THEORY OF COMPUTING, Volume 14 (4), 2018, pp. 1–29
*www.theoryofcomputing.org*

# Explicit Rateless Codes for Memoryless Binary-Input Output-Symmetric Channels

Benny Applebaum* Liron David† Guy Even

*Received September 8, 2015; Revised October 29, 2017; Published April 26, 2018*

**Abstract:** A rateless code encodes a finite-length information word into an infinitely long codeword such that longer prefixes of the codeword can tolerate a larger fraction of errors. A rateless code approaches capacity for a family of channels if, for every channel in the family, reliable communication is obtained by a prefix of the code whose rate is arbitrarily close to the channel's capacity. The encoder is universal in the sense that same encoder is used for all channels in the family.

So far, all known constructions of rateless codes were randomized, giving rise to *ensembles* of such codes. In this paper, we construct the first *explicit* rateless code for memoryless binary-input output-symmetric (MBIOS) channels. Our code achieves an almost exponentially small error probability (e. g., $\exp(-\Omega(k/\log^* k))$ for $k$-bit information word), and can be encoded in almost constant time per-bit (e. g., $O(\log^* k)$). Over binary symmetric channels, the running time of decoding is similar. Previous ensemble-based rateless codes for the binary symmetric channel have polynomial asymptotic error probabilities and the running time of decoding is polynomial only under some conditions.

A preliminary version of this paper appeared in the Proceedings of the 6th Conference on Innovations in Theoretical Computer Science, ITCS 2015 [1].

*Supported by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC, by an ICRC grant and by the Check Point Institute for Information Security.

†Supported by Israel Ministry of Science and Technology (grant 3-9094).

**ACM Classification:** E.4

**AMS Classification:** 94B05

**Key words and phrases:** coding theory, error-correcting codes, rateless codes, memoryless binary-input output symmetric channel, binary symmetric channel, Gaussian channel


DOI: 10.4086/toc.2018.v014a004

Our main technical contribution is a constructive proof for the existence of an infinite generating matrix with the property that each of its prefixes induces a weight distribution that approximates the expected weight distribution of a random linear code.

# 1 Introduction

Consider a transmitter who wishes to send an information word $\mathbf{m} \in \{0,1\}^k$ over a Binary Symmetric Channel with crossover probability $p \in (0,1/2)$ (hereafter denoted as $\mathsf{BSC}(p)$). By Shannon's theorem, using an error-correcting code, reliable communication can be achieved by encoding the information word into an $n$-bit codeword, where $n = k/(1-H(p)-\delta)$. Here $H(p)$ denotes the binary entropy function, and $\delta > 0$ is an arbitrarily small constant. Furthermore, there are explicit capacity-achieving codes in which decoding and encoding can be performed efficiently in polynomial or even linear time (cf. [5, 6]).

The task of reliable communication is more challenging when the transmitter does not know the noise level $p$. Such a situation arises, for example, when the transmitter wishes to broadcast the same message to many receivers $B_1, B_2, \ldots$, where each receiver $B_i$ experiences a different crossover probability $p_i$ (e. g., due to a different distance from the transmitter). Naively, one would use a code which is tailored to the noisiest channel with parameter $p_{\max}$. However, this solution adds an unnecessary communication overhead for receivers $B_i$ with $p_i < p_{\max}$. Furthermore, in absence of an upper bound $p_{\max} < 1/2$, even this naive solution is not possible.

The problem of encoding without knowing the channel's noise level (also studied in [7, 30]) can be solved by a *rateless code*. In a rateless code, the encoder maps the information word $\mathbf{m} \in \{0,1\}^k$ into an infinitely long sequence of bits $\{c_i\}_{i\in\mathbb{N}}$ such that the longer the prefix of the codeword, the higher level of noise can be corrected. That is, for every value of $p \in (0,1/2)$, and every constant gap-to-capacity $\delta > 0$, reliable communication is achieved over $\mathsf{BSC}(p)$ based on a prefix of length $k/(1-H(p)-\delta)$ (as $k$ tends to infinity).

Rateless codes have been extensively studied under various names [18, 16, 9, 14, 7, 30, 24, 8, 13, 26, 15, 22, 23]. Information-theoretically, the problem of rateless transmission is well understood [28], and, for many noise models, random codes achieve nearly-optimal rate. The task of constructing computationally efficient rateless codes, which provide polynomial-time encoding and decoding, is much more challenging. Currently, only a few examples of efficient capacity-achieving rateless codes are known for several important channels such as erasure channels, Gaussian channels, and binary symmetric channels [17, 27, 10, 20]. Interestingly, all known constructions are probabilistic in the sense that the code is chosen randomly from an ensemble. Hence, the encoder and the decoder must share randomness (e. g., by an error-free side-channel or public randomness). This limitation raises the natural question of whether one can construct *explicit* rateless codes.

## 1.1 Our results

In this paper, we answer the question in the affirmative by constructing deterministic efficient rateless codes which achieve the capacity over Memoryless Binary-Input Output-Symmetric (MBIOS) channels. To simplify the exposition, we state the results for the special case of the BSC family. (A generalization to arbitrary MBIOS channels appears in Section 3.)

**Theorem 1.1** (Rateless codes for BSC). *Fix some super-constant function $\beta(k) = \omega(1)$. There exists a deterministic rateless encoding algorithm* Enc *and a deterministic rateless decoding algorithm* Dec *such that for every $k$-bit information word* **m** *the following holds.*

- ***(Capacity achieving)*** *For every crossover probability $p \in (0, 1/2)$, and prefix length*

$$n = k \cdot \frac{1}{1 - H(p) - \delta}$$

*where $0 < \delta < 1 - H(p)$ is an arbitrary constant, the $n$-long prefix of $\mathsf{Enc}(\mathbf{m})$ is decoded correctly by* Dec *over $\mathsf{BSC}(p)$ with probability $1 - 2^{-\Omega(k/\beta^3)}$.*

- ***(Efficiency)*** *For every $n > k$, the $n$-long prefix of $\mathsf{Enc}(\mathbf{m})$ can be computed in time $n \cdot \beta$, and decoding is performed in time $n \cdot \beta$. Both algorithms can be implemented in parallel by circuits of depth $O(\beta + \log n)$.*

Letting $\beta$ be a slowly increasing function, (e.g, $\log^*(k)$) we obtain an "almost" exponential error and "almost" linear time encoding and decoding.

One may also consider a weaker form of capacity achieving rateless codes in which the encoding is allowed to depend on the gap to capacity $\delta$. (This effectively puts an a priori upper-bound on the noise probability which makes things easier.) In this setting we can obtain an asymptotically optimal construction with linear time encoding and decoding and exponentially small error. (See Section 3 for a formal statement.)

**Comparison with spinal codes.** Prior to our work, *spinal codes* [19, 20, 2] were the only known efficient (randomized) rateless codes for the BSC.[1] Apart from being deterministic, our construction has several important theoretical advantages over spinal codes. The upper bound on the decoding error of spinal codes is only inverse polynomial in $k$, and these codes only weakly achieve the capacity (i. e., the encoding depends on the gap $\delta$ to capacity). Moreover, the decoding complexity is polynomial (as opposed to linear or quasilinear in our codes), and both encoding and decoding are highly sequential as they require $\Omega(k)$ sequential steps. It should be mentioned however that, while spinal codes were reported to be highly practical, we currently do not know whether our codes perform well in practice. The question of directly derandomizing spinal codes (while keeping high practical performance) is left for future research.

## 1.2 Overview of our construction

Our starting point is a simple (but inefficient and randomized) construction based on a random linear code. Assume that both the encoder and decoder have access to an infinite sequence of random $k$-bit row vectors $\{R_i\}_{i\in\mathbb{N}}$. To encode the message $\mathbf{m} \in \{0,1\}^k$, viewed as a $k$-bit column vector, the encoder sends the sequence $\{R_i \cdot \mathbf{m}\}_{i\in\mathbb{N}}$ of inner products over the binary field. To decode a noisy $n$-bit prefix of the codeword, we will employ the maximum-likelihood decoder (ML) for the code generated by the $n \times k$

[1] We emphasize that, despite the use of the term "de-randomization" in [2], the construction of spinal codes is *randomized*. In particular, it employs a hash function chosen uniformly from a family of pair-wise independent hash functions.

matrix $R = (R_1, \dots, R_n)$. A classical result in coding theory asserts that such a code achieves the capacity of the BSC. Namely, as long as the gap from capacity $\delta = 1 - H(p) - k/n$ is positive, the random matrix $R \xleftarrow{R} \{0,1\}^{n \times k}$ is likely to generate a code with exponentially small (in $k$) Maximum Likelihood Decoding Error over $\mathsf{BSC}(p)$.

This construction has two important drawbacks: It is probabilistic and it does not support efficient decoding. For now, let us ignore computational limitations, and attempt to derandomize the construction.

### 1.2.1 Derandomization

We would like to deterministically generate an infinite number of rows $\{R_i\}_{i \in \mathbb{N}}$ such that every $n$-row prefix matrix $R[1:n] = (R_1, \dots, R_n)$ has a low ML-decoding error of, say 0.01, for every $p$ for which $1 - H(p) - k/n$ is greater than, say, 0.01.[2]

Although we know that, for every $n$, almost all $n \times k$ matrices satisfy this condition, it is not a priori clear that every such low-error matrix can be extended to a larger matrix while preserving low error.

To solve this problem, we identify a property of *good* matrices which, on the one hand, guarantees low decoding error, and, on the other hand, is *extendable* in the sense that every good matrix can be augmented by some row while preserving its goodness. We will base our notion of goodness on the *weight distribution* of the matrix $R$.

Let $W_{i,n}$ denote the set of information words which are mapped by the matrix $R[1:n]$ to codewords of Hamming weight $i$, and let $w_{i,n}$ denote the size of this set. The sets $(W_{1,n}, \dots, W_{n,n})$ form a partition of $\{0,1\}^k$, and the vector $(w_{i,n})_{i=1,\dots,n}$ is called the weight distribution of the code. When a row $R_{n+1}$ is added, the weight of all information words which are orthogonal to $R_{n+1}$ remains the same, while the weight of non-orthogonal words grows by 1. Thus $R_{n+1}$ splits $W_{i,n}$ into two parts: the orthogonal vectors which "remain" in $W_{i,n+1}$, and the non-orthogonal vectors which are "elevated" to $W_{i+1,n+1}$. A random row $R_{n+1}$ is therefore expected to split $W_{i,n}$ into two *equal* parts.

If in each step we could choose such an "ideal" row which simultaneously halves each set $W_{i,n}$, we would get an "ideal" weight distribution in which $w_i^*(n,k) = \binom{n}{i} \cdot 2^{k-n}$, as expected in a random linear code. Such an ideal weight distribution guarantees a low ML decoding error over $\mathsf{BSC}(p)$ when $1 - H(p) < k/n$ (cf. [21, 29, 4]).

While we do not know how to choose such an ideal row (in fact it is not clear that such a row exists), a probabilistic argument shows that we can always find a row $R_{n+1}$ which approximately splits every sufficiently large $W_{i,n}$ simultaneously. Furthermore, by keeping track of the small sets and choosing $R_{n+1}$ which elevates a constant fraction of the lightest vectors, we make sure that the distance of the code is not too small, e. g., $W_{i,n}$ is empty for all $i < \Omega((n-k)/\log n)$. Using these properties we show that the resulting code has low ML decoding error. (See Section 4.)

### 1.2.2 Making the code efficient

The above approach gives rise to a deterministic rateless code which achieves the capacity of the BSC with a subexponential error of $\varepsilon = 2^{-\Omega(\beta/\log\beta)}$ where $\beta$ is the length of the information word. However,

[2] We use small constants to simplify the presentation, the discussion remains valid when the constants are replaced with a function that decreases with $k$.

the time complexity of encoding/decoding the $n$-bit prefix of a codeword is $n\cdot 2^{O(\beta)}$. We solve this problem by noting that Forney's concatenation technique [11] naturally extends to the rateless setting. We sketch the construction below. (Full details appear in Section 6, see also Figure 1 and Figure 2.)

Let $k$ denote the length of the information word. The construction uses the inefficient rateless code as an "inner code" $C_{\mathsf{in}}:\{0,1\}^{\beta}\to\{0,1\}^{*}$, and, in addition, employs a standard efficient outer code $C_{\mathsf{out}}:B^{k_{\mathsf{out}}}\to B^{n_{\mathsf{out}}}$ where $B\triangleq\{0,1\}^{\beta}$ and $k_{\mathsf{out}}\triangleq k/\beta$.

