CASCADE as Correlation Enhancer for CVQKD Protocols

Micael Andrade Souza, Francisco Marcos de Assis, Bruno Barbosa Albert and Leocarlos Bezerra da Silva Lima

***Abstract*— We propose a secret key reconciliation approach for CVQKD information reconciliation based on CASCADE protocol, which works as a correlation enhancer, in order to ensure a better utilization of binary quantized sequences from CVQKD protocols. We make a probabilistic analysis of the initial block size in order to control the expected value of the corrected errors in the first step of CASCADE and to bound the amount of information leaked. Our method achieved significant enhancements on binary sequences with $BER < 0.35$ and improved sequences with $BER > 0.35$ to became eligible for information reconciliation.**

***Keywords*— CVQKD, Information Reconciliation, CASCADE.**

I. Introduction

Quantum key distribution (QKD) was proposed by Bennet and Brassard [1] as a solution for distributing key elements securely over a quantum channel. QKD systems aims to attend to a necessity of two legitimate parts (Alice and Bob) to establish a secret random key given no previous shared information. The intruder (Eve) has an unlimited computational power and has knowledge of each protocol step. Unconditional security is based on the no-cloning theorem and the uncertainty principle [2]–[4]. This is a major advantage over classical cryptosystems, whose security relies on large prime numbers factorization complexity as its advantages over eavesdropping, putting itself in a vulnerable spot under quantum computing evolution.

The setups for QKD have been divided between the Discrete-Variable (DVQKD) and Continuous-Variable (CVQKD) ones, where the CVQKD protocols, which aims for secret key rates and transmission distances to be at last comparable to DVQKD protocols, uses usual optical telecommunications setups and encodes information on continuous modulation of electromagnetic field of coherent states. This ability to use common optical telecommunication equipment (Optical Multiplexers, amplitude and phase modulators, etc.) enables large-scale deployment, as in urban networks [5], and does not require operating conditions as controlled as single photon detector used in DVQKD protocols.

Despite these advantages, as CVQKD continuous modulate the data (most setups follows the GG02 protocol [6], where Gaussian random variables realizations are encoded on electromagnetic field coherent modulation), the transmitted and received measured states presents also continuous valued discrepancies. Thus, one should choose (1) to correct the continuous values itself, leading to a high noise sensibility, or (2) to quantize the numerical data, performing binary error correction over the quantized values, which is the most suitable solution [7]. The procedure responsible for performing error correction is the information reconciliation (IR) protocol.

Micael Andrade Souza, Francisco Marcos de Assis, Bruno Barbosa Albert and Leocarlos Bezerra da Silva Lima, Center of Electrical Engineering and Informatics, Federal University of Campina Grande (UFCG), Campina Grande-PB, Brazil. E-mails: micael.souza@ee.ufcg.edu.br, fmarcos@dee.ufcg.edu.br, albert@dee.ufcg.edu.br and leocarlos@dee.ufcg.edu.br.

The first IR protocol proposed to be used with QKD was CASCADE [8], under DVQKD protocols [1], [9], [10], whose QBER (Quantum Bit Error Rate) doesn't exceed 15% [1], [11]. On CVQKD, the quantization procedure adds complexity to the system, where a good methodology needs to be applied in order to acquire as many bits as possible from the continuous measured values. Some methods have been proposed as a solution to the quantization problem as multidimentional reconciliation [12], [13] and the SEC protocol [7], [14]. However, in most cases, they will gain gain two bits from each key elements (coherent pulse transmitted from Alice to Bob) due to the high bit error rates found. The complexity of extracting bits from the continuous variable is also high. Recently, a Binary Expansion Protocol has been proposed, relying on information theory arguments of probability distributions [15] presenting itself as a strong option over SEC reconciliation due to its random variable optimal compression rate.

