Finite Length LT Codes over $\mathbb{F}_q$ for Unequal Error Protection with Biased Sampling of Input Nodes

Birgit Schotsch and Radu Lupoaie*
Institute of Communication Systems and Data Processing (ind)
RWTH Aachen University, Aachen, Germany
schotsch@ind.rwth-aachen.de

***Abstract*—Finite length LT codes over higher order Galois fields $\mathbb{F}_q$ for unequal error protection (UEP) are analysed under maximum likelihood (ML) decoding. We consider a biased sampling method to create the LT code graph. In contrast to a previous approach by Rahnavard et al., where a predetermined number of edges is created per importance class given a check node of degree *d*, our procedure allows to precisely adjust the desired class weights. Moreover, we provide upper and lower bounds on the symbol erasure probability for each importance class.**

I. Introduction

Fountain codes are a class of incremental redundancy codes [1] that have been proposed in [2] as an alternative approach to retransmission schemes to recover lost packets in packet-switched communication networks. Fountain codes are rateless erasure correcting codes such as LT (Luby Transform) codes [3], Raptor codes [4] and Online codes [5] for which simple and efficient encoding and decoding algorithms exist. Originally developed for the binary erasure channel (BEC), rateless codes do not require any information about the erasure probability. Especially in point-to-multipoint transmission scenarios, where the individual and independent channel conditions of the users are not known to the transmitter, this characteristic is particularly useful.

Besides the original studies on rateless codes that target at equal error protection (EEP) of data, some proposals for rateless codes for unequal error protection (UEP) have followed. EEP is needed, e.g. for the distribution of bulk data [2], while UEP is better suited for, e.g. audio or video transmission where some parts of the data are more important than others and therefore need a stronger protection. Two examples of rateless UEP schemes for LT codes are the approach by Rahnavard et al. [6] which we will refer to as weighted UEP and the expanding window (EW) method by Sejdinovic et al. [7]. However, in this paper we will only deal with the weighted UEP method, in which we use *biased sampling* of the input symbols in order to allow for *continuous* effective weights of the differently important data parts. Furthermore, we provide upper and lower bounds on the symbol erasure probability under maximum likelihood (ML) decoding.

In the following, we consider LT codes over Galois fields $\mathbb{F}_q$ of order $q = 2^m$, where $m \geq 1$, since it has been shown recently that rateless codes over higher order Galois fields exhibit a better erasure correction performance than their binary counterparts [8], [9]. Additionally, we have demonstrated in [10] that this improved erasure correction performance even comes with a lower computational complexity if the equivalent binary input size is kept constant.

Using rateless codes, the transmitter can generate a potentially infinite number n_T of encoded symbols $\mathbf{y} = (y_1, y_2, \dots y_{n_\mathrm{T}})$ from a finite amount of k input symbols $\mathbf{u} = (u_1, u_2, \dots u_k)$. Though in practice the input and output symbols u_i and y_j consist of l $\mathbb{F}_q$-elements each, where $i \in \{1, 2, \dots k\}$ and $j \in \{1, 2, \dots n_\mathrm{T}\}$, we consider only $l = 1$ in the following as this number l has no influence on the erasure correction performance of the codes [3]. Note that $\mathbb{F}_q$-elements have an equivalent binary representation which requires m bits per element. In order to allow for a fair comparison of codes over Galois fields of different orders, we fix the number $k = k_2$ of input *bits* and distribute them to $k_q = \lceil \frac{k_2}{\mathrm{ld}\, q} \rceil = \lceil \frac{k_2}{m} \rceil$ input *symbols*. Thus, the input size of a code over $\mathbb{F}_q$ with $q = 2^m$ is k_q.

In general, rateless codes are designed such that the receiver is able to decode the original k_q input symbols $\mathbf{u}$ from any $n_\mathrm{R} = k_q(1+\varepsilon_\mathrm{R})$ received code symbols with high probability if $\varepsilon_\mathrm{R} \geq 0$, where ε_R is the required relative reception overhead.

II. LT Codes over $\mathbb{F}_q$

The generator matrix $\mathbf{G} \in \mathbb{F}_q^{n_\mathrm{T} \times k_q}$ of an LT code, with $q = 2^m$, defines a weighted graph that connects the set of k_q input nodes $\mathbf{u} \in \mathbb{F}_q^{1 \times k_q}$ to the set of n_T output nodes $\mathbf{y} \in \mathbb{F}_q^{1 \times n_\mathrm{T}}$, where n_T can be arbitrarily large. A more detailed description of *binary* LT codes can be found in [3].