To encode a message $\mathbf{m}\in\{0,1\}^{k}$, we parse it as $M\in B^{k_{\mathsf{out}}}$, apply the outer code to obtain a codeword $C\triangleq(C_1,\ldots,C_{n_{\mathsf{out}}})$ and then apply the inner code to each of the symbols of $C$ in parallel. Namely, each symbol $C_i$ is encoded by the code $C_{\mathsf{in}}$ to an infinitely-long column vector. The $n_{\mathsf{in}}\cdot n_{\mathsf{out}}$ prefix of the concatenated encoding is obtained by collecting the binary vectors $(X_1,\ldots,X_{n_{\mathsf{out}}})$ where $X_i$ denotes the prefix of length $n_{\mathsf{in}}$ of the inner codeword that corresponds to $C_i$.

Decoding proceeds in the natural way. Let $Y=(Y_1,\ldots,Y_{n_{\mathsf{out}}})$ denote the noisy $n_{\mathsf{in}}\cdot n_{\mathsf{out}}$ prefix of the encoding of the message $m$. First, maximum likelihood decoding is employed to decode each of the inner codewords $Y_i$ into $\hat{X}_i$. Next, the decoder of the outer code recovers an information word $M$ from the noisy codeword $(\hat{X}_1,\ldots,\hat{X}_{n_{\mathsf{out}}})$.

In order to prove Theorem 1.1, we need a somewhat non-standard setting of the parameters. To avoid having to fix the gap to the channel's capacity ahead of time, we use an outer code whose rate tends to 1 (i. e., $n_{\mathsf{out}}=k_{\mathsf{out}}(1+o(1))$). Set $\beta=\omega(1)$. For concreteness, take an outer code $C_{\mathsf{out}}:B^{k_{\mathsf{out}}}\to B^{n_{\mathsf{out}}}$ with $n_{\mathsf{out}}=k_{\mathsf{out}}+k_{\mathsf{out}}/\mathrm{poly}(\beta)$, and assume that the code can be decoded from a fraction of $\varepsilon'=\Omega(1/\mathrm{poly}(\beta))$ errors in time $n_{\mathsf{out}}\cdot\mathrm{poly}(\beta)$ and can be encoded with similar complexity.[3] A standard application of Chernoff's bound shows that the decoding error of $p$-noisy codeword of length $n\geq k\cdot 1/(1-H(p)-\delta)$, is $2^{-\Omega(n_{\mathsf{out}}(\varepsilon'-\varepsilon)^2)}$, which, under our choice of parameters, simplifies to $2^{-\Omega(k/\mathrm{poly}(\beta))}$. For a slowly increasing $\beta=\omega(1)$, we derive an almost-exponential error, and an almost linear encoding/decoding time complexity of $n_{\mathsf{out}}\cdot\beta+n\cdot 2^{O(\beta)}$.

If the gap-to-capacity $\delta$ is a priori given, we can take $\beta$ to be a constant (which depends on $\delta$). As a result the rate of the outer code is bounded away from 1, but the error becomes exponentially small and both encoding and decoding can be performed in linear time.

**Remark 1.2** (An alternative approach)**.** As already mentioned, our analysis shows that a good approximation of the weight distribution guarantees low decoding error. In particular, to achieve the error bounds of Theorem 1.1, it suffices to design a code whose $n$-bit prefix has (1) distance of $\Omega((n-k)/\log n)$, and (2) has at most $w_i^*(n,k)\cdot 2^{o(n)}+\mathrm{poly}(n)$ codewords of weight $i$. Once $n>2^k$ these conditions become trivial to satisfy: The first condition requires distance of $\Omega((n-k)/k)$ (i. e., every $k$ rows should increase the distance by 1) and the second condition becomes vacuous (since the total number of codewords is at most $2^k<\mathrm{poly}(n)$). In particular, if the the first $2^k$ rows of the generating matrix are "good," they can be easily extended by concatenating the identity matrix over and over again. This observation (suggested to us by an anonymous referee) implies that a rateless code can be deterministically constructed in time double-exponential in $k$ by enumerating over all $2^k\times k$ matrices. Of course, this is significantly slower than our construction which (naively) achieves exponential complexity in $k$ and, after concatenation, yields an "almost linear-time" construction.

[3] Such a code can be obtained based on expander graphs, e. g., [32, 31, 12]. In fact, we will employ the code of [12] which achieves a smaller alphabet of absolute size $\beta$. This is not a real issue as we can increase the alphabet to $2^{\beta}$ by parsing $\beta/\log\beta$ symbols as a single symbol without affecting the properties of the code. See Section 6.

### 1.3 Discussion

One of the main conceptual contributions of this work is a formalization of rateless codes from an algorithmic point of view (see Section 2.2). This formulation raises a more general research problem:

> Is it possible to gradually generate an infinite combinatorial object $\mathcal{O} = \{\mathcal{O}_i\}_{i=1}^{\infty}$ via a deterministic algorithm?

Note that the question may be interesting even for inefficient algorithms as it may be infeasible, in general, to decide whether a finite sequence $\mathcal{O}_1, \ldots, \mathcal{O}_n$ is a prefix of some good infinite sequence $\mathcal{O}$. (This is very different than the standard finite setting, where inefficient derandomization is trivially achievable by exhaustive search.) It will be interesting to further explore other instances of this question (e. g., for some families of graphs).

The problem of deterministically constructing a rateless code can be formulated as follows. Refer to a generating matrix as "pseudo-random-weight" if the weight distribution of the code it generates is "close" to the expected weight distribution of random linear codes. Our main technical contribution is a deterministic construction of an infinite generating matrix, every finite prefix of which is "pseudo-random-weight."

An interesting open problem is to obtain stronger approximations for the "ideal" weight distribution. Specifically, it should be possible to improve the code's distance from sublinear ($\Omega((n-k)/\log n)$) to linear ($\Omega((n-k))$) in the redundancy. Such a code will be able to resist adversarial errors (e. g., in Hamming model). It is also natural to try and approximate the ideal weight distribution not only for prefixes but also for consecutive blocks. Namely, is it possible to construct a rateless code which, for every restriction to $n$ consecutive bits, achieves the capacity of the BSC? Getting back to our motivating story of noisy multicast, such a rateless code would allow the receivers to dynamically join the multicast.

## 2 Preliminaries

### 2.1 MBIOS Channels

Let $\mathbf{x} \triangleq (x_1, \ldots, x_k)$ denote a $k$-bit string. Let $W : \{0,1\} \to \mathsf{Y}$ denote a binary-input output-symmetric channel, where $\mathsf{Y}$ denotes the output alphabet of a channel.[4] Let $W^n : \{0,1\} \to \mathsf{Y}^n$ denote the memoryless binary-input output-symmetric (MBIOS) channel obtained by $n$ independent copies of $W$. Let $W^n(\mathbf{z})$ denote the random variable (taking values in $\mathsf{Y}^n$) that describes the channel's output when input $\mathbf{z} \in \{0,1\}^n$. An example of an MBIOS channel is the binary symmetric channel with crossover probability $p \in (0,1/2)$ (denoted by $\mathsf{BSC}(p)$).

### 2.2 Rateless codes

We begin with a syntactic definition of rateless codes over binary-input channels. We denote the output alphabet by $\mathsf{Y}$.

[4] A binary-input channel is output-symmetric if there is a bijection $\pi : \mathsf{Y} \to \mathsf{Y}$ such that $\Pr(W(b) = y) = \Pr(W(1-b) = \pi(y))$, for every $y \in \mathsf{Y}$ and $b \in \{0,1\}$.

**Definition 2.1** (rateless code)**.** A rateless code is a pair of functions, $(\mathsf{Enc},\mathsf{Dec})$, satisfying the following conditions.

1. The encoder $\mathsf{Enc} : \{0,1\}^* \times \mathbb{N} \to \{0,1\}$ takes an information word $\mathbf{x} \in \{0,1\}^*$ and an index $i \in \mathbb{N}$, and outputs the $i$-th bit of the encoding of $\mathbf{x}$. (Equivalently, the encoding of $\mathbf{x}$ is an infinite sequence of bits $(\mathsf{Enc}(\mathbf{x},i))_{i\in\mathbb{N}}$.)
2. The decoder $\mathsf{Dec} : \mathsf{Y}^* \times \mathbb{N} \to \{0,1\}^*$ maps a noisy codeword $\mathbf{y} \in \mathsf{Y}^*$ and an integer $k$ (which corresponds to the length of the information word) to an information word $\hat{\mathbf{x}} \in \{0,1\}^k$.

In our construction, the encoder and the decoder are computed by deterministic algorithms. Constructions based on ensembles require an additional input whose purpose is to serve as shared randomness that selects the specific code from the ensemble. (See Section 2.3 for a formal definition.)

**Conventions.** We let $\mathsf{Enc}(\mathbf{x},[1:n])$ denote the first $n$ bits of the codeword that encodes the information word $\mathbf{x}$. Namely, $\mathsf{Enc}(\mathbf{x},[1:n])$ is the binary string $\mathbf{c} = (c_1,\ldots,c_n)$, where $c_i = \mathsf{Enc}(\mathbf{x},i)$. A rateless code defines $(n,k)$ codes for every $n$ and $k$ via

$$C_{n,k} \triangleq \{\mathsf{Enc}(\mathbf{x},[1:n]) \mid \mathbf{x} \in \{0,1\}^k\}. \tag{2.1}$$

When the information word length is clear from the context, we abbreviate $\mathsf{Dec}(\mathbf{y},k)$ to $\mathsf{Dec}(\mathbf{y})$.

**Remark 2.2** (Additional features.)**.** In some scenarios it is beneficial to have a rateless code with the following additional features.

- (Linearity) A rateless code is *linear* if $\mathsf{Enc}$ is a linear function. Namely, for $\mathbf{x} \in \mathrm{GF}(2)^k$, we have
$$\mathsf{Enc}(\mathbf{x},i) = R_i \cdot \mathbf{x},$$
where $\{R_i\}_{i=1}^{\infty}$, is an infinite sequence of row vectors $R_i \in \mathrm{GF}(2)^k$. We refer to the infinite matrix $G = \{R_i\}_{i=1}^{\infty}$ as the *generator* matrix of the code.[5]
- (Systematic) An encoding is *systematic* if, for every $\mathbf{x} \in \{0,1\}^k$, we have $\mathsf{Enc}(\mathbf{x},[1:k]) = \mathbf{x}$.

We define the (maximum) *error function* of a rateless code $(\mathsf{Enc},\mathsf{Dec})$ over a channel $W$ with respect to information word length $k$ and codeword length $n$ by

$$P_{\mathsf{err}}(W,\mathsf{Enc},\mathsf{Dec},k,n) \triangleq \max_{\mathbf{x}\in\{0,1\}^k} \Pr[\mathsf{Dec}(W^n(\mathsf{Enc}(\mathbf{x},[1:n])),k) \neq \mathbf{x}].$$

Let $C(W)$ denote the Shannon capacity of a channel $W$. Let $\mathcal{F}$ denote a family of MBIOS channels.

**Definition 2.3** (capacity-achieving rateless code with a universal encoder and decoder)**.** A rateless code $(\mathsf{Enc},\mathsf{Dec})$ achieves capacity with respect to a family of channels $\mathcal{F}$ if, for every channel $W \in \mathcal{F}$ and every $\delta \in (0, C(W))$, if $n(k) \triangleq k/(C(W)-\delta)$, then

$$\lim_{k\to\infty} P_{\mathsf{err}}(W,\mathsf{Enc},\mathsf{Dec},k,n(k)) = 0. \tag{2.2}$$

[5] We use the convention that the generating matrix of $C_{n,k}$ is an $n \times k$ matrix $G$ and the information word $\mathbf{x}$ is a column vector in $\mathrm{GF}(2)^k$. A codeword $\mathbf{y}$ is obtained by the multiplication $\mathbf{y} = G\mathbf{x}$.

In Definition 2.3, we assume that neither the encoder nor the decoder knows which channel $W$ is chosen from $\mathcal{F}$. One can consider the variation of this definition in which only the encoder is oblivious to the channel. In such a variation, the decoder depends on $W$. To emphasize this dependency we denote the decoder by $\mathsf{Dec}_W$.