CASCADE has been well established aon DVQKD scenarios but it hasn't take a spot on reconciling continuous variable generated keys, given its inherent high bit error rates resulting of quantization procedures on CVQKD. In this work we propose a secret key reconciliation approach for CVQKD based on CASCADE protocol, which works as a correlation enhancer, in order to ensure a better utilization of binary quantized sequences from CVQKD protocols. After binary expansion, the raw key's binary version owned by Alice and Bob will present high bit error rates as the expansion extracts more bits from the continuous values. A proper manipulation of CASCADE main parameters, as the initial block size, may produce an appropriated trade-off between error correction and information leakage. It is observed that the proposed application of the CASCADE will ease the utilization of LDPC codes with shorter lengths in order to complete the reconciliation step.

The paper is organized as follow: Section II revisits the CASCADE protocol, emphasizing the initial block size designing method and the leaked information estimate; Section III introduces the initial block size modification proposed and its implications on information leakage and error correction. Section IV presents the Binary Expansion protocol and the results of combining it with the proposed CASCADE modification. Section V presents the final considerations.

II. CASCADE

CASCADE is a practical reconciliation protocol proposed in [8] with the purpose of interactively correct errors between two binary sequences: the raw keys owned by Alice and Bob. The main idea is to divide Alice's and Bob's binary sequences into smaller blocks, compare its parities (module 2 sum) and apply a correction function over each mismatching parity block. CASCADE became a well established information reconciliation protocol, probably the most widely used IR protocol [11], being able to reconcile binary sequences while leaks an amount of information close to the theoretical limit over a BSC (Binary Symmetric Channel) for a transition probability up to 0.15. In CASCADE, the suitable error correction function is BINARY [16], which performs a binary search to find an error.

A. BINARY Description

For two non identical binary sequences A and B (owned by Alice and Bob, respectively), an error may be corrected by exchanging its parities if there is an odd number of errors. Therefore, if the sequences A and B differ an odd number of positions and Bob is the one correcting his sequence, BINARY will perform the following procedure:

1) Alice sends to Bob the first half parity of her sequence;
2) Bob compares the received parity with his sequence first half parity and determines if the odd number of error is located in the first or second half;
3) After located in which half is the odd errors, step 1 and 2 are repeated splitting and comparing parities until an error is located.

Example 1: Let $\{a_1, a_2, \cdots, a_8\}$ and $\{b_1, b_2, \cdots, b_8\}$ be two binary sequences and the only different bit on in position 5, $a_5 \neq b_5$. The error correction will proceed as explained above. First, it will obtain the parity ($\otimes$) of the sequence first half, $\{b_1, b_2, b_3, b_4\}$, and will detect no parity mismatch. The odd number of errors must be in the second half, than b_5 to b_8 is now under analysis: the parity of $\{b_5, b_6\}$ is compared, resulting on a mismatch and an error is detected. Finally, b_5 is disclosed and the error is located at position five.

B. CASCADE Description

The protocol proceeds in several steps. First, Alice and Bob must decide the number of steps and the initial block size (k_1). Generally, let $A = A_1, A_2, \cdots, A_n$ and $B = B_1, B_2, \cdots, B_n$ ($A_i, B_i \in \{0, 1\}$) be the binary sequences owned by Alice and Bob, respectively. In the first step, both parts split their sequences into $\lceil \frac{n}{k_1} \rceil$ blocks of length k_1, where the block v in step 1 is defined by $K_v^1 = \{l : (v-1)k_1 < l \leq vk_1\}$, $v = 1, \cdots, \lceil \frac{n}{k_1} \rceil$[1]. Then, for each block, they calculate the parity and whenever there's a parity mismatch, one error may be corrected by BINARY.