The input symbols are assigned to input nodes and the output symbols are assigned to output nodes that are also called check nodes. In vector-matrix notation we encode by $\mathbf{y}^\mathrm{T} = \mathbf{G}\mathbf{u}^\mathrm{T}$. In contrast to traditional block codes, $\mathbf{G}$ is generated online and may differ for each data block. $\mathbf{G}$ is assumed to be known at the decoder. This can be achieved by synchronised pseudo-random processes that produce $\mathbf{G}$.

The erasure correcting performance of an LT code is largely defined by its check node degree distribution $\Omega_1, \Omega_2, \dots \Omega_{k_q}$ on $\{1, 2, \dots k_q\}$, where a check node has degree d with probability Ω_d, i.e. it is connected to d distinct input nodes. The degree distribution is often described by its generating polynomial $\Omega(x) = \sum_{d=1}^{k_q} \Omega_d\, x^d$. For EEP the d connected

*Radu Lupoaie is now with QUALCOMM CDMA Technologies GmbH, Nuremberg, Germany

This work has been supported by the UMIC Research Centre, RWTH Aachen University.

input nodes are chosen uniformly at random, i.e. with probability $p = \frac{1}{k_q}$, from the set of k_q input nodes, while for UEP the input nodes are first assigned to T importance classes. The input nodes of different classes have different probabilities of being chosen to be connected to a check node. The exact UEP construction is explained in the next section.

The d non-zero entries in a row of the generator matrix $\mathbf{G}$ correspond to the weights of the d edges between a check node and d input nodes. The value of a check node is determined by adding up the product of each value of the d input nodes with the weight of the corresponding connecting edge. The non-zero entries of $\mathbf{G}$ are sampled uniformly at random from the set of $q-1$ non-zero $\mathbb{F}_q$-elements.

At the encoder, n_T output symbols are generated, which are then transmitted over a symbol erasure channel (SEC) that randomly erases some of the transmitted Galois field symbols. Finally, the decoder tries to reproduce the original k_q input symbols from the $n_\mathrm{R} \leq n_\mathrm{T}$ received symbols. Having collected n_R output symbols, the decoder uses the n_R rows of $\mathbf{G}$ that are associated with the collected, non-erased symbols to form a new matrix $\mathbf{G}'$ which is used for decoding. Since $\mathbf{G}'$ consists of a set of n_R rows sampled at random from the original matrix $\mathbf{G}$ according to the erasures that occur on the SEC, $\mathbf{G}'$ has the same statistical properties as $\mathbf{G}$.

III. LT Codes for Unequal Error Protection

First, we briefly review the original approach [6, Sec. IV] using our notation. The k_q input nodes are first assigned to T importance classes, where importance class τ has size $k_{q,\tau} = \alpha_\tau k_q$, with $1 \leq \tau \leq T$, $0 \leq \alpha_\tau \leq 1$ and $\sum_{i=1}^{T} \alpha_i = 1$, where α_τ is the relative size of class τ. According to the importance of the classes, weighting factors ω_τ are chosen such that the new initial probability of connecting an input node from class τ to the current check node is $p_\tau = \frac{\omega_\tau}{k_q} = \omega_\tau p$. Thus, $\sum_{i=1}^{T} p_i k_{q,i} = \sum_{i=1}^{T} \omega_i \alpha_i = 1$. In their finite length analysis, the number d_τ of input nodes from an arbitrary class τ that are connected to a check node of degree d is set to $\min([\alpha_\tau \omega_\tau d]\,,\, k_{q,\tau})$, where $[x]$ means rounding to the nearest integer. In general, it is desired to be able to set the weights to arbitrary non-negative values that comply with the side conditions imposed by the code parameters such that the class specific protection reaches the required levels. However, due to this rounding operation, only a discrete set of effective weights $\omega_\tau^{\text{[eff]}}$ is obtained, although the target weights ω_τ are chosen from a continuous set. The effective weights are given as a function of ω_τ:

$$\omega_\tau^{\text{[eff]}} = \frac{\bar{\Omega}_\tau}{\alpha_\tau \bar{\Omega}} = \frac{\sum_{d=1}^{d_{\max}} \Omega_d \min([\alpha_\tau \omega_\tau d]\,,\, k_{q,\tau})}{\alpha_\tau \bar{\Omega}}, \tag{1}$$

where $\bar{\Omega} = \sum_{d=1}^{k_q} d\,\Omega_d$ is the average degree of the given degree distribution $\Omega(x)$. The numerator in (1) is the effective average degree $\bar{\Omega}_\tau$ of class τ, while the denominator is the average degree of class τ in the EEP case. The discontinuities are highly dependent on the underlying degree distribution.

