

# Rate Compatible Protocol for Information Reconciliation: An application to QKD

David Elkouss, Jesus Martinez, Daniel Lanchó and Vicente Martín  
 Facultad de Informática, Universidad Politécnica de Madrid,  
 Campus de Montegancedo, 28660 Boadilla del Monte (Madrid), Spain,  
 e-mail:{delkouss, jmartinez, dlancho, vicente}@fi.upm.es

**Abstract**—Information Reconciliation is a mechanism that allows to weed out the discrepancies between two correlated variables. It is an essential component in every key agreement protocol where the key has to be transmitted through a noisy channel. The typical case is in the satellite scenario described by Maurer in the early 90's. Recently the need has arisen in relation with Quantum Key Distribution (QKD) protocols, where it is very important not to reveal unnecessary information in order to maximize the shared key length.

In this paper we present an information reconciliation protocol based on a rate compatible construction of Low Density Parity Check codes. Our protocol improves the efficiency of the reconciliation for the whole range of error rates in the discrete variable QKD context. Its adaptability together with its low interactivity makes it specially well suited for QKD reconciliation.

**Index Terms**—Reconciliation, low-density parity-check (LDPC) codes, puncturing, shortening, rate-compatible.

## I. INTRODUCTION

The general scenario for information reconciliation is one in which two parties have two sets of correlated data with some discrepancies between them. The situation is equivalent to transmit the data from one party to the other through a noisy channel, akin in the satellite scenario described by Maurer [1].

In a Quantum Key Distribution (QKD) protocol, errors are generated in the communications channel either by the interaction of the quantum information carrier with the environment, by imperfections in the QKD device or by an eavesdropper. The two parties participating in the communication, Alice and Bob, thus have two sets of correlated data from which a common set must be extracted. This problem has been previously subject to consideration [2], [3], [4], [5], [6]. It is a process known as key distillation, that requires a discussion carried over an authenticated classical channel. It is interactive in the sense that it needs communications through the channel. Since it can also be listened by an eavesdropper, it is important to minimize the amount of information that have to be transmitted in the reconciliation process. Any extra information limits the performance of the QKD implementation. In theory one could minimize the information leakage using a highly interactive protocol, but in practical applications this would lead to a prohibitively large communication overhead through the network, limiting also the effective keyrate.

It is in this scenario where modern Forward Error Correction (FEC) is an interesting solution. The idea is to make use of FEC's inherent advantage of requiring a single channel use

to reconcile the two sets. In [6] it was analyzed the use of a discrete number of Low-Density Parity-Check (LDPC) codes optimized for the binary symmetric channel. As a consequence the efficiency exhibited an staircase-like behaviour: each code was used within a range of error rates and the reconciliation efficiency was maximized only in the region close to the code's threshold.

In this work, we develop the idea of using LDPC codes optimized for the binary symmetric channel. We take these codes as an starting point and develop a rate compatible information reconciliation protocol with an efficiency close to optimal. In particular, the proposed protocol builds codes that minimize the exchanged information for error probabilities between 1% and 10%<sup>1</sup>, the expected values in real implementations of QKD systems.

This solution addresses the rate adaptation problem (open problem 2) from the recent review paper of Matsumoto [8] in which he lists the problems that an LDPC solution should overcome in order to compare advantageously to current interactive reconciliation protocols.

The paper is organised as follows: In Section II the main ideas are discussed. A new Information Reconciliation Protocol able to adapt to different channel parameters is presented and its asymptotic behavior discussed. In Section III the results of a practical implementation of the protocol are shown. In particular we have analyzed the rate compared to the optimal value and the reconciliation efficiency.

## II. RATE COMPATIBLE INFORMATION RECONCILIATION

### *Information Reconciliation*

Let  $X$  and  $Y$  be two of correlated variables belonging to Alice and Bob, and  $\mathbf{x}$  and  $\mathbf{y}$  their outcome strings, Information Reconciliation [2] is a mechanism that allows them to eliminate the discrepancies between  $\mathbf{x}$  and  $\mathbf{y}$  and agree on a string  $S(\mathbf{x})$  —with possibly  $S(\mathbf{x}) = \mathbf{x}$ .