**Definition 2.4** (capacity-achieving <– rateless code with a universal encoder and channel dependent decoder)**.** A rateless code $(\mathsf{Enc}, \{\mathsf{Dec}_W\}_{W\in\mathcal{F}})$ achieves capacity if, for every channel $W \in \mathcal{F}$ and every $\delta \in (0, C(W))$, if $n(k) \triangleq k/(C(W)-\delta)$, then

$$\lim_{k\to\infty} P_{\mathsf{err}}(W, \mathsf{Enc}, \mathsf{Dec}_W, k, n(k)) = 0\,. \tag{2.3}$$

## 2.3 Randomized rateless codes

For the sake of future reference, we extend the definitions of rateless codes to the setting of randomized constructions. (These definitions are not used in this paper since all our constructions are deterministic, and the reader may safely skip them.) Roughly speaking, we assume that to encode/decode the $n$-bit prefix of the codeword the encoder and the decoder consume a shared random string $\rho$ of length $\ell(k,n)$ where $k$ is the length of the information word. (Equivalently, the two algorithms have an access to an infinite string and they use its $\ell(k,n)$ prefix to generate/decode the $n$-bit prefix of the code.) The error probability is measured naturally with respect to a randomly chosen string $\rho$. We proceed with formal definitions. Below we consider a binary-input channel with an output alphabet $\mathsf{Y}$.

**Definition 2.5** (randomized rateless code)**.** A randomized rateless code with randomness complexity of $\ell : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ is a pair of functions $(\mathsf{Enc}, \mathsf{Dec})$ such that:

1. The encoder $\mathsf{Enc} : \{0,1\}^* \times \mathbb{N} \times \{0,1\}^* \to \{0,1\}$ takes an information word $\mathbf{x} \in \{0,1\}^*$ an index $i \in \mathbb{N}$, and a random string $\rho \in \{0,1\}^{\ell(|\mathbf{x}|,i)}$ and outputs the $i$-th bit of the encoding of $\mathbf{x}$.

2. The decoder $\mathsf{Dec} : \mathsf{Y}^* \times \mathbb{N} \times \{0,1\}^* \to \{0,1\}^*$ maps a noisy codeword $\mathbf{y} \in \mathsf{Y}^*$ an integer $k$ (which corresponds to the length of the information word) and a random string $\rho \in \{0,1\}^{\ell(k,|\mathbf{y}|)}$ to an information word $\hat{\mathbf{x}} \in \{0,1\}^k$.

We always assume that the randomness complexity function $\ell : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ is monotone increasing in both arguments.

We let $\mathsf{Enc}_\rho(\mathbf{x}, [1:n])$ denote the first $n$ bits of the codeword that encodes the information word $\mathbf{x}$ using the (same) random string $\rho \in \{0,1\}^{\ell(|\mathbf{x}|,n)}$. Namely, $\mathsf{Enc}_\rho(\mathbf{x}, [1:n])$ is the binary string $\mathbf{c} = (c_1, \ldots, c_n)$, where $c_i = \mathsf{Enc}(\mathbf{x}, i, \rho[1:\ell(k,i)])$. We also write $\mathsf{Dec}_\rho(\mathbf{y}, k)$ to denote $\mathsf{Dec}(\mathbf{y}, k, \rho)$.

We define the *error function* of a randomized rateless code $(\mathsf{Enc}, \mathsf{Dec})$ over a channel $W$ with respect to information word length $k$ and codeword length $n$ by

$$P_{\mathsf{err}}(W, \mathsf{Enc}, \mathsf{Dec}, k, n) \triangleq \max_{\mathbf{x}\in\{0,1\}^k} \Pr_\rho[\mathsf{Dec}_\rho(W^n(\mathsf{Enc}_\rho(\mathbf{x}, [1:n])), k) \neq \mathbf{x}]\,,$$

where the probability is taken over the distribution of the channel and over a uniform choice of $\rho \xleftarrow{R} \{0,1\}^{\ell(k,n)}$. A randomized rateless code $(\mathsf{Enc}, \mathsf{Dec})$ achieves capacity with respect to a family

$\mathcal{F}$ of channels with a universal encoder and decoder (or a universal encoder and channel-dependent decoder) if its error function satisfies Definition 2.3 (Definition 2.4, resp.).

One may study variants of the above definition in which the encoder/decoder have some limited access to the randomness. For example, it may be the case that the infinite random string is partitioned into short blocks and the $i$-th bit of the codeword can be generated solely based on the $i$-th block. (The encoder does not have to remember "old parts" of the random string.) Indeed, this property is satisfied by the "random matrix construction" in which the $i$-th bit is generated by sending the inner product of $\mathbf{x}$ with a random $k$-bit vector $R_i$. The study of other variants is left for future research.

# 3 Our results

We present a capacity-achieving rateless code for MBIOS channels with a universal encoder. The code is parameterized by a parameter $\beta = \beta(k)$ which is an arbitrary slowly growing function of the information word length $k$. The decoder $\mathsf{Dec}_W$ of a MBIOS channel $W$ uses maximum likelihood (ML) decoding as a subroutine. (However, computational efficiency is obtained by applying ML decoding only to short blocks of the inner code of a concatenated code.) Let $\mathcal{MBIOS}$ denote the family of all MBIOS channels.

**Theorem 3.1** (Main Theorem). *There exists a systematic linear rateless code $C^\beta = (\mathsf{Enc}, \{\mathsf{Dec}\}_{W\in\mathcal{MBIOS}})$ such that for every channel $W \in \mathcal{MBIOS}$ and every $\delta \in (0, C(W))$, the following hold.*

1. *If $\beta(k)$ tends to infinity with $k$, then*

$$P_{\mathsf{err}}\left(W, \mathsf{Enc}, \mathsf{Dec}_W, k, \frac{k}{C(W)-\delta}\right) = 2^{-\Omega(k/\beta^3)}.$$

2. *Furthermore, if $\beta$ and $k$ are fixed and $n$ tends to infinity then*

$$P_{\mathsf{err}}(W, \mathsf{Enc}, \mathsf{Dec}_W, k, n) = 2^{-\Omega(n/\log(n))}.$$

*The running time of the encoder is $O(\beta \cdot 2^{2\beta})$ per codeword bit and can be computed by a circuit of depth $O(\beta + \log n)$. The running time of the decoder is $O(\beta + (R/\beta) \cdot T_W^{ML}(\beta/R))$ per codeword bit, where $R = k/n$ and $T_W^{ML}(\ell)$ denotes the running time of ML-decoding over the channel $W$ of a linear $[\ell, \beta]$-code. Decoding can be computed in parallel by a circuit of depth $O(PT_W^{ML}(\beta/R)) + \beta + \log n)$, where $PT_W^{ML}(\ell)$ denotes the parallel time of ML-decoding over the channel $W$ of a linear $[\ell, \beta]$-code.*

The "furthermore" part shows that the error decreases quickly even for fixed length inputs (and fixed $\beta$) provided that the prefix is sufficiently long. If the gap-to-capacity $\delta$ is fixed then the construction can be tweaked to work with some constant $\beta = \beta(\delta)$. The resulting error function decreases exponentially as follows.

**Corollary 3.2.** *For every fixed $\delta > 0$, there exists constant $\beta$ such that, for every $W \in \mathcal{MBIOS}$ with $C(W) > \delta$, the rateless code $C^\beta$ satisfies*

$$P_{\mathsf{err}}\left(W, \mathsf{Enc}, \mathsf{Dec}_W, k, \frac{k}{C(W)-\delta}\right) = 2^{-\Omega(k)}. \tag{3.1}$$

*The running time of the encoder is constant per codeword bit and $O(\log n)$ in parallel. The running time of the decoder is*

$$O((C(W)-\delta)\cdot T_W^{ML}\left(O\left(\frac{1}{C(W)-\delta}\right)\right)$$

*per codeword bit.*

**Universal Decoding.** We remark that Theorem 3.1 and Corollary 3.2 can be restricted to families of MBIOS channels $\mathcal{F}$ in which ML-decoding does not depend on the channel $W \in \mathcal{F}$. Examples of such families are: (1) $\mathsf{BSC}(p)$, for $p \in (0,1/2)$, where ML-decoding finds a closest codeword in the $L_1$ distance (Hamming distance). (2) Binary-Input Additive White Gaussian Noise with noise variance $\sigma^2$ (AWGN($\sigma$)) for $\sigma > 0$, where ML-decoding finds a closest codeword in the $L_2$ distance (Euclidean distance). As the dependency of our decoder $\mathsf{Dec}_W$ on the channel $W$ stems only from the need to compute ML-decoding, it follows that in such families of channels the decoder is universal and does not depend on the channel.

# 4 Inefficient explicit rateless code for BSC

In this section we present a computationally inefficient[6] explicit construction of a rateless code that achieves capacity with respect to the family of all binary symmetric channels $\mathsf{BSC}(p)$, where $p \in (0,1/2)$. The decoder in this construction is universal and is simply a maximum-likelihood decoder (i. e., minimizes the $L_1$-distance). This code will be later used as the inner code of our final construction.

**Theorem 4.1.** *There exists a systematic linear rateless code* $(\mathsf{Enc},\mathsf{ML})$ *with the following properties.*

**Capacity achieving and decaying error:** *For every* $p \in (0,1/2)$ *and* $\delta \in (0, C(\mathsf{BSC}(p)))$,

$$P_{\mathsf{err}}\left(\mathsf{BSC}(p),\mathsf{Enc},\mathsf{ML},k,\frac{k}{C(W)-\delta}\right) = 2^{-\Omega(k/\log k)} \qquad \text{(as } k \to \infty\text{)}, \tag{4.1}$$

$$P_{\mathsf{err}}(\mathsf{BSC}(p),\mathsf{Enc},\mathsf{ML},k,n) = 2^{-\Omega(n/\log(n))} \qquad \text{(fixed } k,\ n \to \infty\text{)}. \tag{4.2}$$

**Complexity:** *Encoding and decoding of k-bit information words and n-bit codewords can be done in time* $O(nk \cdot 2^{2k})$.

The encoder multiplies the generating matrix by the information word. Each row of the generating matrix can be computed in time $O(k \cdot 2^{2k})$. Hence, the generating matrix of $C_{n,k}$ can be computed in time $O(nk \cdot 2^{2k})$. Both the encoder and decoder require the generating matrix. Once the generating matrix of $C_{n,k}$ has been computed, the running times of the encoding and the decoding are as follows.

- The encoding of $\mathsf{Enc}(\mathbf{m},[n:1])$ of $\mathbf{m} \in \{0,1\}^k$ can be computed in time $O(n \cdot k)$.
- Maximum likelihood decoding $\mathbf{y} \in \{0,1\}^n$ can be done in time $O(n \cdot k \cdot 2^k)$.

In the following sections we describe the construction of the generating matrix of the code and analyze the error of the maximum likelihood decoder.

[6]Namely, the running time required to compute the rows of the generating matrix and decoding is exponential in $k$.

## 4.1 Computing the generating matrix

Our goal is to construct an infinite generating matrix $G$ with $k$ columns. Let $R_i \in \{0,1\}^k$ denote the $i$th row of the generating matrix. Let $G_n$ denote the $k \times n$ matrix, the rows of which are $(R_i)_{i=1\dots n}$. Let $C_{n,k}$ denote the code generated by $G_n$. The generating matrix $G$ begins with the $k \times k$ identity matrix, and hence each code $C_{n,k}$ is systematic. Subsequent rows $R_i$ (for $i > k$) of the generating matrix are constructed one by one. Let

$$W_{i,n} \triangleq \{x \in \{0,1\}^k : \mathsf{wt}(G_n \cdot x) = i\}$$

denote the $i$th weight class of $C_{n,k}$. The rows are chosen so that the weight distribution $(|W_{1,n}|, \dots, |W_{n,n}|)$ of $C_{n,k}$ is close to that of a random $[n,k]$-linear code $C^*_{n,k}$. Note that when a row vector $R_{n+1}$ is added, if $x \in \{0,1\}^k$ is orthogonal to $R_{n+1}$, then $\mathsf{wt}(G_{n+1} \cdot x) = \mathsf{wt}(G_n \cdot x)$; otherwise, $\mathsf{wt}(G_{n+1} \cdot x) = \mathsf{wt}(G_n \cdot x) + 1$. Thus $R_{n+1}$ splits each weight class $W_{i,n}$ to two parts: the orthogonal vectors which "remain" in $W_{i,n+1}$, and the non-orthogonal vectors which are "elevated" to $W_{i+1,n+1}$.

