LDPC Code Design Aspects for Physical-Layer Key Reconciliation

Nazia Islam, Oana Graur, Alexandra Filip,◇ and Werner Henkel

Jacobs University Bremen
Electrical Engineering and Computer Science
Bremen, Germany
Emails: {n.islam, o.graur, w.henkel}@jacobs-university.de, alexandra.filip@dlr.de

***Abstract*—In this work, we investigate a physical-layer key reconciliation protocol for a reciprocal, flat fading channel between two legitimate users. We consider the scenario when the *n* bits of the secret key are measured independently by Alice and Bob without a transmission over the channel. Due to reciprocity, the generated keys are identical except for noise at both ends. We assume Gaussian noise and ignore non-ideal behavior of circuitry and alike. Redundancy information required to reconciliate the key is transmitted from one legitimate user to the other. LDPC codes are employed for the reconciliation procedure. The main focus of this work lies in designing the code structure through density evolution for a multi-edge-type description.**

I. Introduction

In a wireless scenario with two legitimate users, Alice and Bob, and an eavesdropper Eve, properties of the channel can be used to provide security options to the legitimate users through generating shared secret keys. We consider a reciprocal channel between Alice and Bob, ideally ensuring identical amplitude and phase properties. The secret key is obtained directly via channel measurements, the information theoretic limits of the described system are given in [1]. Independent noise components originating from synchronization and quantization errors, for example, might lead to key disagreement between the users. Thus, to ensure that identical keys are obtained on both sides, reconciliation procedures that require additional side information to be transmitted need to be employed. More details on the exact measurements, correlation between the legitimate channels and eavesdroppers channel as a function of the separation between the antennas relative to the wavelength have been investigated in [1], [4]–[6]. Independent measurements conducted in an indoor environment, for the purpose of key generation, are described in [7]. For the purpose of this work we assume that the eavesdropper is located further away than the minimum coherence distance $\lambda/2$, resulting in uncorrelated channels between the legitimate users and the eavesdropper. In recent works, such as [7], this assumption has been shown not to be very accurate and up to 10% of the information can leak to Eve. However, coupled with privacy amplification, such a problem is resolved, and thus, not the focus of our contribution here.

In previous works, [2], [3], the reconciliation bits were sent over a noiseless channel which is not realistic and in here, we now address the case where the side information is made available over a noisy channel.

While the secret-key agreement idea is based on the wiretap channel model [8], we do not use the main channel by utilizing SNR advantage based methods to transmit messages like in [9], rather the fluctuating channel state is exploited to generate the keys. In case of a line-of-sight channel, either movements or reconfigurable antenna arrays allow to provide the necessary randomization. Hence, the secret-key can be generated by observing the channel state information (CSI) at Alice and Bob [1].

The key generation technique discussed here would in general use the Linde-Buzo-Gray or Lloyd-Max vector quantizers. When assuming a Gaussian channel distribution, obtained, e.g., by a large reconfigurable (RECAP) antenna array, the quantization can easily be precomputed and results in a simple cut in the middle of the distribution for the binary case. For clearness of presentation, this conference paper will focus on this binary case, only. For reconciliation, a Slepian-Wolf [2] based method is employed using LDPC codes due to their capacity approaching performance. Further details about channel data, quantization aspects and results, as well as the Slepian-Wolf LDPC scheme which was used, are provided in [1], [11], [12].

Moreover, since the secret key is not transmitted over the wiretap channel, but results from measurements plus side information, the intrinsic log-likelihood ratio (LLR) calculation required for the LDPC decoder is more involved and resulting message densities are non-Gaussian and non-consistent. The main contribution of this paper lies in designing the LDPC code through density evolution for such a system.

The paper is structured as follows. In Section II, the system description is provided and code design aspects are discussed. In Section III, the intrinsic LLR is derived and the properties of this function are discussed. In Section IV and V, density evolution steps for the system and the linear program for designing the code are presented, respectively. Section VI provides BER simulation results.

◇ A. Filip is now with DLR, Oberpfaffenhofen, Germany.

978-1-4799-5952-5/15/$31.00 ©2015 IEEE

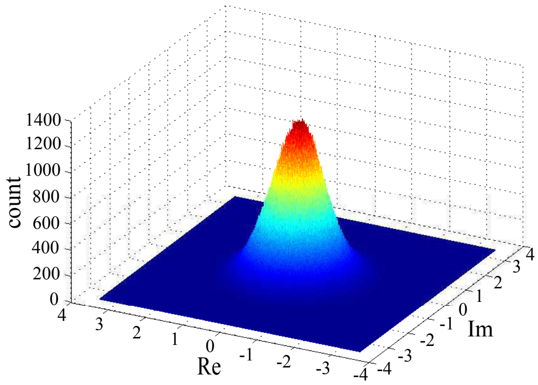

Fig. 1. Channel density for 24 REs.