As an example, two binary UEP LT codes of length $k_2 = 100$ and $k_2 = 10000$ are considered that consist of two classes with relative sizes $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$ and are based on the degree distribution

$$\begin{aligned}\Omega(x) = {} & 0.007969x + 0.49357x^2 + 0.16622x^3 + 0.072646x^4 \\ & + 0.082558x^5 + 0.056058x^8 + 0.037229x^9 \\ & + 0.05559x^{19} + 0.025023x^{65} + 0.003135x^{66}\end{aligned} \tag{2}$$

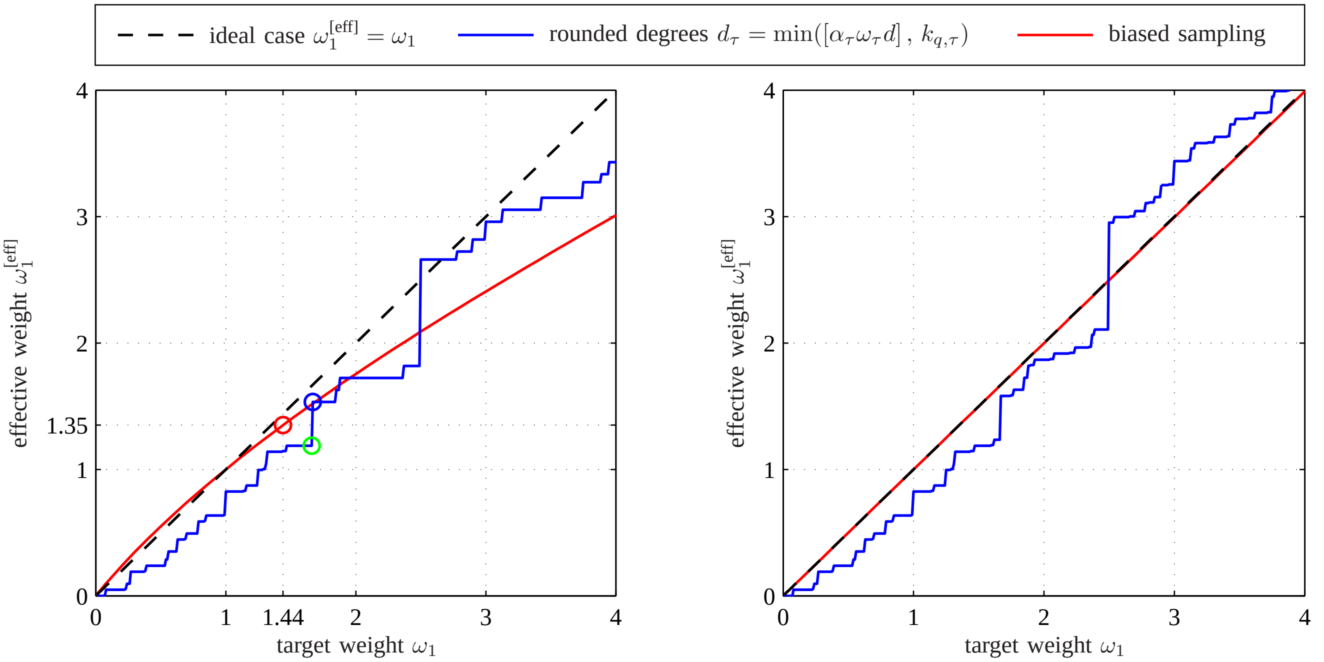
which is taken from [4]. The effective weight $\omega_1^{\text{[eff]}}$ of the respective class 1 is plotted in Fig. 1 in blue as a function of the target weight ω_1. The dashed black line indicates the optimum case $\omega_1^{\text{[eff]}} = \omega_1$. Due to the discontinuities not all effective weights can be attained with the given parameters input size k_q, class sizes $k_{q,\tau}$ and degree distribution $\Omega(x)$.