The problem of information reconciliation can be seen as the source coding problem with side information (see Fig. 1). Thus, as shown by Slepian and Wolf [9], the minimum information  $I$  that Alice would have to send to Bob in order to help him reconcile  $Y$  and  $X$  is  $I_{opt} = H(X|Y)$ . Taking into account that real reconciliation will not be optimal, we

<sup>1</sup>The maximum error thresholds for extracting an absolute secret key in a QKD protocol is 11% [7].

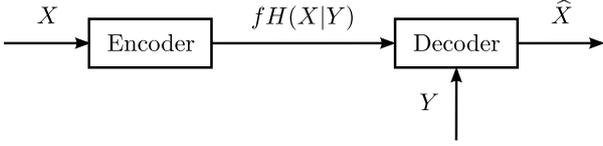


Fig. 1. Source coding with side information.

use a parameter  $f \geq 1$  as a quality figure for the reconciliation efficiency:

$$I_{real} = fH(X|Y) \geq I_{opt} \quad (1)$$

Here we will concentrate on binary variables, which apply to discrete variable QKD, although the ideas are directly applicable to other scenarios.

The most widely used protocol for information reconciliation in QKD is *Cascade* [2], because of its simplicity and good efficiency. *Cascade* is a highly interactive protocol that runs for a certain number of passes. In each pass, Alice and Bob both perform the same permutation on their respective strings, divide them in blocks of the same size and exchange the parities of the blocks. Whenever there is a mismatch they perform a dichotomic search to find an error, finding one usually means discovering more errors left in previous passes.

The main handicap of *Cascade* is its high interactivity. Buttler et al [10] proposed Winnow, a reconciliation protocol where instead of exchanging block parities, Alice and Bob exchange the syndrome of a Hamming code. Their protocol succeeded in reducing the interactivity but, in the error range of interest for QKD, the efficiency was worse than that of *Cascade*.

There has been further work on improving the efficiency of *Cascade*-like protocols. In [11] the block size is optimized, while in [12] the emphasis is put on minimizing the information sent to correct one error on each pass.

### Definitions

LDPC codes were introduced by Gallager in the early 60's [13]. They are linear codes with a sparse parity check matrix.

A family of LDPC codes is defined by two *generating polynomials* [14],  $\lambda(x)$  and  $\rho(x)$ :

$$\lambda(x) = \sum_{i=2}^{d_{smax}} \lambda_i x^{i-1}; \quad \rho(x) = \sum_{j=2}^{d_{cmax}} \rho_j x^{j-1} \quad (2)$$

where  $\lambda(x)$  and  $\rho(x)$  define degree distributions.  $\lambda_i$  and  $\rho_i$  indicate the proportion (normalized to 1) of edges connected to symbol and check nodes of degree  $i$ , respectively. The rate  $R_0$  of the family of LDPC codes is defined as:

$$R_0 = 1 - \frac{\sum_i \lambda_i / i}{\sum_j \rho_j / j} \quad (3)$$

Two common strategies to adapt the rate to the channel parameters are puncturing and shortening [15]. Puncturing

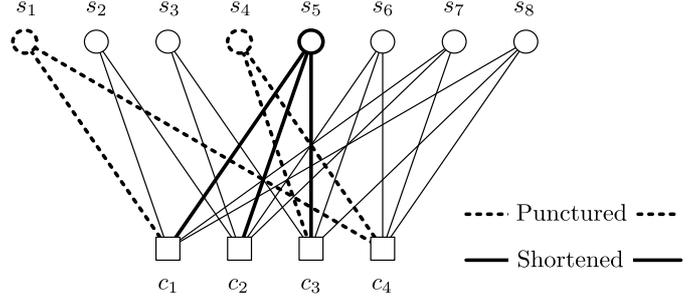


Fig. 2. Example of puncturing and shortening applied to a code represented by a Tanner graph. The rate of the original code is  $R = (n - m)/n = (8 - 4)/8 = 1/2$ . After puncturing two symbol nodes (indicated in the graph with dashed lines) the new rate is increased to  $R = (8 - 4)/(8 - 2) = 2/3$ . Shortening one symbol of the original code (indicated with thick solid lines) leads to a new rate of  $R = ((8 - 1) - 4)/(8 - 1) = 3/7$ . Puncturing two symbols and shortening one the original code leads to a rate of  $R = ((8 - 1) - 4)/(8 - 2 - 1) = 3/5$ .