**Definition 4.2.** A vector $R \in \mathrm{GF}(2)^k$ $\varepsilon$*-splits* a set $S \subseteq \mathrm{GF}(2)^k$ if

$$\left(\frac{1}{2} - \varepsilon\right) \cdot |S| \le |\{s \in S \mid s \cdot R = 1\}| \le \left(\frac{1}{2} + \varepsilon\right) \cdot |S|\,.$$

A vector $R \in \mathrm{GF}(2)^k$ $\varepsilon$*-elevates* a set $S \subseteq \mathrm{GF}(2)^k$ if

$$|\{s \in S \mid s \cdot R = 1\}| \ge \varepsilon \cdot |S|\,.$$

The algorithm for computing the rows $R_i$ of $G$ for $i > k$ is listed as Algorithm 1 (following the high level intuition provided in Section 1.2). Roughly speaking, the algorithm $\varepsilon$-splits all "large" weight classes (which contain more than $2n^2$ information words), and employs a marking strategy to deal with "small" weight classes $W_{i,n}$ (which contain less than $2n^2$ information words). Initially, all the nonzero information words are unmarked. Once an information word becomes a member of a small weight class, it is marked, and remains marked forever (even if it later belongs to a large weight class). The unmarked vectors in $W_{i,n}$ are denoted by $\widehat{W}_{i,n}$. By definition, the set $\widehat{W}_{i,n}$ is either empty or large, and so, by a simple probabilistic argument, there exists a vector $R_{n+1}$ which simultaneously $\varepsilon$-splits all the sets $\widehat{W}_{i,n}$, for $1 \le i \le n$. In addition, $R_{n+1}$ is required to elevate the set of nonzero codewords of minimum weight. As we will later see, the distance of the resulting code grows sufficiently fast as a function of $n$, and its weight distribution is sufficiently close to the expected weight distribution of a random linear code.

We remark that (according to the analysis) the $1/8$-elevation of $W_{d,n}$ can be skipped if $\widehat{W}_{d,n} \neq \emptyset$ (namely, the elevation is required only if every vector in $W_{d,n}$ is marked). It is not hard to verify that Algorithm 1 can compute the first $n$ rows in time $O(nk \cdot 2^{2k})$.

## 4.2 Success in finding a new row

In this section we prove that Algorithm 1 succeeds in finding a new row for the generating matrix.

**Lemma 4.3.** *Algorithm 1 always finds a suitable vector $R_{n+1}$ in Line 3d.*

We begin with the following claim.

**Algorithm 1** Compute-Generating-Matrix - An algorithm for computing rows $R_n$ of the generating matrix of the rateless code for $n > k$.

1. Let $(R_1, \ldots, R_k)$ be the rows of the $k \times k$ identity matrix.
2. Initialize the set of marked information words $M \leftarrow \emptyset$.
3. For $n = k$ to $\infty$ do
   (a) For $1 \leq i \leq n$, let $W_{i,n}$ be the set of information words that are encoded by a codeword of weight $i$.
   (b) Let $d > 0$ be the minimal positive integer for which $W_{d,n}$ is non-empty.
   (c) For every $i$, if $|W_{i,n} \setminus M| < 2n^2$, then mark all the information words in $W_{i,n}$ by setting $M \leftarrow M \cup W_{i,n}$. Let $\widehat{W}_{i,n} \triangleq (W_{i,n} \setminus M)$ denote the unmarked vectors in $W_{i,n}$.
   (d) Let $R_{n+1}$ be the lexicographically first vector in $\mathrm{GF}(2)^k$ that simultaneously $1/(2\sqrt{n}$-splits every unmarked weight class $\widehat{W}_{i,n}$ and $1/8$-elevates $W_{d,n}$.

**Claim 4.4.** *For every set $W \subseteq \{0,1\}^k \setminus \{0^k\}$ of size at least $2n^2$, there are more than $2^k \cdot (1 - 1/(2n))$ vectors that $1/(2 \cdot \sqrt{n})$-split $W$.*

*Proof.* Let $W = \{x_1, \ldots, x_m\}$, where $m \geq 2n^2$. Let $R$ denote a random vector chosen uniformly in $\{0,1\}^k$. This uniform distribution induces $0-1$ random variables $Z_1, \ldots, Z_m$ defined by $Z_i = R \cdot x_i$. The expectation of each random variable $Z_i$ is $1/2$, and the variance of each $Z_i$ is $1/4$. (However, they are not independent.) Since the elements of $W$ are distinct, the random variables $\{Z_i\}_i$ are pairwise independent. By Chebyshev's Inequality,

$$\Pr\left(\left|\frac{1}{m} \cdot \sum_{i=1}^{m} Z_i - \frac{1}{2}\right| > \frac{1}{2 \cdot \sqrt{n}}\right) < \frac{1}{2n}\,. \tag{4.3}$$

Finally, note that $R$ is an $1/(2 \cdot \sqrt{n})$-splitter for $W$ if and only if

$$\left|\frac{1}{m} \cdot \sum_{i=1}^{m} Z_i - \frac{1}{2}\right| \leq \frac{1}{2 \cdot \sqrt{n}}\,.$$ □

*Proof of Lemma 4.3.* If $|W_{i,n} \setminus M| < 2n^2$, then all the information words in $W_{i,n}$ are marked, and $\widehat{W_{i,n}}$ is empty. Therefore, $\widehat{W}_{i,n}$ is either empty or of size at least $2n^2$. It follows, by a union bound, that more than half of the $k$-bit vectors simultaneously $1/(2 \cdot \sqrt{n})$-split each set $\widehat{W}_{i,n}$, for $1 \leq i \leq n$. Therefore, to prove the lemma it suffices to show that at least half of the vectors $R$ $(1/8)$-elevates the set $W_{d,n}$.

Note that any $3/8$-splitter of $W_{d,n}$ is also a $1/8$-elevator of this set. In case $|W_{d,n}| \leq 8$, pick a vector $x \in W_{d,n}$. Half the vectors are not orthogonal to $x$, and hence at least half the vectors are $1/8$-elevators of $W_{d,n}$. If $|W_{d,n}| \geq 9$, we can apply the argument of the above Claim 4.4 and get that at least half of the vectors $R$ $1/8$-elevate $W_{d,n}$. This completes the proof of Lemma 4.3. □

## 4.3 Weight distribution

In this section we analyze the weight distribution of the linear code $C_{n,k}$. We let $w_{i,n}$ be the size of $W_{i,n}$, the set of information words whose encoding under $C_{n,k}$ has Hamming weight $i$. We will show that $w_{i,n}$ is not far from the expected weight distribution $w_i^*(n,k) \triangleq \binom{n}{i} \cdot (2^k - 1) \cdot 2^{-n}$ of a random $[n,k]$ linear code.

**Claim 4.5.** *After n iterations, the number of marked information words is less than* $2n^4$.

*Proof.* For every $i, n' \leq n$ the set $W_{i,n'}$ contributes less than $2n^2$ information words to the set $M$ of marked words. Hence there are most $2n^4$ marked vectors after the $R_n$ is chosen. □

**Claim 4.6.**

$$|\widehat{W}_{i,k+t}| \leq \left(\frac{2}{3}\right)^t \binom{k+t}{i}. \tag{4.4}$$

*Proof.* By induction on $t$. The case of $t = 0$ holds because $|W_{i,k}| = \binom{k}{i}$. To prove the induction step for $t+1$, recall that the choice of $R_{k+t+1}$ splits each $\widehat{W}_{i,k+t}$ so that

$$|\widehat{W}_{i,k+t+1}| \leq \frac{2}{3} \cdot \left(|\widehat{W}_{i-1,k+t}| + |\widehat{W}_{i,k+t}|\right).$$

The induction hypothesis for $t$ now implies that

$$\begin{aligned} |\widehat{W}_{i,k+t+1}| &\leq \left(\frac{2}{3}\right)^{t+1} \cdot \left(\binom{k+t}{i-1} + \binom{k+t}{i}\right) \\ &= \left(\frac{2}{3}\right)^{t+1} \cdot \binom{k+t+1}{i}, \end{aligned}$$

and the claim follows.

□

**Claim 4.7.** *For every n and i, we have that* $w_{i,n} \leq 2n^4 + w_i^*(n,k) \cdot \Pi_{k,n}$ *where*

$$\Pi_{k,n} \triangleq \prod_{j=k+1}^{n-1} \left(1 + \frac{1}{\sqrt{j}}\right) \leq e^{2(\sqrt{n}-\sqrt{k})}.$$

*Proof.* By Claim 4.5, it suffices to bound the unmarked vectors by

$$|\widehat{W}_{i,n}| \leq w_i^*(n,k) \cdot \Pi_{k,n}. \tag{4.5}$$

Indeed, $|\widehat{W}_{i,n}|$ and $w_i^*(n,k)$ satisfy the following recurrences:

$$\begin{aligned} w_i^*(n,k) &= \frac{1}{2} \cdot \left(w_{i-1}^*(n-1,k) + w_i^*(n-1,k)\right), \\ |\widehat{W}_{i,n}| &\leq \left(1 + \frac{1}{\sqrt{n-1}}\right) \cdot \frac{1}{2} \cdot \left(|\widehat{W}_{i-1,n-1}| + |\widehat{W}_{i,n-1}|\right). \end{aligned}$$

We can now prove equation (4.5) by induction on $n \geq k$. Indeed, $w_{i,k} = w^*_{i,k}$, and

$$|\widehat{W}_{i,n}| \leq \left(1 + \frac{1}{\sqrt{n-1}}\right) \cdot \frac{1}{2} \cdot \left(w^*_{i-1}(n-1,k) \cdot \Pi_{k,n-1} + w^*_i(n-1,k) \cdot \Pi_{k,n-1}\right)$$
$$= \frac{1}{2}\left(w^*_{i-1}(n-1,k) + w^*_i(n-1,k)\right) \cdot \Pi_{k,n}$$
$$= w^*_i(n,k) \cdot \Pi_{k,n}\,.$$

The claim follows. □

We will also need to prove that the distance of $C_{n,k}$ is sufficiently large.

**Claim 4.8.** *For every $n > k$, the minimum distance of the code $C_{n,k}$ is greater than $(n-k)/(55 \cdot \log n)$.*

*Proof.* It is easier to view the evolution of the weight distribution of $C_{n,k}$ as a process of shifting balls in $n$ bins. A ball represents a nonzero information word, and a bin corresponds to a weight class. We assume that $bin(1)$ is positioned on the left, and $bin(n)$ is positioned on the right. Moving (or shifting) a ball one bin to the right means that the augmentation of the generating matrix by a new row increases the weight of the encoding of the information word by one. Note that, as the generating matrix is augmented by a new row, a ball either stays in the same bin or is shifted by one bin to the right.

Step $t$ of the process corresponds to the weight distribution of $C_{n',k}$ for $n' = t + k$. Let $bin_t(i)$ denote the set of balls in $bin(i)$ after step $t$. By Algorithm 1, the process treats marked balls and unmarked balls differently.

Let

$$\alpha \triangleq \frac{2}{\log_2(8/7)} < 11 \qquad \text{and} \qquad \Delta \triangleq \frac{n-k}{\alpha \log(2n^4)}\,.$$

Using these terms, we prove a slightly stronger minimum distance, namely,

$$bin_{n-k}(i) = \emptyset\,, \qquad \forall i \leq \Delta\,. \tag{4.6}$$

The proof is divided into two parts. First we consider the unmarked balls, and then we consider the marked balls. We begin by proving that

$$bin_{(n-k)/2}(i) \setminus M = \emptyset\,, \qquad \forall i \leq \Delta\,. \tag{4.7}$$

Namely, after $(n-k)/2$ iterations of Algorithm 1, the bins $bin_{(n-k)/2}(1), \ldots, bin_{(n-k)/2}(\Delta)$ may contain only marked balls. Note that as balls never move left, these bins remain without unmarked balls in all subsequent steps.

The proof of equation (4.7) is based on Claim 4.6. For $t = (n-k)/2$ and $i \leq \Delta$, the RHS of equation (4.4) is smaller than 1, and so equation (4.7) follows.

To prove that $bin_{(n-k)}(i) \cap M = \emptyset$ for every $i \leq \Delta$, let $t(i) \triangleq (n-k)/2 + i \cdot \log_{8/7}(2n^4)$. Note that $t(\Delta) = n-k$. We wish to prove, by induction on $i$, that the leftmost bin with a marked ball after $t(i)$ iterations is $bin(i+1)$. After $\log_{8/7}(2n^4)$ additional iterations, also $bin(i+1)$ lacks marked balls. In this manner, after $(n-k)$ iterations all the marked balls are pushed to the right of $bin(\Delta)$. Formally, we claim that

$$bin_{t(i)}(j) \cap M = \emptyset\,, \qquad \forall j \leq i\,. \tag{4.8}$$

equation (4.8) suffices because $t(\Delta) = n-k$, and hence it implies that $bin_{(n-k)}(j) = \emptyset$ for every $j \leq \Delta$, as required. The proof of equation (4.8) is by induction on $i$. For $i = 0$ the claim is trivial (because the code is systematic every nonzero information word is encoded to a nonzero word). The induction step for $i > 0$ is as follows. For every $t(i-1) < t \leq t(i)$, if $bin_t(i)$ contains a marked ball, then, by the induction hypothesis, it is the leftmost bin that contains a marked ball. Hence, each new row $R_{t+1}$ of the generator matrix $1/8$-elevates $bin_t(i)$. Since $bin_t(i)$ consists only of marked balls, by Claim 4.5, it follows that $|bin_{t(i-1)}(i)| < 2n^4$. Hence, after $\log_{8/7}(2n^4)$ steps, the bin is emptied, namely, $bin_{t(i)}(i) = \emptyset$, as required.