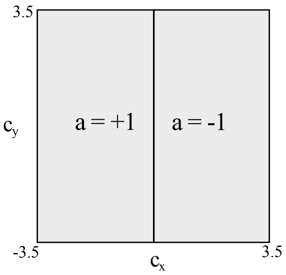

Fig. 2. Binary codebook. (note that we chose -1 to be on the right, which has implications for the following figure)

II. System Description

For our investigation in this paper, we consider an ideal Gaussian channel. In case of stationary users, the randomized Gaussian channel can almost perfectly be obtained by large RECAP antenna arrays; a corresponding channel density is shown in Fig. 1.

At Alice, a parasitic RECAP array with 5×5 elements, i.e., 24 reconfigurable elements and a center feed element, is used and at Bob, a single dipole antenna. The binary quantization is presented in Fig. 2.

Alice and Bob estimate the channel in neighboring time slots (TDD system) to generate the key. On Alice's side, the analog measurement data is then quantized and the key is generated. We take Alice's quantized result to be the correct key symbol. Due to CSI differences arising from independent noise on Bob's side, his quantized results would be erroneous with respect to Alice's "correct" results. Hence, we require some reconciliation procedure. Reconciliation is performed based on Bob's analog measurements through soft decoding.

A. Key Generation and Reconciliation: Slepian-Wolf Coding

For key reconciliation, Alice sends parity bits to Bob. We notice two channels in the scheme at this point. First, there is the channel which is measured by Alice and Bob, no data is transmitted over this channel, which we will refer to as the virtual measurement channel. There is also a transmission channel over which Alice sends parity bits.

We have assumed the variance of the channel distribution Alice and Bob encounter are identical. This assumption is practical given that, in the time interval within which the CSI measurements are made and the parity bits sent, the channel distribution remains stationary, i.e., it is a quasi-static channel. Additionally, there is independent, identically distributed, circular-symmetric Gaussian noise at Alice's and Bob's ends, ignoring other effects due to hardware imperfections and alike. Although we have formally defined two channels in our system, the noise power on both channels (from Bob's perspective) is assumed to be the same i.i.d. AWGN.

As key reconciliation procedure, we use Slepian-Wolf coding [2] based on LDPC codes. Since the key estimates at Alice and Bob can be seen as correlated information due to the channel reciprocity, Bob can decode his data using 'side-information' or 'redundancy' from Alice, hence Slepian-Wolf is appropriate. The lower bound M for the redundancy is given in (1), where $H(a|b)$ is the conditional entropy of 'Alice' given 'Bob'.

$$M = H(a|b)\,\text{bit}. \tag{1}$$

For an n bit key, the number of reconciliation bits M_p are [12]

$$M_p \geq nM = nH(a|b)\,\text{bit}. \tag{2}$$

Here, of the two variants of Slepian-Wolf coding, parity and syndrome, we chose the syndrome method for our implementation.

For the syndrome approach, the reconciliation information is computed as the syndrome of the length n quantized source vector $\mathbf{a}$. A syndrome $\mathbf{s}$ is defined as

$$\mathbf{s} = \mathbf{H}\mathbf{a}^T \ , \tag{3}$$

where $\mathbf{H}$ is the parity-check matrix. From (2), we know that the reconciliation information should have length $nH(a|b)\,\text{bit}$.

The reconciliation information $\mathbf{s}$ is sent over the physical channel and thus is subject to eavesdropping. To protect against eavesdropping, at most twice the number of reconciliation bits are needed. The procedure is termed *privacy amplification* [10], however, we do not discuss this aspect in here further.

B. LDPC Code Construction

The M_p reconciliation bits (Slepian-Wolf syndrome approach) which are necessary for Bob to successfully decode to Alice's key are sent to Bob over the transmission channel and must be error-protected due to the noisy channel. Hence, the final code design has two sets of codes. The first code $\mathcal{C}_m$ with generator and parity-check matrices $\mathbf{G}_m$ and $\mathbf{H}_m$ are used to generate the syndrome for reconciliation by

$$\mathbf{s} = \mathbf{H}_m\mathbf{a}^T \ , \qquad \mathbf{a} = \text{Alice's quantized key}. \tag{4}$$

The length of syndrome $\mathbf{s}$ must satisfy (2). The second code is used to protect the reconciliation bits obtained from (4). The syndrome s is the information vector for the second code $\mathcal{C}_s$

with generator and parity-check matrices $\mathbf{G}_s$ and $\mathbf{H}_s$. The final codeword $\mathbf{v}$ for the overall parity bits is then

$$\mathbf{v} = \mathbf{s}^T\mathbf{G}_s = \mathbf{a}\mathbf{H}_m^T\mathbf{G}_s \ . \tag{5}$$

The length of $\mathbf{v}$ is $M_p(1+\beta)$ where β is the fraction of reconciliation bits required as additional redundancy for forward error correction. Since there is a direct relationship between Alice's quantized vector $\mathbf{a}$ and and the final codeword $\mathbf{v}$, we can think of a single LDPC code with equivalent generator matrix $\mathbf{G} = [I_{n\times n} \ \ \mathbf{H}_m^T\mathbf{G}_s]$ which encodes $\mathbf{a}$ systematically for the information part and also computes the redundancies (5) to be sent over the channel. The rate of the code is given by

$$R = \frac{n}{n + M_p(1+\beta)} \ . \tag{6}$$

We use a multi-edge-type description due to the two channels in our system, which is described in detail in [12]. Although the noise power on both the channels is the same, the intrinsic channel information (L_{ch}) calculation for the measured key bits and received parity bits are different at Bob's end.

