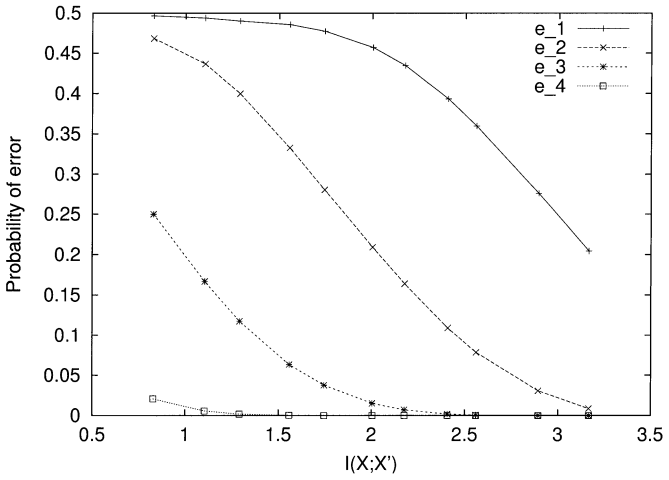

Fig. 3. Error rates $e_{1,2,3,4}$ as a function of the channel capacity $I(X;X')$.

In Fig. 3, we show these error rates for $m = 4$ when the noise level varies. From the role of sketching a codebook, slices gradually gain the role of really extracting information as their error rates decrease with the decreasing noise level.

VII. Conclusion

Current reconciliation procedures are aimed at correcting strings of bits. A new construction for reconciliation was proposed, which can be implemented for extracting common information out of any shared variables, either discrete or continuous. This construction is then applied to the special case of Gaussian key elements, in order to complement Gaussian-modulated QKD schemes [5]–[7]. This might also be applied to other QKD schemes [26]–[29] that deal with continuous variables as well. We showed theoretical results on the optimality of our construction when applied to asymptotically large block sizes. Numerical results about reconciliation of Gaussian key elements show that such a construction does not leak much more information than the theoretical bound.

Acknowledgment

The authors wish to thank Philippe Grangier and Frédéric Grosshans for fruitful discussions.

References

[1] C. H. Bennett and G. Brassard, "Public-key distribution and coin tossing," in *Proc. IEEE Int. Conf. Computers, Systems, and Signal Processing*, Bangalore, India, 1984, pp. 175–179.

[2] C. H. Bennett, "Quantum cryptography using any two nonorthogonal states," *Phys. Rev. Lett.*, vol. 68, p. 3121, 1992.

[3] C. H. Bennett, G. Brassard, and A. K. Ekert, "Quantum cryptography," *Scientific Amer.*, vol. 267, pp. 50–57, Oct. 1992.

[4] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Rev. Mod. Phys.*, vol. 74, p. 145, 2002.

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

[6] F. Grosshans and P. Grangier, "Continuous variable quantum cryptography using coherent states," *Phys. Rev. Lett.*, vol. 88, p. 057902, Feb. 2002.

[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*, vol. 421, pp. 238–241, 2003.

[8] C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy amplification by public discussion," *SIAM J. Comput.*, vol. 17, no. 2, pp. 210–229, 1988.

[9] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *Advances in Cryptology—Eurocrypt'93 (Lecture Notes in Computer Science)*, T. Helleseth, Ed. Berlin, Germany: Springer-Verlag, 1993, pp. 411–423.

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

[11] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, "Generalized privacy amplification," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1915–1923, Nov. 1995.

[12] M. O. Scully and M. S. Zubairy, *Quantum Optics*. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[13] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," *J. Comp. Syst. Sci.*, vol. 18, pp. 143–154, 1979.

[14] C. Cachin, "Entropy measures and unconditional security in cryptography," Ph. D. dissertation, ETH Zürich, Zürich, Switzerland, 1997.

[15] U. Maurer and S. Wolf, "Information-theoretic key agreement: From weak to strong secrecy for free," in *Advances in Cryptology—Eurocrypt 2000 (Lecture Notes in Computer Science)*, B. Preneel, Ed. Berlin, Germany: Springer-Verlag, 2000, pp. 351–368.

[16] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2325–2383, Oct. 1998.

[17] J. Cardinal and G. Van Assche, "Construction of a shared secret key using continuous variables," in *Proc. 2003 IEEE Information Theory Workshop (ITW2003)*, Paris, France, Mar./Apr. 2003.

[18] C. E. Shannon, "Analogue of the Vernam system for continuous time series," Bell Labs. Memo, MM 43-110-44, May 1943.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[20] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 471–480, July 1973.

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

[22] K. Chen, "Reconciliation by public discussion: Throughput and residue error rate," unpublished manuscript, 2001.

[23] T. Sugimoto and K. Yamazaki, "A study on secret key reconciliation protocol Cascade," *IEICE Trans. Fundamentals*, vol. E83-A, no. 10, pp. 1987–1991, Oct. 2000.

[24] K. Yamazaki and T. Sugimoto, "On secret key reconciliation protocol," in *Proc. IEEE Int. Symp. Information Theory and its Applications*, Honolulu, HI, Mar. 2000.

[25] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue, and C. G. Peterson, "Fast, efficient error reconciliation for quantum cryptography," *Phys. Rev. A.*, vol. 67, p. 052303, 2003.

[26] M. D. Reid, "Quantum cryptography with a predetermined key, using continuous-variable Einstein-Podolsky-Rosen correlations," *Phys. Rev. A*, vol. 62, p. 062308, 2000.

[27] T. C. Ralph, "Continuous variable quantum cryptography," *Phys. Rev. A*, vol. 61, p. 010303(R), 2000.

[28] M. Hillery, "Quantum cryptography with squeezed states," *Phys. Rev. A*, vol. 61, p. 022309, 2000.

[29] C. Silberhorn, N. Korolkova, and G. Leuchs, "Quantum key distribution with bright entangled beams," *Phys. Rev. Lett.*, vol. 88, p. 167902, 2002.