We proved that $bin_{(n-k)}(i)$ is empty if $i \leq \Delta$, and the claim follows. □

Overall Claim 4.8 and Claim 4.7 imply that $C_{n,k}$ is close to an "average" code in the following sense. Let $\alpha \triangleq 2/\log_2(8/7) < 11$.

**Lemma 4.9.** *The weight distribution of the constructed code $C_{n,k}$ satisfies the following bound:*

$$w_{i,n} \leq \begin{cases} 0 & \text{if } 0 < i \leq \frac{n-k}{\alpha \log(2n^4)}\,, \\ 2n^4 + w_i^*(n,k) \cdot \Pi_{k,n} & \text{if } i > \frac{n-k}{\alpha \log(2n^4)}\,. \end{cases} \tag{4.9}$$

## 4.4 Analysis of the ML decoding error

In this section we complete the proof of Theorem 4.1. Let $\mathsf{ML}$ be the maximum-likelihood decoder which, given a noisy codeword $\mathbf{y} \in \{0,1\}^n$ and $k$, finds a closest (in $L_1$ distance) codeword $\hat{\mathbf{y}} \in C_{n,k}$ and outputs the message $\mathbf{m} \in \{0,1\}^k$ for which $G_n \cdot \mathbf{m} = \hat{y}$.

*Proof of Theorem 4.1.* Fix $p$ and $\delta$, and consider $n$ and $k$ such that $n \geq k/(C(\mathsf{BSC}(p)) - \delta)$. Let $\delta_{\mathrm{GV}}$ be the root $x \in (0, 1/2)$ of the equation $H(x) = 1 - k/n$. Since the code is linear, we may assume without loss of generality that the all zero codeword was transmitted. Our goal is to upper-bound the event that $\hat{\mathbf{y}}$, the codeword computed by the ML-decoder, is non-zero. We divide the analysis into two cases based on the Hamming weight of $\hat{\mathbf{y}}$.

**Case 1: $\hat{\mathbf{y}}$ is of positive weight smaller than $\delta_{\mathrm{GV}} \cdot n$.** For a fixed codeword $\mathbf{z}$ of weight $i > 0$, erroneous decoding to $\mathbf{z}$ (instead of to the all zero codeword) corresponds to the event that the $\mathsf{BSC}(p)$ flipped at least $i/2$ bits in the support of $\mathbf{z}$. (The support of $\mathbf{z}$ is the set $\{j : z_j = 1\}$.) This event happens with probability

$$P_i \triangleq \textstyle\sum_{j=\lceil i/2 \rceil}^{i} \binom{i}{j} \cdot p^j \cdot (1-p)^{i-j}\,.$$

By a union-bound, we can upper-bound the probability of the event that $0 < \mathsf{wt}(\hat{\mathbf{y}}) < \delta_{\mathrm{GV}} \cdot n$ by

$$\sum_{i=1}^{\delta_{GV} \cdot n-1} w_{i,n} \cdot P_i \leq \sum_{i=(n-k)/(55\log n)}^{\delta_{GV} \cdot n-1} (2n^4 + e^{2\sqrt{n}}) \cdot P_i\,. \tag{4.10}$$

Indeed, Lemma 4.9 implies that (i) $w_{i,n} = 0$ if $i \leq (n-k)/(55\log n)$ and (ii) if $i/n < \delta - \text{GV}$, then $w_{i,n} \leq (2n^4 + e^{2\sqrt{n}})$ (since $w_i^*(n,k) < 1$ if $i/n < \delta_{\text{GV}}$). Below, we show that

$$P_i \leq 2^{-\beta \cdot i} \tag{4.11}$$

where $\beta \triangleq -(1/2)\cdot\log_2(4p(1-p))$ is positive since $p \in (0,1/2)$. It follows that the error probability (equation (4.10)) is upper-bounded by

$$(2n^4 + e^{2\sqrt{n}}) \cdot \sum_{i=(n-k)/(55\log n)}^{\delta_{GV}\cdot n} 2^{-\beta\cdot i} \leq e^{-\Omega(n/\log n)}.$$

It is left to prove equation (4.11). Indeed, by definition, $P_i$ satisfies

$$P_i \triangleq \textstyle\sum_{j=\lceil i/2\rceil}^{i} \binom{i}{j} \cdot p^j \cdot (1-p)^{i-j} \leq p^{i/2}\cdot(1-p)^{i/2}\cdot \sum_{j=i/2}^{i}\binom{i}{j} \leq p^{i/2}\cdot(1-p)^{i/2}\cdot 2^i,$$

which can be written as $(4p(1-p))^{i/2}$. Because $p < 1/2$, it follows that $\beta > 0$, and $P_i \leq 2^{-\beta\cdot i}$, as required.

**Case 2: $\hat{\mathbf{y}}$ is of weight greater than $\delta_{\text{GV}}\cdot n$.** In this regime, the spectrum of our code is sufficiently close to that of a random linear code, and so the error of the ML-decoding can be analyzed via (an extension of) Poltyrev's bound [21] (see also [29]). The extension bounds the probability of the event that ML-decoding returns a "heavy" word. Note that no assumption is made on the minimum distance of the code. The proof is based on an analysis in [3] and is deferred to Section 7.

**Theorem 4.10** (extension of Thm. 1 of [21])**.** *Let $p \in (0,1/2)$ be a constant, $\delta > 0$ be a constant such that $k/n < C(\mathsf{BSC}(p)) - \delta$, and $\tau \in [0,1]$ be a threshold parameter. There exists a constant $\alpha > 0$ for which the following holds. If $C$ is an $[n,k]$ linear code whose weight distribution $\{w_i(C_n)\}_i$ satisfies*

$$w_i \leq 2^{(\delta/3)n}\cdot w_i^*(n,k) \qquad \text{for every } i \geq \tau n.$$

*Then, the probability over $\mathsf{BSC}(p)$ that the all zero word with noise is ML-decoded to a codeword of weight at least $\tau n$ is at most $2^{-\alpha n}$.*

Observe the weight distribution of our code satisfies the Poltyrev's criteria for codewords of weight at least $\tau = \delta_{\text{GV}}\cdot n$. Indeed, in this regime $w_i^*(n,k) \geq 1$ and so the upper-bound $w_{i,n} \leq 2n^4 + w_i^*(n,k)\cdot\Pi_{k,n}$ from Lemma 4.9 simplifies to $w_{i,n} \leq w_i^*(n,k)2^{o(n)}$. We therefore conclude that the decoding error in case (2) is $2^{-\Omega(n)}$.

By combining the two cases, we conclude that the error probability is at most $2^{-\Omega(n/\log n)}$, as required. □

# 5 Extension to MBIOS channels

In this section we prove that the systematic linear rateless code presented in Section 4 achieves capacity for MBIOS channels. The only modification that is required is that the decoder is an ML-decoder with respect to the channel (hence the encoder is universal but the decoder depends on the channel).

Let $\mathsf{ML}_W$ denote an ML-decoder with respect to the channel $W$. For a channel $W$, let $(\mathsf{Enc}, \mathsf{ML}_W)$ denote the systematic linear rateless code where $\mathsf{Enc}$ is the encoder presented in Section 4.

**Theorem 5.1.** *For every $W \in \mathcal{MBIOS}$ and every $\delta \in (0, C(W))$,*

$$P_{\mathsf{err}}\left(W, \mathsf{Enc}, \mathsf{ML}_W, k, \frac{k}{C(W)-\delta}\right) = 2^{-\Omega(k/\log k)} \qquad (\text{as } k \to \infty), \tag{5.1}$$

$$P_{\mathsf{err}}(W, \mathsf{Enc}, \mathsf{ML}_W, k, n) = 2^{-\Omega(n/\log(n))} \qquad (\text{fixed } k,\ n \to \infty). \tag{5.2}$$

*Proof.* The proof is a consequence of [25, Appendix] that states that

$$\begin{aligned} P_{\mathsf{err}}(W, \mathsf{Enc}, \mathsf{ML}_W, k, n) \le n \cdot P_{\mathsf{err}}(\mathsf{BSC}(p), \mathsf{Enc}, \mathsf{ML}_{\mathsf{BSC}(p)}, k, n) \\ + H(P_{\mathsf{err}}(\mathsf{BSC}(p), \mathsf{Enc}, \mathsf{ML}_{\mathsf{BSC}(p)}, k, n)) \end{aligned}$$

if $C(W) \le C(\mathsf{BSC}(p))$ (where $H(x)$ denotes the binary entropy function).[7] Note that $H(x) \le 4\sqrt{x}$ for $x \in [0,1]$, hence the theorem follows from Theorem 4.1. □

# 6 Efficient rateless code for MBIOS channels

In this section we prove Theorem 3.1 by constructing a computationally efficient capacity-achieving rateless code with a universal encoder for MBIOS channels. The construction is based on a concatenated code to obtain computationally efficient encoding and decoding as well as amplification of the probability of successful decoding. The inner code is the computationally inefficient rateless code described in Section 4. The outer code is taken from [12, Lemma 1]. The rate of the outer code tends to one as the block length increases. The outer code is resilient to an adversarial channel that corrupts at most $\Theta(r)$ symbols, where $r$ is the redundancy. Encoding and decoding of the outer code requires almost linear time.

We define the rateless code $(\mathsf{Enc}', \{\mathsf{Dec}'_W\}_W)$ via its restriction to information words of length $k$ and codewords of length $n$. The code is the concatenation of an $[n_{\mathsf{out}}, k_{\mathsf{out}}]$ outer code $C_{\mathsf{out}}$ and an $[n_{\mathsf{in}}, k_{\mathsf{in}}]$ inner code $C_{\mathsf{in}}$ defined as follows.

**Inner Code.** The inner code $C_{\mathsf{in}}$ is the computationally inefficient rateless code described in Section 4 restricted to input length $k_{\mathsf{in}}$ and output length $n_{\mathsf{in}}$. Recall that this is an $[n_{\mathsf{in}}, k_{\mathsf{in}}]$ linear systematic code over $\{0,1\}$ which can be encoded in time $O(n_{\mathsf{in}} k_{\mathsf{in}} \cdot 2^{2k_{\mathsf{in}}})$ and in parallel time $O(k_{\mathsf{in}})$. The error function of ML-decoding is bounded by $2^{-\Omega(n_{\mathsf{in}}/\log n_{\mathsf{in}})}$ as long as $k_{\mathsf{in}}/n_{\mathsf{in}} \le (C(W)-\delta)$.

**Outer Code.** The outer code $C_{\mathsf{out}}$ is taken from [12, Lemma 1]. It is an $[n_{\mathsf{out}}, k_{\mathsf{out}}]$ linear systematic code over an alphabet $\Sigma_{\mathsf{out}}$ with $n_{\mathsf{out}} = k_{\mathsf{out}} \cdot (1 + |\Sigma_{\mathsf{out}}|^{-1/2})$. Hence, the rate of the outer code tends to one as the alphabet $\Sigma_{\mathsf{out}}$ increases. The outer code can be encoded in time $O(n_{\mathsf{out}} \cdot |\Sigma_{\mathsf{out}}|^{1/2})$. Decoding in time $O(n_{\mathsf{out}} \cdot |\Sigma_{\mathsf{out}}|)$ is successful as long as the fraction of errors is bounded by $\varepsilon_{\mathsf{out}} = \Theta(|\Sigma_{\mathsf{out}}|^{-1})$. Furthermore, the code can be encoded and decoded in parallel time of $O(\log(n_{\mathsf{out}} \cdot |\Sigma_{\mathsf{out}}|))$.

We emphasize that the length $k$ of the information word together with a parameter $\beta$ determine the outer code as well as the length $k_{\mathsf{in}}$ of the information word in the inner code. For fixed $k$ and $\beta$, the outer code is fixed, and the rate of the inner code decreases as $n_{\mathsf{in}}$ increases.

[7] The proof in [25] is with respect to the average error probability. However, the code is linear and the channel is symmetric, hence the average and maximum error probabilities are equal.