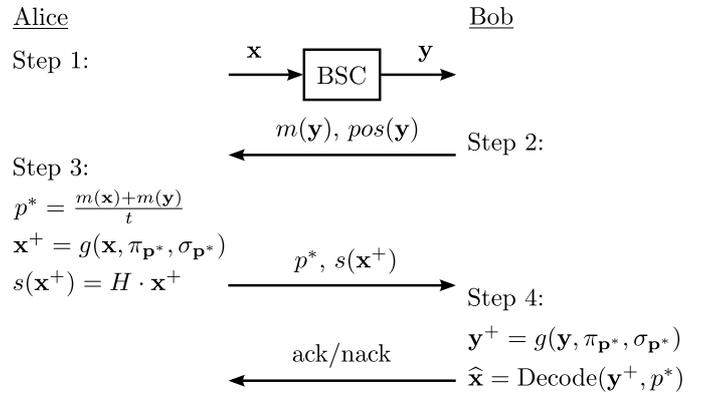


Fig. 3. Protocol sequence diagram.

means deleting a predefined set of  $p$  symbols from each word, converting a  $[n, k]$  code into a  $[n - p, k]$  code. Shortening means deleting a set of  $s$  symbols from the encoding process, converting a  $[n, k]$  code into a  $[n - s, k - s]$  code. Both processes allow to modulate the rate of the code as:

$$R = \frac{R_0 - \sigma}{1 - \pi - \sigma} = \frac{k - s}{n - p - s} \quad (4)$$

where  $\pi$  and  $\sigma$  represent the ratios of information punctured and shortened respectively, and  $R_0$  is the rate of the initial code (see Fig. 2 for an example).

### The protocol

Standard puncturing and shortening need an a priori knowledge about the channel in order to adapt the rate. The Bit Error Rate (BER) in the case of QKD protocols is an a priori unknown value, hence it is important to be able to construct codes that can adapt to the varying BER values that might appear during a QKD transmission. In order to cope with this, we propose an inverse puncturing and shortening protocol, that is performed after the distribution of the correlated variables.

The protocol assumes the existence of a shared pool of codes of length  $n$ , adjusted for different rates. Depending

on the range of crossover probabilities to be corrected, a parameter  $\delta$  is chosen to set the proportion of bits to be either shortened ( $\sigma$ ) or punctured ( $\pi$ ;  $\delta = \pi + \sigma$ ).  $\delta$  defines the achievable rates,  $R$ , through:

$$\frac{R_0 - \delta}{1 - \delta} \leq R \leq \frac{R_0}{1 - \delta} \quad (5)$$

with  $R_0$  being the rate of the code selected from the pool. For an  $[n, k]$  code this would mean  $n \cdot \pi$  bits punctured,  $n \cdot \sigma$  bits shortened and  $n \cdot (1 - \delta)$  bits transmitted over the BSC (see Fig. 4). The number of symbols not to be sent is  $d = \lfloor \delta \cdot n \rfloor$ .

The protocol goes through the following steps:

*Step 1:* Alice sends to Bob a message  $\mathbf{x}$ , an instance of variable  $X$ , of size  $\ell = n - d$  through a BSC of crossover probability  $p$  (or a black box behaving as such). Bob receives the correlated message,  $\mathbf{y}$ .

*Step 2:* Bob chooses randomly  $t$  bits of  $\mathbf{y}$ ,  $m(\mathbf{y})$ , and sends them and their positions,  $pos(\mathbf{y})$ , to Alice.

*Step 3:* Using  $pos(\mathbf{y})$ , Alice extracts  $m(\mathbf{x})$  and estimates the crossover probability:

$$p^* = \frac{m(\mathbf{x}) + m(\mathbf{y})}{t} \quad (6)$$

Once Alice has estimated  $p^*$ , she knows the theoretical rate for a punctured and shortened code able to correct the string. Now she must decide what is the optimal rate corresponding to the efficiency of the code she is using:  $R = 1 - f(p^*)h(p^*)$ ; where  $h$  is the binary entropy function and  $f$  the efficiency (e.g. Tab. I). Then she can derive optimal values for  $s$  and  $p$ :

$$\begin{aligned} s &= \lceil (R_0 - R(1 - d/n)) \cdot n \rceil \\ p &= d - s \end{aligned} \quad (7)$$

Alice creates now a string  $\mathbf{x}^+ = g(\mathbf{x}, \sigma_{\mathbf{p}^*}, \pi_{\mathbf{p}^*})$  of size  $n$ . The function  $g$  defines which  $n - d$  positions take the values of string  $\mathbf{x}$ , the  $p$  positions that take random values, and the  $s$  positions that take values known by Alice and Bob. The set of  $n - d$  positions, the set of  $p$  positions and the set of  $s$  positions and their values come from a synchronized pseudo-random generator. She then sends  $s(\mathbf{x}^+)$ , the syndrome of  $\mathbf{x}^+$ , to Bob as well as the estimated crossover probability  $p^*$ . The information gained by Eve with the side information (e.g. the syndrome) is discussed in [5].