III. Intrinsic Channel Information Derivation

The channel between Alice and Bob is characterized by AWGN noise with standard deviation σ_b. So the log-likelihood ratio $L_{ch_\mathbf{v}}$ for the parity bits is straightforward assuming a Gaussian pdf. For a received bit r_i given that Alice transmitted v_i, the probability density function is

$$p(r_i|v_i = \pm 1) = \frac{1}{\sqrt{2\pi\sigma_b^2}} e^{\frac{-(r_i \mp 1)^2}{2\sigma_b^2}} \ , \tag{7}$$

Hence, the LLR is

$$L_{ch_\mathbf{v}} = \ln\left(\frac{e^{-\frac{(r_i-1)^2}{2\sigma_b^2}}}{e^{-\frac{(r_i+1)^2}{2\sigma_b^2}}}\right) = \frac{2r_i}{\sigma_b^2} \ . \tag{8}$$

The intrinsic LLR calculation for the information bits, i.e., the estimated key bits, is more complicated since the measurement data is obtained by Bob and no information about Alice's quantization values is available to him, yet the LLR formulation is

$$L_{ch}(b) = \ln\left(\frac{P(b|a=+1)}{P(b|a=-1)}\right) , \tag{9}$$

where b refers to Bob's analog measured value, and a denotes Alice's quantized value. Some steps of the derivation are presented here, the complete description can be found in [13]. Since for the binary quantization we use, the decision boundary is parallel to the imaginary axis, L_{ch} values are dependent on one dimension only.

For the 1-D case, a point c_x[1] from the channel distribution is measured as a_x and b_x by Alice and Bob, respectively, with variances $\sigma^2_{a_x}$ and $\sigma^2_{b_x}$, while the channel variance is $\sigma^2_{ch_x}$.

$$p(c_x) = \frac{1}{\sqrt{2\pi}\sigma_{ch_x}} \exp\left[-\frac{c_x^2}{2\sigma^2_{ch_x}}\right] , \tag{10}$$

$$p(b_x|c_x) = \frac{1}{\sqrt{2\pi}\sigma_{b_x}} \exp\left[-\frac{(b_x - c_x)^2}{2\sigma^2_{b_x}}\right] \ , \tag{11}$$

$$p(a_x|c_x) = \frac{1}{\sqrt{2\pi}\sigma_{a_x}} \exp\left[-\frac{(a_x - c_x)^2}{2\sigma^2_{a_x}}\right] \ , \tag{12}$$

Applying Bayes' rule to (11) and (12), this yields

$$p(c_x|b_x) = \frac{1}{\sqrt{2\pi}\sigma_{b_x}} \exp\left[-\frac{(b_x - c_x)^2}{2\sigma^2_{b_x}}\right] \cdot \frac{p(c_x)}{p(b_x)} \ , \tag{13}$$

$$p(c_x|a_x) = \frac{1}{\sqrt{2\pi}\sigma_{a_x}} \exp\left[-\frac{(a_x - c_x)^2}{2\sigma^2_{a_x}}\right] \cdot \frac{p(c_x)}{p(a_x)} \ . \tag{14}$$

Determining the probability of what Alice may have quantized to,

$$P(a_x = +1|b_x) = \int_{R_+}\int_{R} p(a_x|c_x)\cdot p(c_x|b_x)\,\mathrm{d}c_x\,\mathrm{d}a_x \ , \tag{15}$$

$$P(a_x = -1|b_x) = \int_{R_-}\int_{R} p(a_x|c_x)\cdot p(c_x|b_x)\,\mathrm{d}c_x\,\mathrm{d}a_x \ . \tag{16}$$

Applying Bayes' rule to (15) and (16), we finally obtain (17) and (18), and use in (9) to obtain the required intrinsic LLR. Note that, due to equal probability assumption $P(a_x = +1) = P(a_x = -1) = 0.5$.

In Fig. 3, the L_{ch} values for the quantization presented in Fig. 2 are shown. The CSI measurements were mapped to a two-dimensional discrete grid from $[-3.5 \text{ to } 3.5]$ on both axes in increments of 0.005. Here, $\sigma^2_{a_x} = \sigma^2_{b_x}$, some values of which are provided in Table I. The SNR is defined as $10\log_{10}(\frac{\sigma^2_{ch_x}}{\sigma^2_{a_x}})$, where $\sigma^2_{ch_x} = 0.4203$. As expected, the L_{ch} function is symmetric (odd). Additionally, we see that with increasing SNR the magnitudes of the LLRs increase as expected.