**Construction 6.1** (The concatenated code $C^{\beta}_{n,k} = (\mathsf{Enc}', \{\mathsf{Dec}'_W\}_W)$)**.** For lengths $k$ and $n$, and a parameter $\beta$ let

$$|\Sigma_{\mathsf{out}}| = k_{\mathsf{in}} = \beta, \quad k_{\mathsf{out}} = k/\log_2|\Sigma_{\mathsf{out}}|, \quad L_{\mathsf{in}} = (n_{\mathsf{out}} \cdot \log_2|\Sigma_{\mathsf{out}}|)/k_{\mathsf{in}}, \quad n_{\mathsf{in}} = n/L_{\mathsf{in}}.$$

- The *encoder* $\mathsf{Enc}'$ of the concatenated code $C^{\beta}_{n,k}$ maps $k$-bit information word to $n$-bit codeword as follows (see Figure 1).

$$F_2^k \overset{1}{\hookrightarrow} \Sigma_{\mathsf{out}}^{k_{\mathsf{out}}} \xrightarrow{2} \Sigma_{\mathsf{out}}^{n_{\mathsf{out}}} \overset{3}{\hookrightarrow} (F_2^{k_{\mathsf{in}}})^{L_{\mathsf{in}}} \xrightarrow{4} (F_2^{n_{\mathsf{in}}})^{L_{\mathsf{in}}}.$$

  The four steps of the encoder are: (1) A message $\mathbf{m} \in \{0,1\}^k$ is parsed as the message $\mathbf{m}_{\mathsf{out}} \in (\Sigma_{\mathsf{out}})^{k_{\mathsf{out}}}$. Namely, $\Sigma_{\mathsf{out}} = \{0,1\}^{\log\beta}$, and the message $\mathbf{m}$ is broken into $k_{\mathsf{out}}$ blocks of length $\log_2|\Sigma_{\mathsf{out}}|$. (2) The encoder of the outer code maps $\mathbf{m}_{\mathsf{out}}$ to a codeword $\mathbf{c}_{\mathsf{out}} \in (\Sigma_{\mathsf{out}})^{n_{\mathsf{out}}}$. (3) The outer codeword $\mathbf{c}_{\mathsf{out}}$ is parsed as $L_{\mathsf{in}}$ messages $(\mathbf{m}^1_{\mathsf{in}}, \dots, \mathbf{m}^{L_{\mathsf{in}}}_{\mathsf{in}})$ each over $\{0,1\}^{k_{\mathsf{in}}}$. (4) The encoder of the inner code maps each message $\mathbf{m}^j_{\mathsf{in}}$ to an inner codeword $\mathbf{c}^j_{\mathsf{in}} \in \{0,1\}^{n_{\mathsf{in}}}$.

- The *decoder* $\mathsf{Dec}'_W$ for channel $W$ of the concatenated code $C^{\beta}_{n,k}$ maps a noisy $n$-bit codeword to a $k$-bit information word as follows (see Figure 2).

$$(F_2^{n_{\mathsf{in}}})^{L_{\mathsf{in}}} \xrightarrow{4} (F_2^{k_{\mathsf{in}}})^{L_{\mathsf{in}}} \overset{3}{\hookrightarrow} \Sigma_{\mathsf{out}}^{n_{\mathsf{out}}} \xrightarrow{2} \Sigma_{\mathsf{out}}^{k_{\mathsf{out}}} \overset{1}{\hookrightarrow} F_2^k.$$

  The four steps of the decoder correspond to the encoding steps in reversed order: (4) The decoder of the inner code is an ML-decoder and is applied in parallel to each noisy inner codeword $\hat{\mathbf{c}}^j_{\mathsf{in}} \in \{0,1\}^{n_{\mathsf{in}}}$. We denote the ML-decoding of $\hat{\mathbf{c}}^j_{\mathsf{in}} \in \{0,1\}^{n_{\mathsf{in}}}$ by $\hat{\mathbf{m}}^j_{\mathsf{in}}$. (3) The $L_{\mathsf{in}}$ (inner) information words $(\hat{\mathbf{m}}^1_{\mathsf{in}}, \dots, \hat{\mathbf{m}}^{L_{\mathsf{in}}}_{\mathsf{in}})$ each over $\{0,1\}^{k_{\mathsf{in}}}$ are parsed as a noisy codeword $\hat{\mathbf{c}}_{\mathsf{out}} \in (\Sigma_{\mathsf{out}})^{n_{\mathsf{out}}}$ of the outer code. (2) The decoder of the outer code maps the noisy codeword $\hat{\mathbf{c}}_{\mathsf{out}} \in (\Sigma_{\mathsf{out}})^{n_{\mathsf{out}}}$ to a message $\hat{\mathbf{m}}_{\mathsf{out}} \in (\Sigma_{\mathsf{out}})^{k_{\mathsf{out}}}$. (1) The message $\hat{\mathbf{m}}_{\mathsf{out}}$ is parsed as a message $\hat{\mathbf{m}} \in \{0,1\}^k$.

The encoder of the rateless code (when $n$ is not predetermined) outputs the encoding of $\mathbf{m}^1_{\mathsf{in}}, \dots, \mathbf{m}^{L_{\mathsf{in}}}_{\mathsf{in}}$ "row by row." Namely, after the $i$'th bit of the encodings is output, the encoder outputs bit $i+1$ of each inner-codeword. Hence, the code $C^{\beta}_{n,k}$ is a prefix of the code $C^{\beta}_{n',k}$ for $n < n'$ and so the code defines a rateless code. Also note that the code is systematic and the complexity of encoding is

$$O(n_{\mathsf{out}} \cdot |\Sigma_{\mathsf{out}}|^{1/2} + L_{\mathsf{in}} \cdot n_{\mathsf{in}} \cdot k_{\mathsf{in}} \cdot 2^{2k_{\mathsf{in}}}) = O(n \cdot \beta \cdot 2^{2\beta})$$

and the complexity of decoding is

$$O(n_{\mathsf{out}} \cdot |\Sigma_{\mathsf{out}}| + L_{\mathsf{in}} \cdot n_{\mathsf{in}} \cdot k_{\mathsf{in}} \cdot 2^{2k_{\mathsf{in}}} + L_{\mathsf{in}} \cdot T^{ML}_W(n_{\mathsf{in}})) = O(n \cdot \beta \cdot 2^{2\beta} + L_{\mathsf{in}} \cdot T^{ML}_W(n_{\mathsf{in}}))\,.$$

(We assume pessimistically that the encoder and the decoder need to compute the generating matrix.) Furthermore, encoding can be performed in parallel-time of $O(k_{\mathsf{in}} + \log(n_{\mathsf{out}} \cdot |\Sigma_{\mathsf{out}}|)) = O(\beta + \log n)$ and decoding in parallel time of $O(\beta + \log n + T^{ML}_W(n_{\mathsf{in}}))R$.

In the following proof of Theorem 3.1, the error function is analyzed with respect to two scenarios: (1) An arbitrarily slow growing function $\beta = \beta(k)$ and rate approaching the capacity. (2) Fixed $k$ and $\beta$ (and hence the outer code is fixed), but decreasing rate of the inner code.

*Proof of Theorem 3.1.* Let $\hat{\mathbf{c}}_{\mathsf{in}} = (\hat{\mathbf{c}}^1_{\mathsf{in}}, \ldots, \hat{\mathbf{c}}^{L_{\mathsf{in}}}_{\mathsf{in}})$ denote the noisy prefix of length $n = n_{\mathsf{in}} \cdot L_{\mathsf{in}}$ of the encoding of the message $\mathbf{m}$. Let $\hat{e}$ denote the fraction of the inner-code information words that are incorrectly decoded by the ML-decoder. The decoder of the outer-code is successful as long as $\hat{e} < \varepsilon_{\mathsf{out}}$. (Note that each decoded inner information word is parsed into $k_{\mathsf{in}}/\log_2|\Sigma_{\mathsf{out}}|$ symbols of the outer code. Hence, the fraction of erroneous outer code symbols is also bounded by $\hat{e}$.) We bound the probability of the event that $\hat{e} \geq \varepsilon_{\mathsf{out}}$ using an additive Chernoff bound. Let $\varepsilon_{\mathsf{in}}$ denote the probability of erroneous decoding of a noisy inner codeword $\hat{\mathbf{c}}^j_{\mathsf{in}}$. As the ML-decoding errors are $L_{\mathsf{in}}$ independent random events, we conclude that

$$\Pr[\hat{e} \geq \varepsilon_{\mathsf{out}}] \leq 2^{-2L_{\mathsf{in}}(\varepsilon_{\mathsf{out}}-\varepsilon_{\mathsf{in}})^2}\,.$$

By Theorem 5.1, $\varepsilon_{\mathsf{in}} = e^{-\Omega(k_{\mathsf{in}}/\log k_{\mathsf{in}})}$ if the rate of the inner code is less than $C(W)$. Indeed, the rate of the outer code is

$$1/(1+\sqrt{\beta}) \stackrel{\beta\to\infty}{\longrightarrow} 1\,.$$

Hence, if $\beta$ is sufficiently large, then the rate of the inner code is less than $C(W) - \delta/2$, as required.

As $k_{\mathsf{in}} = \beta$, it follows that $\varepsilon_{\mathsf{in}} = o(1/\beta)$, and $\varepsilon_{\mathsf{out}} - \varepsilon_{\mathsf{in}} = \Theta(1/\beta)$. By definition,

$$L_{\mathsf{in}} = (n_{\mathsf{out}} \cdot \log\beta)/\beta > (k_{\mathsf{out}} \cdot \log\beta)/\beta = k/\beta\,,$$

and so the the bound on the error probability simplifies to $2^{-\Omega(k/\beta^3)}$.

In the second setting, when everything but $n_{\mathsf{in}}$ is fixed (i. e., $k, \beta$, and hence the outer code and $k_{\mathsf{in}}$ are fixed), we bound the probability of the event that $\hat{e} \geq \varepsilon_{\mathsf{out}}$ by a union bound over all $\varepsilon_{\mathsf{out}}$-fractions of $L_{\mathsf{in}}$. Namely,

$$\Pr(\hat{e} \geq \varepsilon_{\mathsf{out}}) \leq \binom{L_{\mathsf{in}}}{\varepsilon_{\mathsf{out}} \cdot L_{\mathsf{in}}} \cdot {\varepsilon_{\mathsf{in}}}^{\varepsilon_{\mathsf{out}} \cdot L_{\mathsf{in}}}\,.$$

Recall that $\varepsilon_{\mathsf{out}}$ and $L_{\mathsf{in}}$ are fixed. By Theorem 5.1, $\varepsilon_{\mathsf{in}} = e^{-\Omega(n_{\mathsf{in}}/\log n_{\mathsf{in}})}$. Because $n_{\mathsf{in}}$ is linear in $n$, the probability of the event is bounded by $2^{-\Omega(n/\log n)}$, as required. □

The proof of Corollary 3.2 is similar, except that now, when we are given the gap to capacity $\delta$ ahead of time, we can set $\beta$ to be a sufficiently large constant.

## 7 An extension of Poltyrev's theorem (Theorem 4.10)

In this section we prove Theorem 4.10 restated here for the convenience of the reader. (Recall that $\delta_{\mathrm{GV}}(n,k)$ be the root $x \in (0,1/2)$ of the equation $H(x) = 1 - k/n$ and that $w^*_i(n,k) \triangleq \binom{n}{i} \cdot 2^{k-n}$ denote the expected weight distribution of a random linear $[n,k]$ code.)

**Theorem 7.1** (Theorem 4.10 restated)**.** *Let $p \in (0,1/2)$ be a constant, $\delta > 0$ be a constant such that $k/n < 1 - H(p) - \delta$, and $\tau \in [0,1]$ be a threshold parameter. There exists a constant $\alpha > 0$ for which the following holds. If $C$ is an $[n,k]$ linear code whose weight distribution $\{w_i(C_n)\}_i$ satisfies*

$$w_i \leq 2^{(\delta/3)n} \cdot w^*_i(n,k) \qquad \textit{for every } i \geq \tau n.$$

*Then, the probability over $\mathsf{BSC}(p)$ that the all zero word is ML-decoded to a codeword of weight at least $\tau n$ is $2^{-\alpha n}$.*

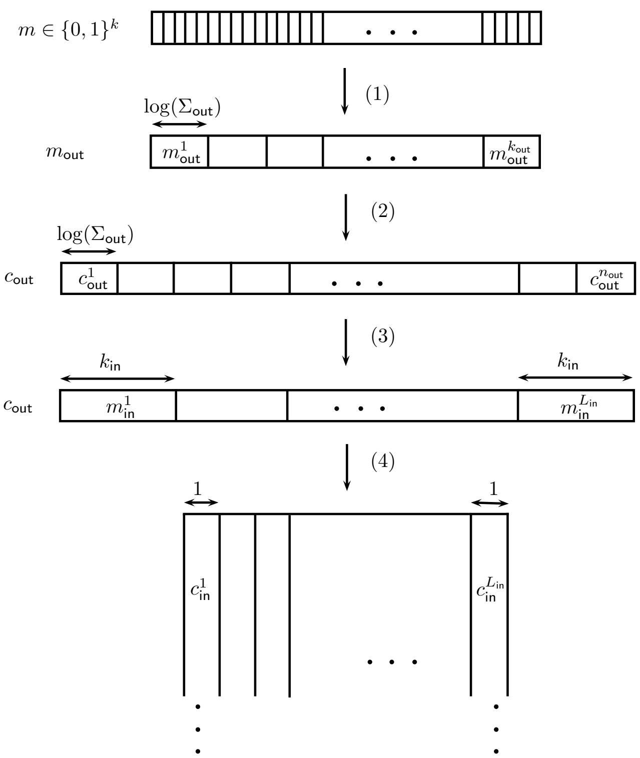


Figure 1: Encoder: concatenation of the outer code and the inner code. Recall that $|\Sigma_{\mathsf{out}}| = k_{\mathsf{in}} = \beta$, $k_{\mathsf{out}} = k/\log_2|\Sigma_{\mathsf{out}}|$, $L_{\mathsf{in}} = (n_{\mathsf{out}} \cdot \log_2|\Sigma_{\mathsf{out}}|)/k_{\mathsf{in}}$ and $n_{\mathsf{in}} = n/L_{\mathsf{in}}$.