*Step 4:* Bob can reproduce Alice's estimation of the optimal rate  $R$ , the positions of the  $p$  punctured bits, and the positions and values of the  $s$  shortened bits: he can create the corresponding string  $\mathbf{y}^+ = g(\mathbf{y}, \sigma_{\mathbf{p}^*}, \pi_{\mathbf{p}^*})$ . Bob should now be able to decode Alice's codeword with high probability, as the rate has been adapted to the channel crossover probability. He finally sends an acknowledge to Alice to indicate if he successfully recovered  $\mathbf{x}^+$ .

*Example:* Calculation of  $s$  and  $p$  for step 3. Alice and Bob use a  $[10^6, 5 \times 10^5]$  code,  $d = 10^5$ , and they have found out that the efficiency of their reconciliation behaves as  $f(p) = 1.1 + |p - 0.1|$ . When Alice estimates the discrepancy, she finds that  $p^* = 0.08$ . If the code were optimal, it would have

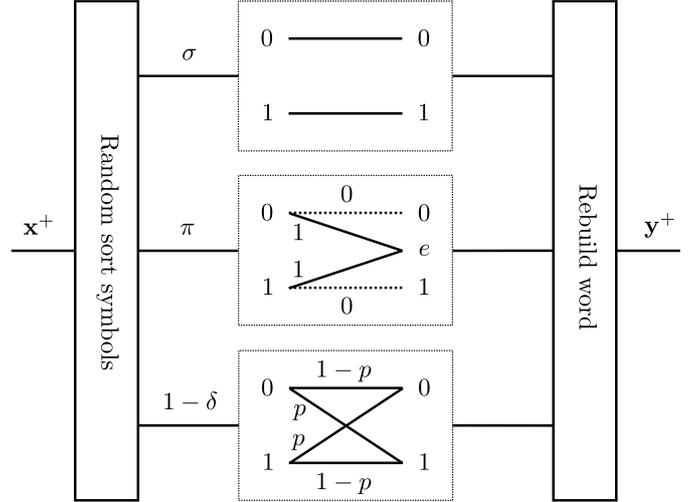


Fig. 4. Channel model. The protocol described can be interpreted as a communication through three channels with different probabilities: a noiseless channel with probability  $\sigma$ , a BEC(1) with probability  $\pi$ , and a BSC( $p$ ) with probability  $1 - \delta$ .

been designed with a rate  $R = 1 - f(0.08)h(0.08) = 1 - (1.12)(0.402) = 0.55$ . Then she obtains  $s = 2.25 \times 10^5$ , and  $p = 2.75 \times 10^5$ .

In the case in which the protocol is used to reconcile secret keys, several modifications have to be done. In step 1 the size should be increased by  $t$ ,  $\ell = n - d + t$ . In step 2, Bob should discard from his string,  $\mathbf{x}$ , the  $t$  bits that have been published. Finally, in step 3, Alice should also discard the  $t$  published bits from hers.

#### Performance analysis

We are first interested in the range of rates in which the protocol can be used and the expected efficiency if the codes are long enough. The threshold value of a code is calculated using the density evolution algorithm [14], and in particular we have implemented the discretized version of Chung et al [16]. The equation used to track the evolution of the density function is:

$$p_u^{(l+1)} = \rho(p_{u_0} * \lambda(p_u^{(l)})) \quad (8)$$

where  $p_u^{(l)}$  is the probability mass function at the symbols during iteration  $l$ , and  $p_{u_0}$  is the initial message density distribution, which in our case is:

$$p_{u_0}(x) = (1 - \delta)p_{u_0}^{\text{BSC}}(x) + \pi\Delta_0(x) + \sigma\Delta_\infty(x) \quad (9)$$

where  $p_{u_0}^{\text{BSC}}(x) = p\Delta_{-\log \frac{1-p}{1-p}}(x) + (1-p)\Delta_{-\log \frac{1-p}{1-p}}(x)$ , and  $\Delta_t(x) = \delta_{\text{dirac}}(x - t)$ .