At the end of first step, all blocks of length k_1 has an even number of errors, possibly zero. Therefore, for steps $i > 1$, Alice and Bob choose a k_i and a random function $f_i : [1 \cdots n \to [1 \cdots \lceil \frac{n}{k_1} \rceil]]$ representing a permutation at step i. With that, the block j at step i is defined by $K_j^i = \{l : f_i(l) = j\}$[2]. Now, Alice and Bob repeat the process of parity exchange for each block K_j^i and perform BINARY to correct an error. For now on, before advance to step $i+1$, when an error located at position l in the block K_j^i is corrected in a step $i > 1$, it means that each step $u < i$ contains a block with an odd parity. Therefore, a set $\mathcal{K}$ may be formed by the blocks K_v^u, with $1 \leq u < i$, containing the bit l, and the protocol shall choose the smallest block in $\mathcal{K}$ to correct another error. Let l' be the corrected error position when the smallest block on $\mathcal{K}$ was chosen. Another set $\mathcal{B}$ is created formed by the blocks from step 1 to i containing l'. At this point, the set $\mathcal{K}' = \mathcal{K} \cup \mathcal{B} \setminus \mathcal{K} \cap \mathcal{B}$ is constructed and possess all odd parity blocks from steps 1 to i. The protocol proceeds choosing the smallest block in $\mathcal{K}'$ and updating it until $\mathcal{K}' = \emptyset$ and conclude step i.

C. Initial Block Size Design and Information Leakage

To explain the original k_1 design method, first lets define a binomial random variable $X \sim \text{Bin}(k_1, p_e)$ where k_1 represents its length (number of trials) and p_e is the success probability. The variable X models the errors between Alice's and Bob's sequences, being p_e the error probability (or, in a communication point of view, a BSC channel transition probability), and a success at position l represents the occurrence of an error at this position.

After CASCADE first step, the probability of remaining $2j$ errors on a certain block of length k_1 is (1) the probability of this blocks already have $2j$ initial errors (no error correction would be applied) or (2) the probability of this block have $2j+1$ initial errors (an error correction with BINARY would fix one error). Than, the total probability $\delta(j)$ of a block of length k_1 to keep $2j$ errors after the first step could be expressed as

$$\delta_1(j) = P[X = 2j] + P[X = 2j + 1]. \tag{1}$$

So, the expected value of remaining errors in a block of length k_1 after the first step is

$$E_1 = \sum_{j=1}^{\lfloor \frac{k_1}{2} \rfloor} 2j\delta_1(j) = k_1 p - \frac{1 - (1-2p)^{k_1}}{2}. \tag{2}$$

In [8] the authors of CASCADE [8] made the choice of the initial block size as the greatest integer satisfying two conditions, as stated in Equations (3) and (4):

$$E_1 \leq 0.346, \tag{3}$$

$$\sum_{l=j+1}^{\lfloor \frac{k_1}{2} \rfloor} \delta_1(l) \leq \frac{1}{4}\delta_1(j) \to P[X > 2j] \leq \frac{1}{4}P[X = 2j]. \tag{4}$$

The first condition imposes a small amount of errors when step one is finished and the second one demand that, for

[1]For example, in a sequence of length 16 and $k_1 = 4$, $v = \{1, 2, 3, 4\}$ and the blocks K_i^1 will assign the positions $K_1^1 = \{1, 2, 3, 4\}$, $K_2^1 = \{5, 6, 7, 8\}$ and so on.

[2]Given the f_i function, the block K_j^i is formed by all domain points led to the image point j.

TABLE I
CASCADE BENCHMARK

BER	k_1	$I(4)$	$\hat{I}(4)$	$k_1\mathcal{H}(p_e)$
0.01	73	6.81	6.67	5.89
0.05	14	4.64	4.60	4.01
0.10	7	3.99	3.81	3.28
0.15	5	4.125	3.984	3.049
0.20	4	3.509	3.362	2.888
0.25	3	3.422	3.221	2.434
0.30	3	3.692	3.598	2.644

a given j, the probability of remaining $2(j+1), \cdots, \lfloor \frac{k_1}{2} \rfloor$ errors after step 1 be smaller than the probability of remaining $2j$ errors. Both conditions guarantee a fast reconciliation procedure (few steps will be needed).