For the short code an additional deviation of the effective weight from the target weight becomes apparent in Fig. 1(a), i.e. for higher target values the effective weight deviates towards lower values. This is due to the $\min(\cdot)$ operation, which clips the number of edges connected to this class to the number of available nodes. Clipping obviously only occurs for high degrees that are in the same order of magnitude as the sizes of the involved classes. Accordingly, for the long code this effect is not visible in the depicted range of target weights (see Fig. 1(b)).

In order to prevent the discontinuous relation between ω_τ and $\omega_\tau^{\text{[eff]}}$, we propose to use biased sampling in order to select the input nodes from the different classes to be connected to the current check node of degree d. This biased sampling of the input nodes is equivalent to drawing d balls one by one without replacement from an urn that contains $k_q = \sum_{i=1}^{T} k_{q,i}$ balls of T different colours, where each ball of colour τ has weight ω_τ. The probability of picking a ball of a particular colour at a particular draw is proportional to its relative weight with respect to the total weight of the remaining balls. Biased sampling has been analysed by Wallenius [11] for the univariate case ($T = 2$) and has been generalised to the multivariate case by Chesson [12]. The partitioning of the overall degree d into *class degrees* d_τ, where $\sum_{\tau=1}^{T} d_\tau = d$, therefore follows the so-called multivariate Wallenius' noncentral hypergeometric distribution $\mathrm{mwnchypg}\left(\mathbf{d}\middle|d;\, \mathbf{k_q},\, \boldsymbol{\omega}\right)$ [11], [12]. This distribution expresses the conditional probability mass function (pmf)

$$\begin{aligned} P\left(d_1, \dots d_{T-1}\middle|d;\, \mathbf{k_q},\, \boldsymbol{\omega}\right) &= \mathrm{mwnchypg}\left(\mathbf{d}\middle|d;\, \mathbf{k_q},\, \boldsymbol{\omega}\right) \\ &= \left(\prod_{i=1}^{T} \binom{k_{q,i}}{d_i}\right) \int_0^1 \prod_{i=1}^{T} \left(1 - t^{\frac{\omega_i}{\boldsymbol{\omega}(\mathbf{k_q}-\mathbf{d})}}\right)^{d_i} dt \end{aligned} \tag{3}$$

with the vectors $\mathbf{d} = (d_1,\, d_2, \dots d_T)$ comprising the class degrees, $\mathbf{k_q} = (k_{q,1},\, k_{q,2}, \dots k_{q,T})$ comprising the class sizes and $\boldsymbol{\omega} = (\omega_1,\, \omega_2, \dots \omega_T)$ comprising the class specific target weights. This pmf can be evaluated by numerical integration as described in [13] using the BiasedUrn R package [14]. Without loss of generality, the degree d_T is not explicitly mentioned in the left-hand side of (3), since it is included



(a) Input size $k_2 = 100$ bits.

(b) Input size $k_2 = 10000$ bits.

Fig. 1. The effective weight $\omega_1^{[\text{eff}]}$ of class 1 as a function of the target weight ω_1 for two UEP LT codes of input size $k_2 = 100$ (a) and $k_2 = 10000$ (b) with two classes of relative sizes $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$ and the degree distribution in (2). The round markers are operating points used for Fig. 2.

implicitly according to $\sum_{\tau=1}^{T} d_\tau = d$. In order to simplify the notation, we omit the parametrisation with the class sizes $\mathbf{k_q}$ and weights $\boldsymbol{\omega}$ and define the joint pmf as

$$\begin{aligned} P(\mathbf{d}) &= P(d_1, \ldots d_T) = P(d_1, \ldots d_{T-1}, d) \\ &= P(d) \cdot P\left(d_1, \ldots d_{T-1} \middle| d; \mathbf{k_q}, \boldsymbol{\omega}\right), \end{aligned} \tag{4}$$

where $P(d) = \Omega_d$, i.e. the coefficients Ω_d of the check node degree distribution $\Omega(x)$.

In the remainder of this paper we will use the following simplified notation: Given an arbitrary function $f(\mathbf{d})$, the collated sum

$$\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} f(\mathbf{d}) \qquad \text{denotes} \qquad \sum_{d_1} \cdots \sum_{d_T} f(\mathbf{d}),$$

where the sums are calculated for all combinations of the values of $\mathbf{d} = (d_1, d_2, \ldots d_T)$ for which $1 \le d \le d_{\max}$ and $\sum_{\tau=1}^{T} d_\tau = d$. Additionally, $0 \le d_\tau \le \min(d, k_{q,\tau})$ with $1 \le \tau \le T$.