On Fig. 5 we track the evolution of the threshold for the code with rate one half in [6], it can be observed how different values of  $\delta$  offer a tradeoff between the range of rates achievable and the efficiency.

In [14] it is presented a condition for decoding stability:

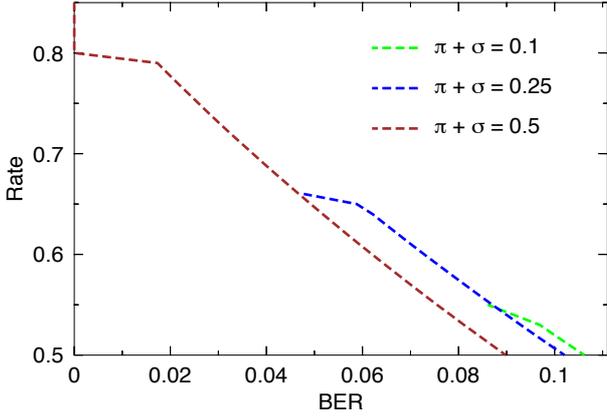


Fig. 5. Theoretical threshold.

$$\lambda'(0)\rho'(1) < \frac{1}{e^{-r}} \quad (10)$$

where  $e^{-r}$  is defined as:

$$e^{-r} = \int_{\mathcal{R}} p_{u_0}(x) e^{-x/2} dx \quad (11)$$

operating:

$$\lambda_2 < \frac{1}{(2\sqrt{p(1-p)}(1-\delta) + \pi)\rho'(1)} \quad (12)$$

which imposes a limitation when choosing a code: it has to be stable for the whole range of rates in which it will be used. A code with  $\lambda_2$  close to the stability limit for  $R_0$  can become unstable for high values of  $\pi$ .

### III. SIMULATION RESULTS

In order to understand the behavior of the protocol described in section II, we analyze the rate compared to the optimal value.

The family of LDPC codes used in our simulations have been obtained from [6] and the Tanner graphs have been constructed using a modified Progressive Edge-Growth (PEG) algorithm [17]. This improved PEG construction is based on the original [18], but it also takes into account  $\rho(x)$ , the check distribution polynomial. We have used a single code of length  $n = 200,000$ , a reasonable lower bound of the expected length in QKD transmission. Bigger  $n$  values would improve the performance of the protocol (by increasing the reconciliation efficiency). The rate is one half, that allows to cover all range of expected BERs. Simulations have been done with an LDPC decoder based on belief propagation, with a maximum number of 2000 iterations per simulation. The LDPC decoder has been modified to work with puncturing and shortening, adding two new log-likelihood ratios for the initialization of puncturing,  $\gamma_p = 0$ , and shortening,  $\gamma_s = \infty$ , respectively. The points in the different figures have  $p_{\text{bit}} < 10^{-6}$ .

In Fig. 6 we present the maximum BER reached over a BSC with the rates going from  $R = 0.5$  to 0.7 using different

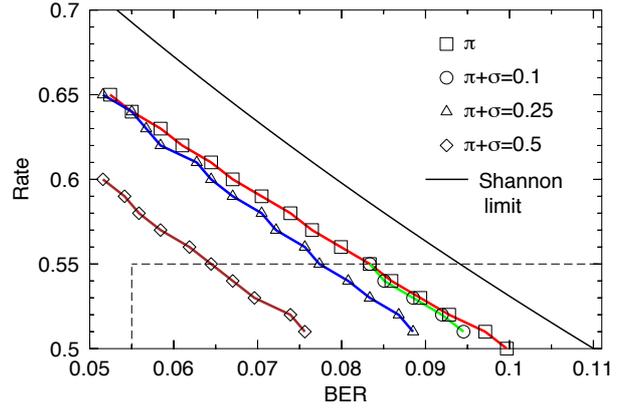


Fig. 6. Rate achieved over a BSC with  $\delta \in \{0.1, 0.25, 0.5\}$ .