With k_1 chosen as stated above and $k_i = 2 \cdot k_{i-1}$, the information leaked (bits per block of size k_1) after w steps may be upper bounded by [8]:

$$I(w) \leq 2 + \frac{1-(1-2p_e)^{k_1}}{2}\lceil log(k_1)\rceil + \sum_{l=2}^{i} \frac{E_1}{2^{l-1}}\lceil log(k_1)\rceil, \tag{5}$$

Table I gives the allowed largest values of k_1, the amount of leaked information after step 4 ($I(4)$) according to Equation (5) and the simulation results after step 4 ($\hat{I}(4)$) according to the original implementation.

The values of k_1, $I(4)$ and $\hat{I}(4)$ are the same as in [8], for BER ≤ 0.15. The scenarios where $0.15 <$ BER ≤ 0.30 were simulated under the same k_1 choice conditions, revealing CASCADE behaviour under high BER conditions. In the range $0.15 <$ BER ≤ 0.25, the secret key is left with less than one secrecy bit per block, which implies an excessive amount of information leakage and, for BER ≥ 0.25, the protocol expose more information bits than the block length, i. e., $\hat{I}(4) > k_1$.

III. Initial Block Size Modification

This Section describes the main contributions of this paper. In the previous Section, it was shown that the protocol tends to leak an extremely high amount of information for BER $>$ 0.15, specially for BER $\geq$ 0.25. Since no application of CASCADE has been proposed to operate in such BER levels [11], [17], [18], commonly being analyzed for BER $<$ 0.15, it's proposed a usage of CASCADE as correlation improover by correcting some amount of error between the sequences while leaks a controlled amount of information during several reconciliation steps.

The idea is to adjust E_1 by an arbitrary parameter ε, which will indicates the fraction of errors that remains after the first step. The subsequent steps shall proceed unmodified, doubling k_1 in each step advanced and performing recursive searches.

The proposed formulation demands that

$$E_1 \leq k_1 p_e \varepsilon, \tag{6}$$

and using Equation (2) in Equation (6), it is obtained

$$k_1 p_e(1-\varepsilon) - \frac{1-(1-2p_e)^{k_1}}{2} \leq 0. \tag{7}$$

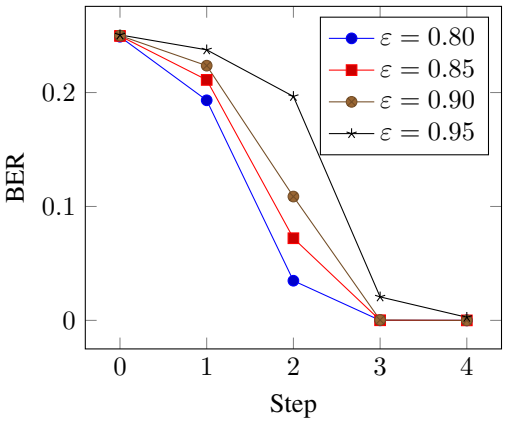

Fig. 1. BER improvement for different values of ε and $p_e = 0.25$

Therefore, for a fixed error rate and some ε previously specified, there will be a finite set of integers that satisfies Equation (7), which are the possible values of k_1. Within this set, the bigger value may be used.

Several simulations were performed in order to model the behavior of the proposed modification under high BER scenarios, once E_1 controls the error correction at the first step but tells nothing about information leakage on further steps. First, some information reconciliation simulations were performed for BER $= 0.25$ and ε equal to 0.80, 0.85, 0.90, 0.95, shown in Figure 1. Second, two step reconciliation simulations for BER equal to 0.25, 0.35 and 0.45 and for ε assuming the same values of Figure 1 were realized and the information leakage estimated at each step, once Figure 1 revealed that at the third step almost all errors have been fixed, meaning excessive information leakage. The results are presented in Table II, where its possible to see that information reconciliation proceeds in the first step as expected, correcting errors proportionally to the value of ε, but for some scenarios, a two step reconciliation leaks more information than the theoretical bound given in Equation (5). It is due to the modification made on k_1, whose design method doesn't guarantee that the expected number of errors after each pass decreases exponentially, which is a fundamental outcome for achievement of Equation (5), as explained in [8].