Using the biased sampling method, the effective weight is

$$\omega_\tau^{[\text{eff}]} = \frac{\bar{\Omega}_\tau}{\alpha_\tau \bar{\Omega}} = \frac{\sum_{d_\tau} P(d_\tau)\, d_\tau}{\alpha_\tau \bar{\Omega}} = \frac{\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} P(\mathbf{d})\, d_\tau}{\alpha_\tau \bar{\Omega}},$$

where $\bar{\Omega}_\tau$ is the average degree of class τ and $P(d_\tau)$ is obtained by marginalising $P(\mathbf{d})$.

Applied to our example codes of sizes $k_2 = 100$ and $k_2 = 10000$, the resulting effective weight $\omega_1^{[\text{eff}]}$ of the respective class 1 is plotted in Fig. 1 in red and is now a continuous function of the target weight ω_1. For the long code the effective weight even equals the target weight. In case of the short code the deviation of the effective weight from the target weight is due to the fact that the given degree distribution is not well suited for the current code and class sizes. As an example we consider a given degree $d = 66$ and weight $\omega_1 = 2$. The target degree of class 1 is then $d_1 = \alpha_1 d \omega_1 = 13.2$. However, since class 1 contains only 10 symbols, the effective degree of class 1 will be considerably smaller and thus also the effective weight. In order to compensate for this saturation effect, it is possible to (iteratively) find a weight vector that results in the original target weights. For two classes this compensation is quite simple. In order to obtain, e.g. an effective weight $\omega_1^{[\text{eff}]} = 1.35$, the actually assigned weight has to be 1.44 in case of the short example code as marked by the red circle in Fig. 1(a).

IV. Bounds on the Symbol Erasure Probability for UEP LT Codes Using Biased Sampling

In the following, we will derive a lower and an upper bound[1] on the symbol erasure probability $P_{q,\tau}^{[\text{ML,S}]}$ in importance class τ for weighted UEP LT codes over $\mathbb{F}_q$ under ML decoding, where the codes are constructed using biased sampling.

Analogously to the binary EEP case [4], a lower bound on the symbol erasure rate $P_{q,\tau}^{[\text{ML,S}]}$ is given by the probability that

[1]For notational convenience, we will implicitly assume that probabilities and their bounds are limited from above by one, i.e. the operation $\min\{1, \cdot\}$ is omitted.

an input node in class τ is not connected to a check node:

$$\underline{P}_{q,\tau}^{[\mathrm{ML,S}]} = \left(1 - \sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} P\left(\mathbf{d}\right) \frac{d_\tau}{k_{q,\tau}}\right)^{k_q \gamma_\mathrm{R}}. \quad (5)$$

The following derivation of the upper bound on the symbol erasure probability $P_{q,\tau}^{[\mathrm{ML,S}]}$ in class τ for UEP LT codes over $\mathbb{F}_q$ with biased sampling is inspired by the one in [6] for binary weighted UEP LT codes based on rounded class degrees.

Lemma 1. *Given the generator matrix* $\mathbf{G}$ *of a UEP LT code of length* k_q*, with check node degree distribution* $\Omega(x)$*,* T *importance classes of sizes* $\mathbf{k_q} = (k_{q,1}, k_{q,2}, \ldots k_{q,T})$ *and weights* $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots \omega_T)$ *that is created using biased sampling, an upper bound on the symbol erasure probability* $P_{q,\tau}^{[\mathrm{ML,S}]}$ *is given in Eq.* (6) *on the following page, where* $\gamma_\mathrm{R} = 1 + \varepsilon_\mathrm{R}$ *is the inverse reception rate. The non-zero elements in* $\mathbf{G}$ *are chosen with equal probability from* $\mathbb{F}_q \setminus \{0\}$*.*