TABLE I
Variance values

SNR [dB]	$\sigma^2_{a_x} = \sigma^2_{b_x}$
11	0.0334
13	0.0211
15	0.0133
17	0.0084
19	0.0053

A. *Properties of the Intrinsic Log-likelihood Ratio Function*

We first describe the consistency property of densities. A density of probability $f(x)$ is said to be consistent (i.e. with exponential symmetry) if

$$f(x) = e^x \cdot f(-x) \ , \forall x \in \mathbb{R} \ . \tag{19}$$

Consistent densities have the property that

$$\mu = \frac{\sigma^2}{2} \ . \tag{20}$$

For the design of an LDPC soft decoder, it is necessary to differentiate between consistent and inconsistent LLR message densities, as this property dictates the exact equations to be used

[1] The subscript x is used to address 1-D values of the complex distribution.

$$p(b_x|a_x=+1) = \frac{1}{2\pi\sigma_{a_x}^2}\frac{1}{\sqrt{2\pi}\sigma_{ch_x}}\int_{R+}\int_{R}\exp\left[-\frac{(a_x-c_x)^2}{2\sigma_{a_x}^2}-\frac{(b_x-c_x)^2}{2\sigma_{b_x}^2}-\frac{c_x^2}{2\sigma_{ch}^2}\right]\mathrm{d}c_x\,\mathrm{d}a_x\cdot\frac{1}{0.5}, \tag{17}$$

$$p(b_x|a_x=-1) = \frac{1}{2\pi\sigma_{a_x}^2}\frac{1}{\sqrt{2\pi}\sigma_{ch_x}}\int_{R_-}\int_{R}\exp\left[-\frac{(a_x-c_x)^2}{2\sigma_{a_x}^2}-\frac{(b_x-c_x)^2}{2\sigma_{b_x}^2}-\frac{c_x^2}{2\sigma_{ch}^2}\right]\mathrm{d}c_x\,\mathrm{d}x_a\cdot\frac{1}{0.5}. \tag{18}$$

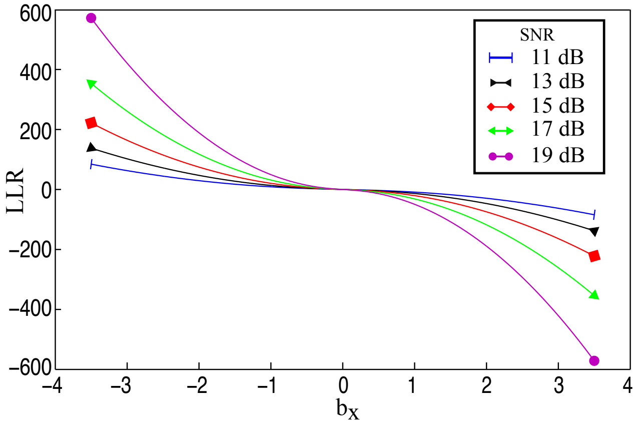

Fig. 3. LLR (L_{ch}) values for the binary codebook case

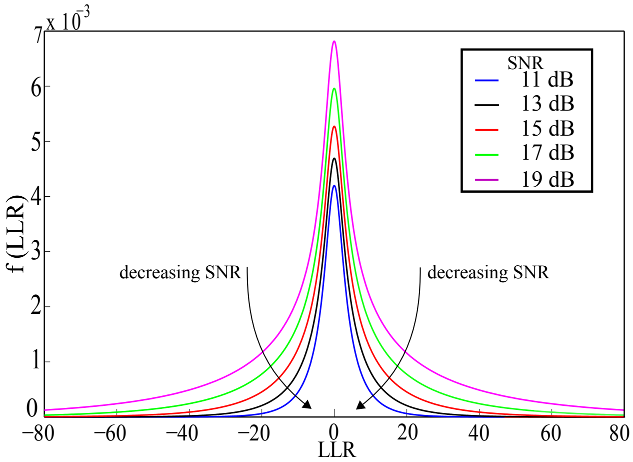

Fig. 4. PDF of the L_{ch}

in Density Evolution for calculating the mutual information [15].

The L_{ch} function for the binary case is a function of b_x. Using the density functions of b_x, we obtain the densities of the L_{ch} functions. The density of b_x is a Gaussian with variance,

$$\sigma_1^2 = \sigma_{ch_x}^2 + \sigma_{b_x}^2, \tag{21}$$

where, $\sigma_{b_x}^2$ decreases with an increase in SNR. The density functions $f(L_{ch})$ are plotted in Fig. 4 for SNRs of 11 to 19 dB. We now take a look at the (onesided) mean μ and variance of the distribution in Table II and conclude that the density is not consistent.

TABLE II
MEAN $(\mu_{L_{ch}})$ AND VARIANCE $(\sigma^2_{L_{ch}})$ OF THE L_{ch} DENSITY

SNR [dB]	$\mu_{L_{ch}}$	$\sigma^2_{L_{ch}}$
11	3.1353	12.4572
13	4.1273	26.5784
15	5.4931	57.7666
17	7.4049	127.3862
19	10.2589	294.3257

IV. DENSITY EVOLUTION STEPS FOR LDPC CODE DESIGN

The LDPC code design is done through a linear optimization algorithm based on a rate maximization criterion and uses density evolution at both sets of nodes to check the convergence through mutual information calculations. Hence, we take a look at the Belief - Propagation (BP) decoding updates at the check and variable nodes, respectively,

$$L_{v_ic_j}^{(l)} = L_{ch,i} + \sum_{k\neq j} L_{c_kv_i}^{(l-1)}\ , \tag{22}$$

$$L_{c_iv_j}^{(l)} = 2\tanh^{-1}\left(\prod_{k\neq j}\tanh\left(\frac{L_{v_kc_j}^{(l)}}{2}\right)\right)\ . \tag{23}$$