Besides the information leakage, another way of analyzing the error correction performed is to observe it as a step by step channel improvement, as described in Figure 2. For either direct or reverse reconciliation, Alice's or Bob's strings must be corrected meanwhile the other stays unmodified. We indicate the string S under correction after the i-th step as S^i. It is clear that S^i has less errors than S^{i-1}, so the bit error rate after each step is smaller than in the previous step. We remark that information reconciliation performs a correlation improvement and, in an information theory point of view, a channel capacity enhancement[3] (both terms will be used

[3]If the sequences S^i are understood as the result of a transmission through a $BSC(p_e)$, which has a very specified channel capacity [19], any error correcting code used for reconciliation is upper bounded by the $BSC(p_e)$ inherent channel capacity. Then, with the usage of CASCADE, the bit error rate between the sequences may decrease step by step, which implies on a channel capacity improvement.

TABLE II
INITIAL BLOCK SIZE MODIFICATION INFORMATION LEAKAGE AND CAPACITY ENHANCEMENT FOR THE FIRST AND SECOND STEPS.

BER	ε	k_1	$I(1)$	$\hat{I}(1)$	$I(2)$	$\hat{I}(2)$	ΔC_1	ΔC_2
0.25	0.80	9	3.99	2.98	7.49	9.80	0.11	0.60
	0.85	13	3.99	3.02	9.50	11.72	0.07	0.44
	0.90	19	4.49	3.46	15.12	15.86	0.05	0.32
	0.95	39	5,00	3.94	32.74	14.67	0.03	0.10
0.35	0.80	7	3.49	2.49	6.42	8.53	0.08	0.65
	0.85	9	3.99	2.98	2.29	11.67	0.06	0.50
	0.90	14	3.99	2.98	12.79	12.10	0.03	0.27
	0.95	28	4.49	3.48	27.74	13.23	0.02	0.09
0.45	0.80	5	3.49	2.49	6.12	8.21	0.06	0.69
	0.85	7	3.49	2.27	7.47	9.76	0.03	0.51
	0.90	11	3.99	2.99	12.89	13.69	0.02	0.26
	0.95	22	4.50	3.48	28.00	14.47	0.01	0.07

interchangeably). Than, the "channel capacity"implied by both sequences after the i-th step is denoted by C_i, C_0 is its capacity before reconciliation process, and the quantity

$$\Delta C_i = C_i - C_0 \tag{8}$$

is defined as the channel capacity enhancement. Table II last two columns gives the capacity enhancements achieved by the protocol. The values of C_0 for the simulated BER values are 0.188, 0.066 and 0.007 bits, respectively.

IV. RECONCILIATION OF CVQKD KEYS

A. Quantization Method

On continuous variable QKD, as Gaussian Modulated Coherent States (GMCS) protocols, the first step before start any error correction is to decide whether to perform a real valued error correction or to apply some quantization method in order to extract binary sequences from the coherent states [7], which has proved to be the most adopted option.

The Sliced Error Correction (SEC) has established itself through the years as the main error correcting solution for CVQKD protocols. It combines the design of S_m slicing functions, which will slice the interval $(-\infty, \infty)$ into m partitions to produce binary sequences (m indicates the number of bits extracted from each quantum state transmitted), with powerful LDPC codes to accomplish error correction for the deteriorated binary sequences obtained by the slicing functions.