Proof: The probability $P_{q,\tau}^{[\mathrm{ML,S}]}$ is equal to the probability that the ith $\mathbb{F}_q$-symbol cannot be determined by ML decoding for an arbitrary $i \in \boldsymbol{\tau}$, where $\boldsymbol{\tau}$ is the set of input node indices in importance class τ

$$P_{q,\tau}^{[\mathrm{ML,S}]} = \Pr\left\{\exists \mathbf{u} \in \mathbb{F}_q^{1\times k_q},\, u_i = a : \mathbf{G}'\mathbf{u}^\mathrm{T} = \mathbf{0}^\mathrm{T}\right\}, \quad (7)$$

with arbitrary but fixed $a \in \mathbb{F}_q \setminus \{0\}$. The right-hand side of (7) is the probability of the ith column of the decoding matrix $\mathbf{G}'$ being linearly dependent on a non-empty set of columns. This can be upper bounded by the probability of any possible set of columns of $\mathbf{G}'$ being linearly dependent on column $i \in \boldsymbol{\tau}$

$$P_{q,\tau}^{[\mathrm{ML,S}]} \leq \overline{P}_{q,\tau}^{[\mathrm{ML,S}]} = \sum_{\substack{\mathbf{u}\in\mathbb{F}_q^{1\times k_q},\\ u_i=a}} \Pr\left\{\mathbf{G}'\mathbf{u}^\mathrm{T} = \mathbf{0}^\mathrm{T}\right\}. \quad (8)$$

Due to the random and independent construction of check nodes, the $k_q\gamma_\mathrm{R}$ rows of $\mathbf{G}'$ can be viewed as the outcomes of independent trials of a random variable $\mathbf{r} \in \mathbb{F}_q^{1\times k_q}$, where $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_T)$ with $\mathbf{r}_j \in \mathbb{F}_q^{1\times k_{q,j}}$ and $1 \leq j \leq T$. Also the vector $\mathbf{u}$ can be expressed as $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_T)$ with $\mathbf{u}_j \in \mathbb{F}_q^{1\times k_{q,j}}$:

$$P_{q,\tau}^{[\mathrm{ML,S}]} = \sum_{\substack{\mathbf{u}\in\mathbb{F}_q^{1\times k_q},\\ u_i=a}} \left(\Pr\left\{\mathbf{r}\mathbf{u}^\mathrm{T} = 0\right\}\right)^{k_q\gamma_\mathrm{R}}. \quad (9)$$

The weight of a vector over $\mathbb{F}_q$ equals the number of non-zero elements and is denoted $|\cdot|$. Now, the probability $\Pr\left\{\mathbf{r}\mathbf{u}^\mathrm{T} = 0\right\}$ is determined, conditioned on $|\mathbf{r}_j| = d_j$ and $|\mathbf{u}_j| = w_j$, $\forall j$. A row $\mathbf{r}$ has weights $(|\mathbf{r}_1|, |\mathbf{r}_2|, \ldots |\mathbf{r}_T|) = \mathbf{d}$ with probability $P(\mathbf{d})$ as given in (4), and there are

$$(q-1)^{w-1}\left(\prod_{j=1}^{T}\binom{k_{q,j}-\delta_{\tau,j}}{w_j-\delta_{\tau,j}}\right)$$

choices of $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_T)$ of weights $\mathbf{w} = (w_1, w_2, \ldots w_T)$ with $u_i = a$ and $i \in \boldsymbol{\tau}$, where $\delta_{\tau,j}$ is the Kronecker delta function. Let $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_T) = (v_1, v_2, \ldots v_{k_q})$ with $v_l = r_l u_l$, where v_l, r_l and u_l are the lth elements of the vectors $\mathbf{v}$, $\mathbf{r}$ and $\mathbf{u}$, respectively, then we obtain Eq. (10) on the next page. The first term in (11) is

$$\Pr\left\{\sum_{j=1}^{k_q} v_j = 0 \,\middle|\, |\mathbf{v}| = s\right\} = \frac{1}{q}\left(1-(1-q)^{1-s}\right). \quad (12)$$

according to [10, Eq. (13)] while the second term in (11) is

$$\begin{aligned}\Pr&\left\{\,|\mathbf{v}| = s \,\middle|\, |\mathbf{r}| = d,\ |\mathbf{u}| = w\right\}\\ &= \prod_{j=1}^{T}\Pr\left\{|\mathbf{v}_j| = s_j \,\middle|\, |\mathbf{r}_j| = d_j,\ |\mathbf{u}_j| = w_j\right\} \quad (13)\end{aligned}$$

with the probability of occurrence of exactly s_j non-zero elements in the subvector $\mathbf{v}_j$

$$\Pr\left\{|\mathbf{v}_j| = s_j \,\middle|\, |\mathbf{r}_j| = d_j,\ |\mathbf{u}_j| = w_j\right\} = \frac{\binom{w_j}{s_j}\binom{k_{q,j}-w_j}{d_j-s_j}}{\binom{k_{q,j}}{d_j}}.$$

Reassembling all terms yields Eq. (6) on the following page which concludes the assertion. □

The resulting bounds are exemplarily illustrated in Fig. 2 for three short LT codes with degree distribution $\Omega(x)$ as given in (2), $k_2 = 100$ and two importance classes. The codes are constructed either according to the method in [6] (green and blue) or by using biased sampling (red).