The subscript vc denotes an edge from variable-to-check node and cv denotes a check-to-variable node edge.
At the variable node side, the outgoing message from node v_i to c_j in the l^{th} iteration is the sum of the incoming messages from the remaining check nodes from the $(l-1)^{th}$ iteration and the channel intrinsic information, $L_{ch,i}$. In order to calculate the mutual information $x_{v_ic_j}$, we need the density of message $L_{v_ic_j}$. $L_{ch,i}$ is represented by $f(L_{ch,i})$. The addition of LLR values to be passed along the edge connecting v_i and c_j leads to a convolution of the contributing PDFs. Hence,

$$L_{v_ic_j}^{(l)} : {}^{2}f(L_{cv}^{(l-1)}) \star f(L_{ch,i}) = f(L_{v_ic_j}^{(l)})\ . \tag{24}$$

Equation (24) requires some modification. We know, in general, the LLR for a bit value x_i given a received vector $\mathbf{y}$ is given by (25).

From (25), it becomes clear that the $a_x = +1$ and $a_x = -1$ probabilities are treated separately (i.e. under the $a_x =$

[2] This density relates to the sum $\sum_{k\neq j} L_{c_kv_i}^{(l-1)}$ from (22), the same applies to (26) and (27).

$$\ln\frac{P(x_i=+1|\mathbf{y})}{P(x_i=-1|\mathbf{y})}=\ln\frac{P(\mathbf{y}|x_i=+1)}{P(\mathbf{y}|x_i=-1)}+\ln\frac{P(x_i=+1)}{P(x_i=-1)}$$

$$=\underbrace{\ln\frac{P(y_i|x_i=+1)}{P(y_i|x_i=-1)}}_{L_{\text{intrinsic}}}+\underbrace{\ln\frac{P(\mathbf{y}_{\setminus i}|x_i=+1)}{P(\mathbf{y}_{\setminus i}|x_i=-1)}}_{L_{\text{extrinsic}}}+\underbrace{\ln\frac{P(x_i=+1)}{P(x_i=-1)}}_{L_{\text{a-priori}}}. \quad (25)$$

±1 conditions, only probabilities under the same condition are multiplied) and thus, in density domain, they are convolved separately, too. Hence, following (24), we define

$$f(L_{v_ic_j}^{(l)}|a_x=+1){=}f(L_{cv}^{(l-1)}|a_x=+1)\star f(L_{ch,i}|a_x=+1), \quad (26)$$

$$f(L_{v_ic_j}^{(l)}|a_x=-1){=}f(L_{cv}^{(l-1)}|a_x=-1)\star f(L_{ch,i}|a_x=-1)\,. \quad (27)$$

We now write (28), used to calculate the mutual information on the variable node side, using (26) and (27).

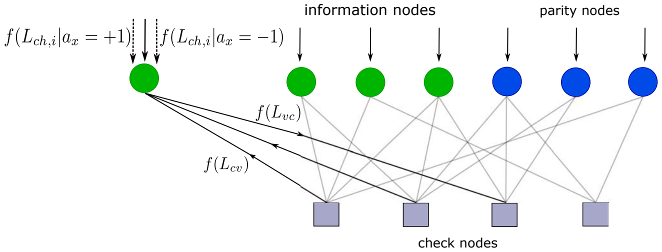

Fig. 5. Density evolution steps at variable nodes. The dotted incoming lines at a variable node represent the constituent densities of the incoming channel intrinsic information.

We now focus on the check-node side, in order to find the message densities required for the aforementioned convolution. From (23), we see that the outgoing message from check node i to variable node j in the l^{th} iteration is a non-linear combination of the l^{th} iteration messages from variable nodes $v_{k,\,k\neq j}$ to node c_i. From [14], a general assumption can be made that allows us to treat the outgoing message from a check-node as having a Gaussian distribution due to the central limit theorem, since there are many messages being combined. Additionally, there we assume the messages to be consistent, and thus, we employ a simplified version of (28) which makes use of the exponential symmetry from (19) by using the mean m of a consistent density, $z\sim\mathcal{N}(m,2m)$, and delivers the mutual information x_z [15]–[17].

$$J(m)=1-\frac{1}{\sqrt{4\pi m}}\int_{\mathcal{R}}e^{-\frac{(z-m)^2}{4m}}\log_2(1+e^{-z})\mathrm{d}z=x_z. \quad (29)$$

$J(m)$ is a continuous and strictly monotonous function, so J^{-1} exists and allows for computing the mean m of the LLR from the mutual information x_z.

Since all messages $L_{c_kv_i}$ are considered consistent Gaussians, in order to obtain $f(L_{cv}^{(l-1)}|a_x=+1)$ used in (26), we need only add the means of the individual messages. J^{-1} is used to obtain the means m and then added, due to consistency, the variance can be calculated and the resultant message represented. Due to symmetry, the mean has to be mirrored only, to obtain $f(L_{cv}^{(l-1)}|a_x=-1)$ used in (27).