A much easier solution is proposed in [15] where it is performed a binary expansion of continuous values that acts as a quantizer method, ensuring expansion bits interdependence based on the following well known Lemma [20]:

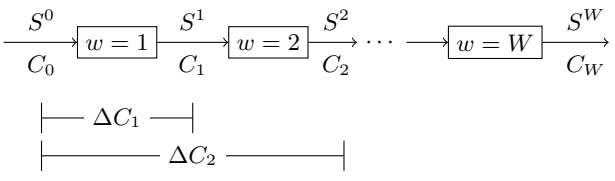

Fig. 2. CASCADE based channel improvement scheme.

Lemma 1: Let Y be a random variable with continuous probability distribution function $F(y)$. Let $U = F(Y)$ (i.e., U is a function of Y defined by its distribution function). Then U is uniformly distributed on $[0, 1]$.

A direct result of the Lemma above is that the continuous distribution function of a random variable X maps the raw key values directly into the interval $[0, 1]$ uniformly distributed. Moreover, with a binary expansion of the uniform distributed values, the bits will be pairwise independent and each one will be distributed as a Bernoulli variable with parameter $\frac{1}{2}$, resulting on a compressed representation of X. The expansion

$$x = 0.x_1x_2x_3\cdots x_l = \sum_{j=1}^{l} x_j 2^{-j} \tag{9}$$

of a $x \in [0, 1]$ has the format $0.F_1F_2\cdots F_l$ [20], each $F_i \in GF(2), 1 \leq i \leq l$, has equal outcome probability and l is the expansion order. The process to obtain a proper GMCS expanded values raw keys is the following:

1) For Alice's and Bob's raw key value, calculate $F(X)$ ($[0, 1]$ value);
2) Expand each value as in Equation (9);
3) Treat each bit of the r values of the raw key as a BSC channel.

Each raw key value is a Gaussian random variable realization and every realization will be expanded in l bits, as showed above, where a sequence of r realization of the Gaussian random variable is represented as a $l \times r$ matrix as in Equation (10).

$$(X_1, X_2, \cdots, X_r) = \begin{bmatrix} F_1^1 & F_1^1 & \cdots & F_r^1 \\ F_1^2 & F_2^2 & \cdots & F_r^2 \\ \vdots & \vdots & \ddots & \vdots \\ F_1^l & F_2^l & \cdots & F_r^l \end{bmatrix} \tag{10}$$

B. Simulation Results and Discussion

In order to simulate the quantum state transmission, the quantum channel is modeled as an Additive White Gaussian Noise Channel (AWGNC) since the quantum states are coherently modulated [21]. The performed experiments followed the methodology of Section IV-A, where a Gaussian random variable X is transmitted and the channel output Y is received. Obtaining a function $F(\cdot)$ for each transmitted and received value, the continuous values are expanded in the form of Equation (10), ensuring that each bit of expansion configures a BSC channel. Each channel in the expansion was treated individually, and simulations of reconciliation were performed using the proposed modification of CASCADE with $0.75 \leq \varepsilon \leq 0.95$. The experiment operated under a 5dB SNR and applied a four-bit binary expansion, resulting on sequences with BER values of $0.1637, 0.3538, 0.4369, 0.4708$ and channel capacity C_0 of $0.357, 0.062, 0.011, 0.002$, respectively.

For the information leakage, the first bit of expansion (F^1), with BER $= 0.1637$, presents a situation more compatible with the original CASCADE usage. The protocol ran two steps of reconciliation and the leaked information stood below the values of the corresponding k_1. In every value of ε the better