V. Conclusion

In this paper we provide an analysis of finite length LT codes over higher order Galois fields $\mathbb{F}_q$ for unequal error protection (UEP) under ML decoding when the UEP property is established by assigning appropriate weights to the different importance classes. According to the weights, the edges of a check node have different probabilities to be connected to the input nodes of the different classes. In contrast to the literature, the number of input nodes in an importance class that is connected to a check node of degree d is not fixed. Instead, the connections are determined using biased sampling, which can be described by the multivariate Wallenius' noncentral hypergeometric distribution. Our main contribution is the derivation of importance class specific upper and lower bounds on the symbol erasure probability under ML decoding for the biased sampling code construction method.

Our biased sampling approach enables continuous effective UEP weights as a function of the target weights. For properly chosen degree distributions, the effective UEP weights are even equal to the target weights, while this is not the case in the approach from the literature, in which a rounding operation is used. Though this rounding operation significantly simplifies especially the computation of the upper bound on the symbol erasure probability, it introduces discontinuities in the effective weights as a function of the target weights, which leads to deviations of the protection levels of the respective importance classes from the targeted ones.

$$
\begin{aligned}
\overline{P}_{q,\tau}^{[\mathrm{ML,S}]} = & \sum_{\substack{w=w_1+\ldots+w_T=1 \\ w_\tau \geq 1}}^{k_q} (q-1)^{w-1} \left(\prod_{i=1}^{T} \binom{k_{q,i}-\delta_{\tau-i}}{w_i-\delta_{\tau-i}} \right) \\
& \cdot \left(\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} P(\mathbf{d}) \sum_{s=s_1+\ldots s_T=0}^{d} \frac{1}{q} \left(1-(1-q)^{1-s}\right) \prod_{i=1}^{T} \frac{\binom{w_i}{s_i}\binom{k_{q,i}-w_i}{d_i-s_i}}{\binom{k_{q,i}}{d_i}} \right)^{k_q \gamma_{\mathrm{R}}}
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
\overline{P}_{q,\tau}^{[\mathrm{ML,S}]} = & \sum_{\substack{w=w_1+\ldots+w_T=1 \\ w_\tau \geq 1}}^{k_q} (q-1)^{w-1} \left(\prod_{j=1}^{T} \binom{k_{q,j}-\delta_{\tau,j}}{w_j-\delta_{\tau,j}} \right) \\
& \cdot \left(\sum_{d=d_1+\ldots+d_T=1}^{d_{\max}} P(\mathbf{d}) \Pr\left\{ \mathbf{r}\mathbf{u}^{\mathrm{T}} = 0 \,\Big|\, |\,\mathbf{r}_1\,| = d_1, \ldots |\,\mathbf{r}_T\,| = d_T,\ |\,\mathbf{u}_1\,| = w_1, \ldots |\,\mathbf{u}_T\,| = w_T \right\} \right)^{k_q \gamma_{\mathrm{R}}}
\end{aligned} \tag{10}
$$

with

$$
\begin{aligned}
& \Pr\left\{ \mathbf{r}\mathbf{u}^{\mathrm{T}} = 0 \,\Big|\, |\,\mathbf{r}_1\,| = d_1, \ldots |\,\mathbf{r}_T\,| = d_T,\ |\,\mathbf{u}_1\,| = w_1, \ldots |\,\mathbf{u}_T\,| = w_T \right\} \\
& \qquad = \sum_{s=s_1+\ldots s_T=0}^{d} \Pr\left\{ \sum_{j=1}^{k_q} v_j = 0 \,\Big|\, |\,\mathbf{v}\,| = s \right\} \cdot \Pr\left\{ |\,\mathbf{v}\,| = s \,\Big|\, |\,\mathbf{r}\,| = d,\ |\,\mathbf{u}\,| = w \right\}.
\end{aligned} \tag{11}
$$

REFERENCES

[1] B. Dorsch, "Successive check digits rather than information repetition," in *IEEE International Conference on Communications (ICC)*, 1983, pp. 323–327.