Now, we derive $f(L_{ch,i}|a_x=+1)$ and $f(L_{ch,i}|a_x=-1)$, also used in (26, 27). However, let us first explain the LLR random variable of the form $(L|a_x=\pm1)$. L is a log-likelihood ratio and thus already compares the probabilities of $p(b_x|a_x=+1)$ and $p(b_x|a_x=-1)$. Further requiring the density of this random variable under the conditions $a_x=+1$ and $a_x=-1$ may seem counterintuitive at first sight, however, the LLR is a value dependent on b_x and we only then consider the $a_x=\pm1$ range, i.e., we consider the values of b_x as they result from the $a_x=+1$ or the $a_x=-1$ ranges.

$$f(L_{ch,i}|a_x=+1)=\sum_{b_x:K^{-1}(L_{ch,i})=b_x}\frac{f_{b_x|a_x=+1}(b_x)}{|\frac{d}{d(b_x)}L_{ch,i}(b_x)|}, \quad (30)$$

$$f(L_{ch,i}|a_x=-1)=\sum_{b_x:K^{-1}(L_{ch,i})=b_x}\frac{f_{b_x|a_x=-1}(b_x)}{|\frac{d}{d(b_x)}L_{ch,i}(b_x)|}. \quad (31)$$

Where, $K(b_x)=L_{ch,i}$ is shown in Fig. 3.

V. Linear Optimization

After deriving the density evolution steps for our case, we now present the linear programming algorithm for finding the optimized degree distributions in Algorithm **1**. The subscript j is used to distinguish the two classes of variable nodes, $j=1$ refers to information nodes, while $j=2$ is for parity nodes. The presented proportion distribution constraints are discussed in detail in [12]. The densities required for convergence conditions for variable nodes dealing with information estimates were discussed in Section IV. Equations (41) and (43), the mutual information updates at variable nodes for parity bits and check nodes, respectively, follow from [17].

Note the check node side update step (43). We notice, within the summation, the J^{-1} function is applied to $(1-x_{vc})$, i.e., addresses the mutual information on an outgoing edge of a variable node. We know, in order to use the J^{-1} function, we need the density of the LLR random variable to be consistent but $f(L_{vc})$ is not since it is obtained by convolving with $f(L_{ch})$ at every iteration, which is not consistent. We however keep this assumption on the check node side, for now. Under this assumption, the linear optimization algorithm still converges. We will address the more exact treatment of the incoming and outgoing messages at the check node side in future realizations.

Density evolution can be summarized as a function of the degree distributions, densities of the messages, and the mutual information from the previous iterations. In order to assure convergence, we require the mutual information to increase after every iteration, as shown in (36). λ and ρ are the degree distribution polynomials for the variable and check nodes, respectively. We provide the results of the linear optimization for 50 iterations. The routine delivers the fraction $(1+\beta)$ of reconciliation bits (M_P reconciliation bits mentioned in (2), according to entropy calculations at the specified SNR [11]) required as total redundancy for maximizing the rate

$$x^{(l)}_{v_i c_j} = \int_{\mathcal{R}} f(L^{(l)}_{v_i c_j}|a_x = +1) \log_2 \left(\frac{2f(L^{(l)}_{v_i c_j}|a_x = +1)}{f(L^{(l)}_{v_i c_j}|a_x = +1) + f(L^{(l)}_{v_i c_j}|a_x = -1)} \right) \mathrm{d}(L_{v_i c_j}) \ . \tag{28}$$

Optimize

$$\min_{\beta \in \mathbb{R}^+} (1+\beta) \ , \tag{32}$$

subject to

1) Proportion distribution constraints [12]

1.1

$$\sum_{j=1}^{2} \sum_{i=2}^{d_{\text{vmax}_j}} \lambda_i^{(j)} = 1 \tag{33}$$

1.2

$$\sum_{i=2}^{d_{\text{vmax}_2}} \frac{\lambda_i^{(2)}}{i} = \sum_{i=2}^{d_{\text{cmax}}} \frac{\rho_i}{i} \tag{34}$$

1.3

$$\frac{M_p}{n}(1+\beta) \sum_{i=2}^{d_{\text{vmax}_1}} \frac{\lambda_i^{(1)}}{i} = \sum_{i=2}^{d_{\text{cmax}}} \frac{\rho_i}{i} \ , \quad \beta \geq 0 \tag{35}$$

2. Convergence condition

$$F\left(\boldsymbol{\lambda}, \boldsymbol{\rho}, x^{(l)}_{vc}\right) > x^{(l-1)}_{vc} \ \text{ with ,} \tag{36}$$

$$x^{(l)}_{vc_{j=1}} = 0; \tag{37}$$

for $i = 2 : d_{\text{vmax}_j}$, $j = 1$: information bits,

$$f(L^{(l)}_{vc}|a_x = +1) \ = \ f_{(L^{(l-1)}_{cv}|a_x=+1)}((i-1)m) \star f(L_{\text{intrinsic}}|a_x = +1) \ , \tag{38}$$

$$f(L^{(l)}_{vc}|a_x = -1) \ = \ f_{(L^{(l-1)}_{cv}|a_x=-1)}((i-1)m) \star f(L_{\text{intrinsic}}|a_x = -1) \ , \tag{39}$$

m = mean of the Gaussian densities.