Our proof will follow an analysis given by [3] (for a related statement). Let us first collect some useful facts.

## 7.1 A technical lemma

The extension of the binomial coefficients to reals is defined by

$$\binom{n}{k} = \frac{\Gamma(n+1)}{\Gamma(k+1)\Gamma(n-k+1)}\,, \tag{7.1}$$

where $\Gamma(x)$ is the Gamma function that extends the factorial function to the real numbers. (In particular, $\Gamma(x)$ is monotone increasing for $x \geq 1$ and $\Gamma(x+1) = x \cdot \Gamma(x)$.)

**Lemma 7.2.** *Let $0 < a \leq b$. Define the function $f : [0,b] \to \mathbb{R}$ by*

$$f(x) \triangleq \binom{a}{x}\binom{b}{x}.$$

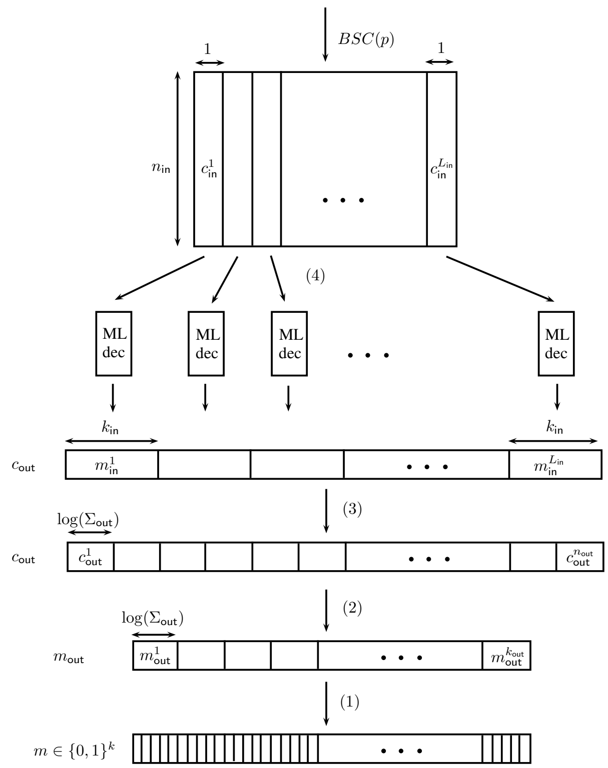


Figure 2: Decoder of the concatenated code uses ML-decoding for the inner code and the decoder of the outer code.

*Then,*

$$\left| \arg\max f(x) - \frac{ab-1}{a+b+2} \right| \le 1\,, \tag{7.2}$$

$$a^2 \ge a+b \Longrightarrow \max\{f(x)\} \le f\left(\frac{ab}{a+b}\right) \cdot \frac{(ab)^4}{(a+b)^4}\,. \tag{7.3}$$

*Proof.* Let $t \triangleq (ab-1)/(a+b+2)$. We first prove the following "discrete" monotonicity property.

1. If $i \le t$, then $f(i) \le f(i+1)$.

2. If $i \ge t$, then $f(i) \ge f(i+1)$.

The proof of this monotonicity property is by evaluating the quotient

$$Q \triangleq \frac{f(i)}{f(i+1)} = \frac{\binom{a}{i}\binom{b}{i}}{\binom{a}{i+1}\binom{b}{i+1}}\,.$$

It is easy to check that $Q \leq 1$ if $i \leq t$, and $Q \geq 1$ if $i \geq t$.

Let $x^* \triangleq \arg\max f(x)$. The monotonicity property implies that $t \leq x^* \leq t+1$, which proves the first part of the lemma.

Let $y \triangleq ab/(a+b)$. Note that $y$ is also between $t$ and $t+1$. This implies that $|x^* - y| \leq 1$. Note also that $y \geq 1$, $(a-y) = a^2/(a+b) \geq 1$, and $(b-y) = b^2/(a+b) \geq 1$. Hence by the properties of the Gamma function we obtain:

$$\begin{aligned}
\frac{f(x^*)}{f(y)} &= \frac{\Gamma^2(y+1)\cdot\Gamma(a+1-y)\cdot\Gamma(b+1-y)}{\Gamma^2(x^*+1)\cdot\Gamma(a+1-x^*)\cdot\Gamma(b+1-x^*)} \\
&\leq \frac{\Gamma^2(y+1)\cdot\Gamma(a+1-y)\cdot\Gamma(b+1-y)}{\Gamma^2(y)\cdot\Gamma(a-y)\cdot\Gamma(b-y)} \\
&= y^2\cdot(a-y)\cdot(b-y) \\
&= \left(\frac{ab}{a+b}\right)^2\cdot\frac{a^2}{a+b}\cdot\frac{b^2}{a+b} \\
&= \frac{(ab)^4}{(a+b)^4}\,.
\end{aligned}$$

□

### 7.2 Proof of Theorem 4.10

The following proof is based on [3]. Let $y$ be the received word when the all zero word is transmitted (i.e, $\{y_i\}_i$ are independent Bernoulli variables with probability $p$). Let $\hat{y}$ denote the codeword computed by the ML-decoder with respect to the input $y$. Our goal is to upper-bound the event that $\hat{y}$ has weight at least $\ell \triangleq \tau n$.

Let $\varepsilon > 0$ denote a sufficiently small constant that depends only on $p$ and $\delta$; in particular $\varepsilon$ satisfies:

$$\varepsilon \leq \min\left\{\frac{1}{2}-p, p\right\}. \tag{7.4}$$

We divide the analysis into two cases based on the Hamming weight of $y$:

$$\begin{aligned}
\Pr_y(\mathsf{wt}(\hat{y}) \geq \ell) \leq &\Pr_y[|\mathsf{wt}(y)-np| > \varepsilon n] \\
&+\Pr_y[\mathsf{wt}(\hat{y}) \geq \ell \;\&\; |\mathsf{wt}(y)-np| \leq \varepsilon n]\,.
\end{aligned}$$

**Case 1: The weight of $y$ is far from $np$, i.e $|\mathsf{wt}(y)-np| > \varepsilon n$.**

By additive Chernoff-Hoeffding inequality we know that,

$$\Pr[|\mathsf{wt}(y)-np| > \varepsilon n] \leq 2\cdot e^{-2\varepsilon^2\cdot n} = 2^{-\Omega(n)}\,.$$

**Case 2: The weight of $y$ is close to $np$, i.e $|\mathsf{wt}(y)-np| \leq \varepsilon n$.**

Let $r \triangleq \mathsf{wt}(y)$. Note that,

$$pn - \varepsilon n \leq r \leq pn + \varepsilon n\,. \tag{7.5}$$

Let $P_{\ell,r}$ denote the following probability

$$P_{\ell,r} \triangleq \Pr_y[\mathsf{wt}(\hat{y}) \geq \ell \;\&\; \mathsf{wt}(y) = r]\,.$$

Because all words $y$ of weight $r$ are equiprobable, we have

$$\Pr_y[\mathsf{wt}(\hat{y}) \geq \ell \mid \mathsf{wt}(y) = r] = \frac{|\{y : \mathsf{wt}(y) = r, \mathsf{wt}(\hat{y}) \geq \ell\}|}{|\{y : \mathsf{wt}(y) = r\}|}\,.$$

Hence,

$$\begin{aligned} P_{\ell,r} &= \Pr_y[\mathsf{wt}(y) = r] \cdot \frac{|\{y : \mathsf{wt}(y) = r, \mathsf{wt}(\hat{y}) \geq \ell\}|}{|\{y : \mathsf{wt}(y) = r\}|} \\ &\leq \sum_{i=\ell}^{n} \sum_{c \in C : \mathsf{wt}(c) = i} \frac{|\{y : \mathsf{wt}(y) = r, \hat{y} = c\}|}{|\{y : \mathsf{wt}(y) = r\}|}\,. \end{aligned} \tag{7.6}$$

Let

$$\alpha_{c,r} \triangleq |\{y : \hat{y} = c \;\&\; \mathsf{wt}(y) = r\}|\,.$$

Fix a codeword $c \in C$ of weight $i$. A word $y$ of weight $r$ is ML-decoded to $c$ only if $\mathsf{dist}(y,c) \leq r$. Without loss of generality $c = 1^i \circ 0^{n-i}$ (i. e., $c$ consists of $i$ ones followed $n-i$ zeros). Note that $\mathsf{wt}(y) = \mathsf{dist}(y, 0^n)$. Let $y'$ and $y''$ denote the prefix of length $i$ of $y$ and the suffix of length $n-i$ of $y$, respectively. Because $\mathsf{dist}(y,c) \leq r$, it follows that $0 \leq r - \mathsf{dist}(y,c) = \mathsf{dist}(y,0^n) - \mathsf{dist}(y,c)$. But

$$\begin{aligned} \mathsf{dist}(y,0^n) - \mathsf{dist}(y,c) &= \mathsf{dist}(y',0^i) + \mathsf{dist}(y'',0^{n-i}) - \mathsf{dist}(y',1^i) + \mathsf{dist}(y'',0^{n-i}) \\ &= \mathsf{dist}(y',0^i) - \mathsf{dist}(y',1^i)\,. \end{aligned}$$

Namely, in the prefix $y'$, the majority of the bits are ones. We conclude that at least $i/2$ of the coordinates of the support $y$ have to be chosen from the coordinates of the support of $c$. Hence,

$$\alpha_{c,r} = \sum_{w=i/2}^{r} \binom{i}{w} \binom{n-i}{r-w}\,. \tag{7.7}$$

Because, $\binom{i}{w} \leq \binom{i}{i/2}$, we can upper-bound equation (7.7) by,

$$\alpha_{c,r} \leq \binom{i}{i/2} \sum_{w=0}^{r-i/2} \binom{n-i}{w}\,.$$

Because $\varepsilon \leq 1/2 - p$ the maximal summand is $\binom{n-i}{r-i/2}$, and we get an upper-bound of

$$\alpha_{c,r} \leq n \binom{i}{i/2} \binom{n-i}{r-i/2}\,. \tag{7.8}$$

Substituting equation (7.8) in equation (7.6), we get,

$$P_{\ell,r} \leq \sum_{i=\ell}^{n} w_{i,n} \frac{n\binom{i}{i/2}\binom{n-i}{r-i/2}}{\binom{n}{r}}\,.$$

The weight distribution $w_{i,n}$ satisfies $w_{i,n} = 2^{(\delta/3)n} \cdot w_i^*(n,k)$, therefore,

$$P_{\ell,r} \leq n\binom{n}{r}^{-1} \sum_{i=\ell}^{n} 2^{(\delta/3)n} \cdot w_i^*(n,k)\binom{i}{i/2}\binom{n-i}{r-i/2}\,.$$

Recall that the average weight distribution $w_i^*(n,k)$ satisfies

$$w_i^*(n,k) = 2^{k-n}\binom{n}{i}$$

therefore,

$$P_{\ell,r} \leq n\binom{n}{r}^{-1} \sum_{i=\ell}^{n} 2^{k-n+(\delta/3)n}\binom{n}{i}\binom{i}{i/2}\binom{n-i}{r-i/2}\,. \tag{7.9}$$

Now, we show that,

$$\binom{n}{i}\binom{i}{i/2}\binom{n-i}{r-i/2} = \binom{n}{r}\binom{r}{i/2}\binom{n-r}{i/2}\,. \tag{7.10}$$

The combinatorial proof proceeds by counting the number of possibilities of dividing students to two classes and choosing committee members in two ways. Consider $n$ students that we wish to partition to two classes one of size $i$ and the other of size $n-i$. We want to choose a committee of $r$ students that consists of $i/2$ students from the first class, and $r-i/2$ students from the second class. The left hand side in equation (7.10) counts the number of possible partitions into two classes and choices of committee members as follows. First partition the students by choosing the members of the first class, then choose the committee members from each class. The right hand side in equation (7.10) counts the same number of possibilities by first choosing the committee members (before dividing the students into classes). Only then we partition the committee members to two classes. Finally, the non-committee members of the first class are chosen.