TABLE III
INFORMATION LEAKAGE COMPARISON OF THE RECONCILIATION FIRST TWO STEPS IN CVQKD KEYS BINARY EXPANDED

p_e	ε	k_1	$I(1)$	$\hat{I}(1)$	$I(2)$	$\hat{I}(2)$	C_1	C_2
0.1637	0.75	12	3.98	2.99	6.92	9.01	0.46	0.85
	0.80	15	3.99	2.99	7.90	10.18	0.44	0.79
	0.85	20	4.49	3.51	11.43	12.05	0.85	0.42
	0.90	30	4.50	3.53	15.52	13.22	0.39	0.56
	0.95	61	5.00	4.14	33.45	24.03	0.37	0.50
0.3538	0.75	5	3.49	2.49	5.40	7.10	0.18	0.86
	0.80	7	3.50	2.51	6.46	8.65	0.14	0.70
	0.85	9	4.00	3.00	9.36	11.73	0.12	0.55
	0.90	14	4.00	3.00	12.90	12.06	0.09	0.31
	0.95	28	4.50	3.51	28.01	13.65	0.07	0.15
0.4369	0.75	4	3.00	1.99	4.24	5.43	0.10	0.86
	0.80	5	3.50	2.49	6.02	8.06	0.07	0.72
	0.85	7	3.50	2.50	7.33	9.62	0.05	0.53
	0.90	11	4.00	3.00	12.61	13.29	0.03	0.29
	0.95	22	4.50	3.55	27.28	14.60	0.02	0.08
0.4768	0.75	4	3.00	1.99	4.40	5.75	0.06	0.81
	0.80	5	3.50	2.49	6.32	8.41	0.04	0.65
	0.85	6	3.50	2.49	7.04	9.46	0.03	0.58
	0.90	10	4.00	3.00	12.53	11.80	0.01	0.20
	0.95	20	4.50	3.49	27.09	13.49	0.00	0.05

result was for $\varepsilon = 0.75$ and two steps performed, where 9 bits per block of 12 bits were nearly leaked and the channel was improved with $\Delta C_2 = 0.502$, the highest in the category.

The second bit, F^2, with $\mathrm{BER} = 0.3538$, was very similar with the simulations presented in Section III ($\mathrm{BER} = 0.35$). The suitable values of ε for a two step reconciliation performed by CASCADE laid in the interval $[0.90, 0.95]$, as for $\varepsilon \leq 0.90$ the reconciliation leaked $\hat{I}(2) > k_1$, making the resulting key unusable. On the other hand, the results shown in Table III includes the $\varepsilon = 0.75$, setting k_1 to 5 bits and leaking 2.5 bits in the first step, while $\Delta C_1 = 0.12$. That's a much better option than use $\varepsilon = 0.95$, when $k_1 = 28$, $\hat{I}(2) = 13.658$ and $\Delta C_2 = 0.09$.

For F^3 and F^4, the bit error rates were extremely high ($\mathrm{BER} = 0.4369$ and $\mathrm{BER} = 0.4768$, respectively) letting ε assume values above 0.95 (possibly some value between 0.90 and 0.95 will result in a suitable information leakage). Both scenarios behaved similarly, leaking information in a close range, which implies that the reconciliation behaves similarly for bit error rates above 0.40. The best channel improvements and information leakage were obtained for one step of reconciliation and $\varepsilon = 0.75$, where $\hat{I}(1) = 1.99$ bits in both cases.

V. CONCLUSIONS

The reconciliation with CASCADE following the proposed modification of the initial block size and the Binary Expansion protocol behaved differently for the first two bits of expansion and for the third and fourth bits. As F^1 and F^2 were found in an usual error correction code operation, with a mutual information greater than 0.02, the proposed correlation improvement tended to decrease the effort made by the second error correction procedure (LDPC, for example), by reducing both its complexity and its computer demanded power. For F^3 and F^4 we had a different outcome, as they presented mutual information below the threshold of 0.02 bit, when they would originally be disclosed. Using the proposed method, Table III showed that it is possible to improve correlation between keys without compromising the entire information. Future works would contemplate an application with LDPC codes and further researches into the optimal values of ε .

REFERÊNCIAS

[1] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theoretical Computer Science*, vol. 560, pp. 7–11, dec 2014.

[2] P. W. Shor and J. Preskill, "Simple proof of security of the BB84 quantum key distribution protocol," *Physical Review Letters*, vol. 85, no. 2, pp. 441–444, jul 2000.