$$x^{(l)}_{vc_{j=1}} = x^{(l)}_{vc_{j=1}} + \lambda_i^{(j=1)} \int_{\mathcal{R}} f(L^{(l)}_{vc}|a_x = +1) \ldots$$

$$\log_2\left(\frac{2f(L^{(l)}_{vc}|a_x = +1)}{f(L^{(l)}_{vc}|a_x = +1) + f(L^{(l)}_{vc}|a_x = -1)} \right) \mathrm{d}(L_{vc}) \ . \tag{40}$$

end for.

$$x^{(l)}_{vc_{j=2}} = \sum_{i=2}^{d_{\text{vmax}_{j=2}}} \lambda_i^{(j=2)} J\left(\frac{2}{\sigma_b^2} + (i-1)\, J^{-1}\left(x^{(l-1)}_{cv} \right) \right) \ , \tag{41}$$

$$x^{(l)}_{vc} = \sum_{j=1}^{2} x^{(l)}_{vc_j} \tag{42}$$

$$x^{(l-1)}_{cv} = 1 - \sum_{h=2}^{d_{\text{cmax}}} \rho_h J\left((h-1)\, J^{-1}\left(1 - x^{(l-1)}_{vc} \right) \right) \ . \tag{43}$$

Algorithm 1: Linear programming algorithm

and the optimized degree distributions. The check node degree distribution was fixed to be

$$\rho(x) = 0.98x^9 + 0.02x^{10} \ . \tag{44}$$

The maximum variable node degrees for the two classes were chosen as $d_{vmax_1} = 15$ and $d_{vmax_2} = 15$. The length of the information word is $n = 2^{10} = 1024$. $M_p(1+\beta)$ is the total required redundancy.

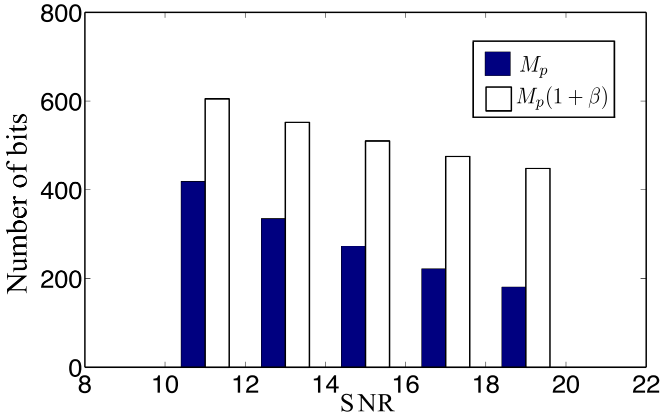

Fig. 6. Redundancy requirements delivered by the linear program

The results are illustrated in Fig. 6 and Table III. As expected, with increasing SNR, the total required redundancy decreases. The corresponding rates of the codes are also shown in Table III. There is a lower bound for $M_p(1+\beta)$ derived in [12] which explains the somewhat surprising behavior of the growth of $1+\beta$, making up for the decrease in the conditional entropy with growing SNR.

TABLE III
REDUNDANCY AND RATE RESULTS

SNR [dB]	M_p	$(1+\beta)$	$M_p(1+\beta)$	Rate
11	419	1.4446	605	0.6286
13	335	1.6475	552	0.6497
15	273	1.8669	510	0.6675
17	222	2.1377	475	0.6831
19	181	2.4750	448	0.6957

VI. BER Simulation

The BER simulation was performed to check the performance of the code. Note that the BER is an indicator for the key agreement rate as it measures the mismatch between Alice's measured bits and Bob's decoded bits, i.e., the binaries of the secret key. Figure 7 shows the BER against SNR plot for the code designed at 11 dB. The BER ratio for the unreconciled case is also provided for comparison purposes. Degree distributions are provided in Table IV. No error floor is visible.

VII. Conclusion

In this paper we considered a physical-layer key reconciliation scheme between two legitimate users of a reciprocal channel. The LDPC code designed for the reconciliation procedure is obtained via density evolution. From the simulation results,