[2] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Computer Communication Review*, vol. 28, pp. 56–67, 1998.

[3] M. Luby, "LT Codes," in *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.

[4] A. Shokrollahi, "Raptor Codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[5] P. Maymounkov, "Online Codes," Secure Computer Systems Group, New York University, Tech. Rep. TR2002-833, Nov. 2002.

[6] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless Codes With Unequal Error Protection Property," *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1521–1532, 2007.

[7] D. Sejdinovic, D. Vukobratovic, A. Doufexi, V. Senk, and R. Piechocki, "Expanding Window Fountain Codes for Unequal Error Protection," *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2510–2516, 2009.

[8] G. Liva, E. Paolini, and M. Chiani, "Performance versus overhead for fountain codes over $\mathbb{F}_q$," *IEEE Communications Letters*, vol. 14, no. 2, pp. 178–180, 2010.

[9] A. Shokrollahi and M. Luby, *Raptor Codes*, ser. Foundations and Trends in Communications and Information Theory. Now Publishers, 2011.

[10] B. Schotsch, R. Lupoaie, and P. Vary, "The Performance of Low-Density Random Linear Fountain Codes over Higher Order Galois Fields under Maximum Likelihood Decoding," in *49th Allerton Conference on Communication, Control and Computing*, September 2011.

[11] K. T. Wallenius, "Biased sampling: The noncentral hypergeometric probability distribution," Ph.D. dissertation, Stanford University, 1963.

[12] J. Chesson, "A non-central multivariate hypergeometric distribution arising from biased sampling with application to selective predation," *Journal of Applied Probability*, vol. 13, no. 4, pp. 795–797, 1976.

[13] A. Fog, "Calculation Methods for Wallenius' Noncentral Hypergeometric Distribution," *Communications in Statistics - Simulation and Computation*, vol. 37, no. 2, pp. 258–273, 2008.

[14] ——, "BiasedUrn: Biased Urn Model Distributions," 2011, Version 1.04. [Online]. Available: http://cran.r-project.org/web/packages/BiasedUrn/index.html

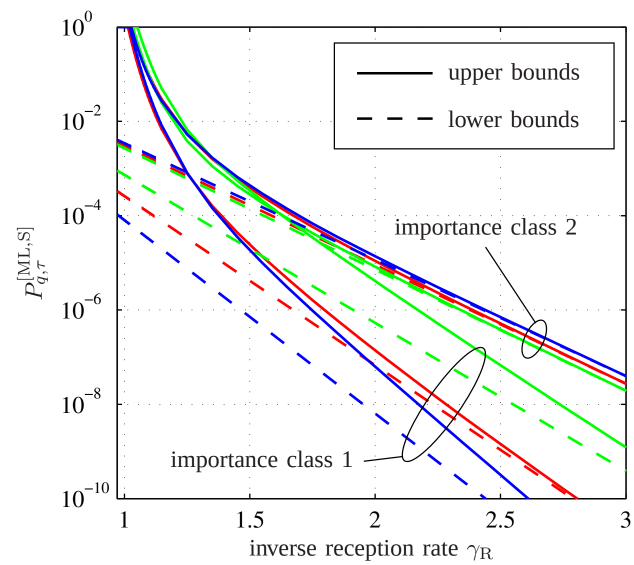


Fig. 2. Upper and lower bounds for three UEP LT codes with $k_2 = 100$, two importance classes and relative class sizes of 0.1 and 0.9. An effective weight $\omega_1^{[\mathrm{eff}]} = 1.35$ shall be obtained. The weights ω_1 that yield the closest values to $\omega_1^{[\mathrm{eff}]}$ for the different code construction methods are highlighted in Fig. 1(a) by round markers with corresponding colours. Using the rounded degrees method, the envisaged $\omega_1^{[\mathrm{eff}]} = 1.35$ cannot be reached. The closest one can get is by using either $\omega_1 = 1.66$ ($\omega_1^{[\mathrm{eff}]} = 1.187$, green curves) or $\omega_1 = 1.67$ ($\omega_1^{[\mathrm{eff}]} = 1.534$, blue curves). In one case (blue), class 1 is protected too well at the cost of class 2, while in the other case (green), class 1 is not protected sufficiently. With this method, the protection levels between the green and the blue curves are not achievable with the given code parameters. Using the biased sampling method, however, any effective weight and thus any protection level can be obtained, in this case by setting $\omega_1 = 1.44$ (red curves).