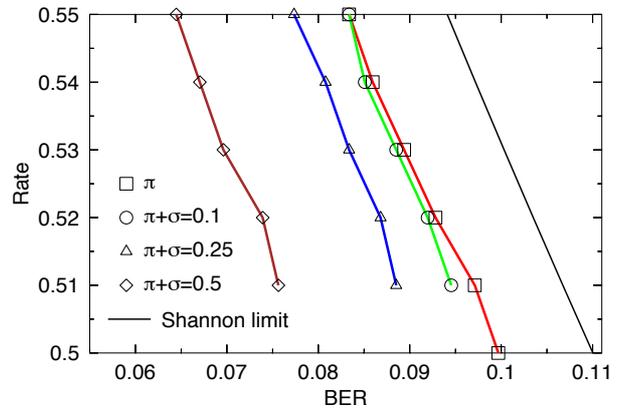


Fig. 7. Enlarged figure of the portion marked with a dashed line in Fig. 6.

values of the  $\delta$  parameter to regulate the puncturing and shortening. The strong dependence of the rate with parameter  $\delta$  is clearly seen. This figure shows the rate achievable for  $\delta \in \{0.1, 0.25, 0.5\}$ , and it is compared with the rate achieved by the code in the case that it were only punctured and with the Shannon limit. These results highlight that, once the reconciliation problem has been characterized and it is known the range of possible error rates,  $\delta$  should be chosen as small as possible. If  $\delta$  is found to be too big, then it should be considered enlarging the pool with codes that cover different rates. This behaviour can be more clearly seen in the enlarged figure (Fig. 7) displaying the rate range from  $R = 0.5$  to 0.55. The minimum value of  $\delta$  that allows to cover the entire interval is  $\delta = 0.1$ . For this value the decoding performance is similar to [6]. However, with this protocol we are able to reconcile a continuum of crossover probabilities. For the other values of  $\delta$ , the performance is worse, however it should be noted that carefully choosing which symbols should be punctured and which ones shortened could improve on these results [19], [20], [21].

Looking at Table I we can see the effect of the protocol on the efficiency of the reconciliation. When close enough to  $R_0$  it is close to one, and for small enough  $\delta$  values it remains close to one for the whole set of rates, which is not the case

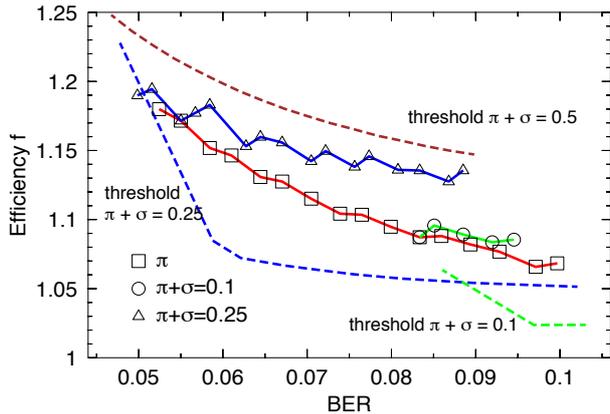


Fig. 8. Reconciliation efficiency calculated from Eq. 1.

TABLE I  
EFFICIENCY CALCULATED FROM EQ. 1.

$R^a$	$\delta = 0.1$		$\delta = 0.25$		$\delta = 0.5$	
	BER <sup>b</sup>	$f^c$	BER	$f$	BER	$f$
0.51	0.0945	1.0855	0.0885	1.1356	0.0756	1.2675
0.52	0.092	1.0836	0.0868	1.1276	0.0739	1.262
0.53	0.0885	1.0892	0.0834	1.1355	0.0696	1.2895
0.54	0.0851	1.0957	0.0808	1.136	0.067	1.2966
0.55	0.0834	1.0877	0.0773	1.1457	0.0645	1.3048
0.56			0.0756	1.1382	0.0619	1.314
0.57			0.0722	1.1496	0.0584	1.3386
0.58			0.0705	1.1423	0.0559	1.3513
0.59			0.067	1.1557	0.0541	1.3659
0.6			0.0645	1.1598	0.0516	1.3651
0.61			0.0627	1.1531		
0.62			0.0584	1.183		
0.63			0.0567	1.1772		
0.64			0.055	1.1715		
0.65			0.0516	1.1945		

<sup>a</sup>Rate after puncturing and shortening.

<sup>b</sup>Maximum bit error rate corrected.