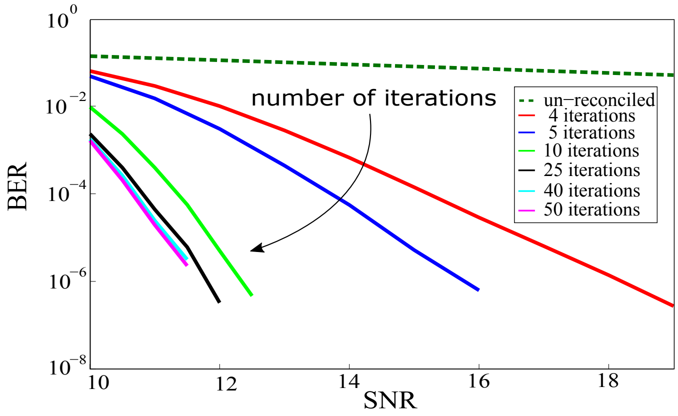

Fig. 7. BER results

TABLE IV
VARIABLE NODE SUB-DEGREE DISTRIBUTIONS

SNR	$(1+\beta)$	$\boldsymbol{\lambda}^{(1)}(x)$	$\boldsymbol{\lambda}^{(2)}(x)$
11 dB	1.4446	$\lambda_2 = 0.1147$ $\lambda_3 = 0.1574$ $\lambda_4 = 0.1938$ $\lambda_8 = 0.1266$ $\lambda_{13} = 0.0417$ $\lambda_{14} = 0.1441$	$\lambda_2 = 0.2218$

we conclude that the final design of the LDPC code delivers promising BER results for the key reconciliation procedure when the secret-key is generated by CSI measurements at Alice and Bob. Extensions of this work would focus on exact density evolution treatment on the check node side. The stability conditions of the optimization have to be formulated for our case, too.

ACKNOWLEDGMENT

This work is funded by the German Research Foundation (DFG).

REFERENCES

[1] J. Wallace, “Secure Physical Layer Key Generation Schemes: Performance and Information Theoretic Limits,” in proc. *IEEE International Conference on Communications*, June 1-5, 2009.

[2] X. Sun, X. Wu, C. Zhao, M. Jiang, and W. Xu, “Slepian-Wolf Coding for Reconciliation of Physical Layer Secret Keys,” proc. *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6, 18-21 Apr. 2010.

[3] M. Bloch, A. Thangaraj, S.W. McLaughlin, and J.-M. Merolla, “LDPC-Based Secret Key Agreement over the Gaussian Wiretap Channel,” proc. *IEEE ISIT*, Seattle, WA, pp. 1179-1183, July 9-14, 2006.

[4] J. Wallace, R. Kurma, “Automatic Secret Keys From Reciprocal MIMO Wireless Channels: Measurement and Analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 5, pp. 381-392, Sept. 2010.

[5] R. Mehmood and J. Wallace, “Wireless Security Enhancement Using Reconfigurable Aperture Antennas,” *European Conference on Antennas and Propagation (EuCAP’11)*, Rome, Italy, Apr. 12-16, 2011, pp. 1-5.

[6] R. Mehmood and J. Wallace, “MIMO Capacity Enhancement Using Parasitic Reconfigurable Aperture Antennas (RECAPs),” *IEEE Transactions on Antennas and Propagation*, vol. 60, pp. 665-673, Feb. 2012.

[7] A. J. Pierrot, R. A. Chou, and M. R. Bloch, “Experimental Aspects of Secret Key Generation in Indoor Wireless Environments,” *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on. IEEE*, pp. 669-673, 2013.

[8] A.D. Wyner, “The Wiretap Channel,” *Bell Syst. Tech. J.*, vol. 54, no. 8, pp. 1355-1367, oct. 1975.

[9] C.W. Wong, T.F. Wong., and J.M. Shea, “LDPC Code Design for the BPSK-constrained Gaussian Wiretap Channel,” proc. *IEEE GLOBECOM Workshops 2011*, pp. 898-902, 2011.

[10] C. Cachin and U. M. Maurer, “Linking Information Reconciliation and Privacy Amplification,” *Journal of Cryptology*, vol. 10, no. 2, pp. 97-110, 1997.

[11] A. Filip, R. Mehmood, J. Wallace, and W. Henkel, “Physical-Layer Key Generation Supported by RECAP Antenna Structures,” proc. *9th International ITG Conference on Source and Channel Coding (SCC)*, Munich, Germany, 2013.

[12] J. Etesami and W. Henkel, “LDPC Code Construction for Wireless Physical-Layer Key Reconciliation,” proc. *First IEEE International Conference on Communications in China (ICCC 12)*, Beijing, China, 2012.

[13] O. Graur, N. Islam, A. Filip, and W. Henkel, “Quantization Aspects in LDPC Key Reconciliation for Physical Layer Security” *10th International ITG Conference on Systems, Communications and Coding (SCC)*, Hamburg, Germany, February 2-5, 2015.

[14] T. Richardson and R. Urbanke, Binary Erasure Channel, in Modern Coding Theory. *New York*, NY, USA: Cambridge University Press, 2008.

[15] T. J. Richardson, M. A. Shokrolahi, and R. L. Urbanke, “Design of capacity-Approaching Irregular-Low-Density parity-Check Codes,” in *IEEE Trans, on Information Theory*, vol. 47, no.2, pp. 619-637, Feb. 2001.

[16] S. Y. Chung, T. J. Richardson, and R. L. Urbanke,, “Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes using a Gaussian Approximation,” in *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 657-670, Feb. 2001.

[17] N.von Deetzen and S. Sandberg “On the UEP Capabilities of Several LDPC Construction Algorithms,” in *IEEE Trans, on Communications*, vol. 58, no.3, pp. 3041-3046, March 2010.