[3] H. K. Lo and H. F.Chau, "Unconditional Security Proof of Quantum Key Distribution Over Arbitrarily Long Distaces," *Science*, vol. 283, no. 5410, pp. 2050–2056, mar 1999.

[4] D. Mayers, "Unconditional Security in Quantum Criptography," *Journal of the ACM*, vol. 48, no. 3, pp. 351–406, mai 2001.

[5] E. O. Kiktenko, N. O. Pozhar, A. V. Duplinskiy, A. A. Kanapin, A. S. Sokolov, S. S. Vorobey, A. V. Miller, V. E. Ustimchik, M. N. Anufriev, A. S. Trushechkin, R. R. Yunusov, V. L. Kurochkin, Y. V. Kurochkin, A. K. Fedorov, "Demonstration of a quantum key distribution network in urban fibre-optic communication lines," *ArXiv*, arXiv:1705.07154 [quant-ph].

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

[7] G. V. Assche, J. Cardinal, and N. J. Cerf, "Reconciliation of a quantum-distributed gaussian key," *IEEE Transactions on Information Theory*, vol. 50, no. 2, pp. 394–400, Feb. 2004.

[8] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *Advances in Cryptology – EUROCRYPT '93*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 410–423.

[9] C. H. Bennett, F. Bessette, G. Brassard, L. Savail and J. Smolin, "Experimental Quantum Criptography," *Journal of Criptology*, vol. 5, no. 1, 1992.

[10] C. H. Bennett, "Quantum Criptography Using Any Two nonorthogonal States," *Physical Review Letters*, vol. 68, no. 21, pp. 3121–3124, mai 1992.

[11] J. Martinez-Mateo, C. Pacher, M. Peev, A. Ciurana, and V. Martin, "Demystifying the information reconciliation protocol cascade," *Quantum Info. Comput.*, vol. 15, no. 5-6, pp. 453–477, Apr. 2015.

[12] A. Leverrier, R. Alléaume, J. Boutros, G. Zémor, and P. Grangier, "Multidimensional reconciliation for a continuous-variable quantum key distribution," *Phys. Rev. A*, vol. 77, no. 4, p. 042325, Apr. 2008.

[13] P. Jouguet, S. Kunz-Jacques, and A. Leverrier, "Long-distance continuous-variable quantum key distribution with a gaussian modulation," *Phys. Rev. A*, vol. 84, no. 6, p. 062317, Dec. 2011.

[14] P. Jouguet, D. Elkouss, and S. Kunz-Jacques, "High-bit-rate continuous-variable quantum key distribution," *Phys. Rev. A*, vol. 90, no. 4, p. 042329, Oct. 2014.

[15] L. M. C. Araújo, F. M. Assis, and B. B. Albert, "Novo protocolo de reconciliação de chaves secretas geradas quanticamente utilizando códigos LDPC no sentido Slepian-Wolf," in *Simpósio Brasilieiro de Telecomunicações e Processamento de Sinais*, 2018.

[16] C. Kollmitzer and M. Pivk, Eds., *Applied Quantum Cryptography*. Springer Berlin Heidelberg, 2010.

[17] P. Bellot and M. D. Dang, "Bb84 implementation and computer reality," in *2009 IEEE-RIVF International Conference on Computing and Communication Technologies*, 2008, pp. 1–8.

[18] T. Brochmann Pedersen and M. Toyran, "High Performance Information Reconciliation for QKD with CASCADE," *ArXiv e-prints*, Jul. 2013.

[19] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, jul 1948.

[20] J. A. T. Thomas M. Cover, *Elements of Information Theory*. Wiley John + Sons, 2006.

[21] Z. Lu, L. Yu, K. Li, B. Liu, J. Lin, R. Jiao, and B. Yang, "Reverse reconciliation for continuous variable quantum key distribution," *Science China Physics, Mechanics and Astronomy*, vol. 53, no. 1, pp. 100–105, jan 2010.