<sup>c</sup>Corresponding efficiency for random puncturing and shortening.

for the higher  $\delta$  values as expected by the thresholds found in Fig. 5.

#### IV. CONCLUSION

We have demonstrated how to adapt an LDPC code for rate compatibility. The capability to adapt to different error rates while minimizing the amount of published information is an important feature for QKD key reconciliation. The present protocol allows to reach efficiencies close to one while limiting the information leakage and having the important practical advantage of low interactivity.

Future work will concentrate on the optimization of the puncturing and shortening processes, now done randomly.

#### ACKNOWLEDGMENT

This work has been partially supported by grant UPM/CAM Q061005127.

The authors acknowledge the computer resources and assistance provided by Centro de Supercomputación y Visualización de Madrid (CeSViMa).

#### REFERENCES

- [1] U. M. Maurer, "Secret key agreement by public discussion from common information," *Information Theory, IEEE Transactions on*, vol. 39, pp. 733–742, 1993.
- [2] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *Eurocrypt'93, Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, ser. Lecture Notes in Computer Science, vol. 765. Springer-Verlag, 1994, pp. 410–423.
- [3] S. Watanabe, R. Matsumoto, T. Uyematsu, and Y. Kawano, "Key rate of quantum key distribution with hashed two-way classical communication," in *Information Theory, IEEE International Symposium on*, Jun. 2007, pp. 2601–2605.
- [4] S. Watanabe, R. Matsumoto, and T. Uyematsu, "Tomography increases key rates of quantum-key-distribution protocols," *Phys. Rev. A*, vol. 78, no. 4, p. 042316, 2008.
- [5] A. Leverrier, R. Alléaume, J. Boutros, G. Zémor, and P. Grangier, "Multidimensional reconciliation for continuous-variable quantum key distribution," *Phys. Rev. A*, vol. 77, no. 4, pp. 042325–+, Apr. 2008.
- [6] D. Elkouss, A. Leverrier, R. Alleaume, and J. J. Boutros, "Efficient reconciliation protocol for discrete-variable quantum key distribution," in *Information Theory, 2009 IEEE International Symposium on*, Jul. 2009, pp. 1879–1883.
- [7] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Rev. of Mod. Phys.*, vol. 74, no. 1, pp. 145–195, Mar. 2002.
- [8] R. Matsumoto, "Problems in application of ldpc codes to information reconciliation in quantum key distribution protocols," 2009. [Online]. Available: arXiv.org:0908.2042
- [9] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *Information Theory, IEEE Transactions on*, vol. 19, no. 4, pp. 471–480, Jul. 1973.
- [10] W. Buttler, S. Lamoreaux, J. Torgerson, G. Nickel, C. Donahue, and C. Peterson, "Fast, efficient error reconciliation for quantum cryptography," *Phys. Rev. A*, vol. 67, no. 5, pp. 052303–+, May 2003.
- [11] 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.
- [12] S. Liu, H. C. V. Tilborg, and M. V. Dijk, "A practical protocol for advantage distillation and information reconciliation," *Des. Codes Cryptography*, vol. 30, no. 1, pp. 39–62, 2003.
- [13] R. G. Gallager, *Low-density parity-check codes*. MIT Press, Cambridge., 1963.
- [14] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [15] T. Tian and C. R. Jones, "Construction of rate-compatible ldpc codes utilizing information shortening and parity puncturing," *EURASIP J. Wirel. Commun. Netw.*, vol. 2005, no. 5, pp. 789–795, 2005.
- [16] S.-Y. Chung, J. Forney, G.D., T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *Communications Letters, IEEE*, vol. 5, no. 2, pp. 58–60, Feb. 2001.
- [17] J. Martinez, D. Elkouss, D. Lanchos, and V. Martin, "An easy and improved progressive edge growth construction," *To be published elsewhere*.
- [18] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *Information Theory, IEEE Transactions on*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [19] J. Ha, J. Kim, and S. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *Information Theory, IEEE Transactions on*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.
- [20] G. Richter, S. Stiglmayr, and M. Bossert, "Optimized asymptotic puncturing distributions for different ldpc code constructions," in *Information Theory, IEEE International Symposium on*, Jul. 2006, pp. 831–835.
- [21] D. Klinc, J. Ha, and S. McLaughlin, "Optimized puncturing and shortening distributions for nonbinary ldpc codes over the binary erasure channel," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, Sep. 2008, pp. 1053–1058.