Plugging in equation (7.10) and equation (7.9), we get,

$$P_{\ell,r} \leq n\sum_{i=\ell}^{n} 2^{k-n+(\delta/3)n}\binom{r}{i/2}\binom{n-r}{i/2}\,.$$

By Lemma 7.2,

$$\begin{aligned}
\binom{r}{i/2}\binom{n-r}{i/2} &\leq \binom{r}{r(n-r)/n}\binom{n-r}{r(n-r)/n} \cdot \left(\frac{r(n-r)}{n}\right)^4 \\
&\leq \binom{r}{r(n-r)/n}\binom{n-r}{r(n-r)/n} \cdot n^4\,.
\end{aligned}$$

It follows that

$$P_{\ell,r} \leq n^6 \cdot 2^{k-n+(\delta/3)n} \cdot \binom{r}{r(n-r)/n}\binom{n-r}{r(n-r)/n}. \tag{7.11}$$

Let $\hat{p} \triangleq r/n$. By equation (7.5) it follows that

$$P_{\ell,r} \leq n^6 \cdot 2^{k-n+(\delta/3)n} \cdot \binom{\hat{p}n}{\hat{p}(1-\hat{p})n}\binom{(1-\hat{p})n}{\hat{p}(1-\hat{p})n}. \tag{7.12}$$

Because

$$\binom{n}{k} \leq 2^{nH(\frac{k}{n})},$$

it follows that

$$P_{\ell,r} \leq n^6 \cdot 2^{k-n+(\delta/3)n} \cdot 2^{\hat{p}nH(1-\hat{p})} \cdot 2^{(1-\hat{p})nH(\hat{p})}.$$

Because $H(\hat{p}) = H(1-\hat{p})$, we get,

$$P_{\ell,r} \leq n^6 \cdot 2^{k-n+(\delta/3)n+nH(\hat{p})}.$$

Our goal now is to prove that the exponent $k-n+(\delta/3)n+nH(\hat{p})$ is at most $-\delta \cdot n/3$. Indeed,

$$\begin{aligned} k-n+(\delta/3)n+nH(\hat{p}) &= -n \cdot \Big(-\frac{k}{n}+1-\frac{\delta}{3}-H(\hat{p})\Big) \\ &\leq -n \cdot \Big(H(p)-H(\hat{p})+\frac{2}{3}\cdot\delta\Big). \end{aligned}$$

To complete the proof, it suffices to show that $|H(\hat{p})-H(p)| < \delta/3$. Indeed, $|p-\hat{p}| \leq \varepsilon$, and hence by continuity, this holds if $\varepsilon$ is sufficiently small (as a function of $p$ and $\delta$). □

**Acknowledgments.** We thank Uri Erez, Meir Feder, Nissim Halabi, Simon Litsyn, Ronny Roth, and Rami Zamir for useful conversations. We thank Rüdiger Urbanke for pointing out reference [25, Appendix]. We thank the anonymous referees for their helpful comments and, specifically, for pointing out Remark 1.2.

# References

[1] BENNY APPLEBAUM, LIRON DAVID, AND GUY EVEN: Deterministic rateless codes for BSC. In *Proc. 6th Innovations in Theoretical Computer Science Conf. (ITCS'15)*, pp. 31–40. ACM Press, 2015. [doi:10.1145/2688073.2688117, arXiv:1406.0157] 1

[2] HARI BALAKRISHNAN, PETER IANNUCCI, JONATHAN PERRY, AND DEVAVRAT SHAH: Derandomizing Shannon: The design and analysis of a capacity-achieving rateless code, 2012. [arXiv:1206.0418] 3

[3] ALEXANDER BARG: Lecture notes ENEE 739C: Advanced topics in signal processing: Coding theory (lecture 4), 2003. Available from author's website. 16, 20, 22

[4] ALEXANDER BARG AND GEORGE DAVID FORNEY, JR.: Random codes: Minimum distances and error exponents. *IEEE Trans. Inform. Theory*, 48(9):2568–2573, 2002. [doi:10.1109/TIT.2002.800480] 4

[5] ALEXANDER BARG AND GILLES ZÉMOR: Error exponents of expander codes. *IEEE Trans. Inform. Theory*, 48(6):1725–1729, 2002. Preliminary version in ISIT'01. [doi:10.1109/TIT.2002.1003853] 2

[6] ALEXANDER BARG AND GILLES ZÉMOR: Error exponents of expander codes under linear-complexity decoding. *SIAM J. Comput.*, 17(3):426–445, 2004. [doi:10.1137/S0895480102403799] 2

[7] JOHN W. BYERS, MICHAEL LUBY, MICHAEL MITZENMACHER, AND ASHUTOSH REGE: A digital fountain approach to reliable distribution of bulk data. *SIGCOMM Comput. Commun. Rev.*, 28(4):56–67, 1998. [doi:10.1145/285243.285258] 2

[8] GIUSEPPE CAIRE AND DANIELA TUNINETTI: The throughput of hybrid-ARQ protocols for the Gaussian collision channel. *IEEE Trans. Inform. Theory*, 47(5):1971–1988, 2001. [doi:10.1109/18.930931] 2

[9] DAVID CHASE: Code combining–a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets. *IEEE Trans. Commun.*, 33(5):385–393, 1985. [doi:10.1109/TCOM.1985.1096314] 2

[10] URI EREZ, MITCHELL D. TROTT, AND GREGORY W. WORNELL: Rateless coding for Gaussian channels. *IEEE Trans. Inform. Theory*, 58(2):530–547, 2012. [doi:10.1109/TIT.2011.2173242, arXiv:0708.2575] 2

[11] GEORGE DAVID FORNEY, JR.: *Concatenated Codes*. MIT Press, 1966. 5

[12] VENKATESAN GURUSWAMI AND PIOTR INDYK: Linear-time encodable/decodable codes with near-optimal rate. *IEEE Trans. Inform. Theory*, 51(10):3393–3400, 2005. Preliminary version in STOC'02. [doi:10.1109/TIT.2005.855587] 5, 17

[13] JEONGSEOK HA, JAEHONG KIM, AND STEVEN W. MCLAUGHLIN: Rate-compatible puncturing of low-density parity-check codes. *IEEE Trans. Inform. Theory*, 50(11):2824–2836, 2004. [doi:10.1109/TIT.2004.836667] 2

[14] JOACHIM HAGENAUER: Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Trans. Commun.*, 36(4):389–400, 1988. [doi:10.1109/26.2763] 2

[15] TINGFANG JI AND WAYNE STARK: Rate-adaptive transmission over correlated fading channels. *IEEE Trans. Commun.*, 53(10):1663–1670, 2005. [doi:10.1109/TCOMM.2005.857147] 2

[16] SHU LIN, DANIEL COSTELLO, AND MICHAEL MILLER: Automatic-repeat-request error-control schemes. *IEEE Communications Magazine*, 22(12):5–17, 1984. [doi:10.1109/MCOM.1984.1091865] 2

[17] MICHAEL LUBY: LT codes. In *Proc. 43rd FOCS*, pp. 271–280. IEEE Comp. Soc. Press, 2002. [doi:10.1109/SFCS.2002.1181950] 2

[18] DAVID MANDELBAUM: An adaptive-feedback coding scheme using incremental redundancy. *IEEE Trans. Inform. Theory*, 20(3):388–389, 1974. [doi:10.1109/TIT.1974.1055215] 2

[19] JONATHAN PERRY, HARI BALAKRISHNAN, AND DEVAVRAT SHAH: Rateless spinal codes. In *Proc. 10th ACM Workshop on Hot Topics in Networks (HotNets-X'11)*, pp. 6:1–6:6. ACM Press, 2011. [doi:10.1145/2070562.2070568] 3

[20] JONATHAN PERRY, PETER A. IANNUCCI, KERMIN E. FLEMING, HARI BALAKRISHNAN, AND DEVAVRAT SHAH: Spinal codes. In *Proc. 2012 Conf. on Applications, Technologies, Architectures, and Protocols for Comput. Commun. (SIGCOMM'12)*, pp. 49–60. ACM Press, 2012. [doi:10.1145/2342356.2342363] 2, 3

[21] GREGORY POLTYREV: Bounds on the decoding error probability of binary linear codes via their spectra. *IEEE Trans. Inform. Theory*, 40(4):1284–1292, 1994. [doi:10.1109/18.335935] 4, 16

[22] DORON RAJWAN: Method of encoding and transmitting data over a communication medium through division and segmentation, 2007. US Patent 7,304,990. 2

[23] DORON RAJWAN, EYAL LUBETZKY, AND JOSEPH YOSSI AZAR: Data streaming, 2008. US Patent 7,327,761. 2

[24] DOUGLAS N. ROWITCH AND LAURENCE B. MILSTEIN: On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes. *IEEE Trans. Commun.*, 48(6):948–959, 2000. [doi:10.1109/26.848555] 2

[25] EREN ŞAŞOĞLU: *Polar Coding Theorems for Discrete Systems*. Ph. D. thesis, EPFL, 2011. 17, 25

[26] STEFANIA SESIA, GIUSEPPE CAIRE, AND GUILLAUME VIVIER: Incremental redundancy hybrid ARQ schemes based on low-density parity-check codes. *IEEE Trans. Commun.*, 52(8):1311–1321, 2004. [doi:10.1109/TCOMM.2004.833022] 2

[27] AMIN SHOKROLLAHI: Raptor codes. *IEEE Trans. Inform. Theory*, 52(6):2551–2567, 2006. [doi:10.1109/TIT.2006.874390] 2

[28] NADAV SHULMAN: *Communication over an Unknown Channel via Common Broadcasting*. Ph. D. thesis, Tel Aviv University, 2003. LINK at SemanticScholar. 2

[29] NADAV SHULMAN AND MEIR FEDER: Random coding techniques for nonrandom codes. *IEEE Trans. Inform. Theory*, 45(6):2101–2104, 1999. [doi:10.1109/18.782147] 4, 16

[30] NADAV SHULMAN AND MEIR FEDER: Static broadcasting. In *Proc. 2000 IEEE Internat. Symp. on Inform. Theory (ISIT'00)*, p. 23. IEEE Comp. Soc. Press, 2000. [doi:10.1109/ISIT.2000.866313] 2

[31] DANIEL A. SPIELMAN: *Computationally efficient error-correcting codes and holographic proofs*. Ph. D. thesis, MIT, 1996. Available at DSpace@MIT. 5

[32] DANIEL A. SPIELMAN: Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Inform. Theory*, 42(6):1723–1731, 1996. Preliminary version in STOC'95. [doi:10.1109/18.556668] 5

## AUTHORS

Benny Applebaum
Associate professor
Tel Aviv University
Tel Aviv, Israel
bennyap@post.tau.ac.il
`https://www.eng.tau.ac.il/~bennyap`

Liron David
Ph. D. candidate
Tel Aviv University
Tel Aviv, Israel
lirondavid@gmail.com

Guy Even
Professor
Tel Aviv University
Tel Aviv, Israel
guy@eng.tau.ac.il
`http://hyde.eng.tau.ac.il/wordpress/`

## ABOUT THE AUTHORS

BENNY APPLEBAUM is an Associate Professor of Electrical Engineering at Tel-Aviv University. He received his B. Sc. from the Hebrew University of Jerusalem in 2002, and his Ph. D. from the Computer Science Department of the Technion in 2007 under the supervision of Yuval Ishai and Eyal Kushilevitz. Before joining Tel-Aviv he was a postdoc at Princeton University and Weizmann Institute of Science. He is interested in the theory of computation, mainly in cryptography and computational complexity. He enjoys spending time with his family and playing the guitar.

LIRON DAVID is a Ph. D. candidate in the Electrical Engineering Department at Tel-Aviv University, under the supervision of Avishai Wool. She received a Weinstein paper prize in 2018 and a Weinstein award for excellence in studies in 2017. She got a Rector's award for excellence in teaching for the academic year 2015-2016. She completed her M. Sc. in 2014 under the supervision of Benny Applebaum and Guy Even in the Electrical Engineering Department at Tel-Aviv University. She owes much of her interest in research to her M. Sc. advisors. She obtained her B. Sc. in Electrical and Electronics Engineering and Computer Science at Tel-Aviv University in 2011. Liron's hobby is playing the piano. She completed four years of piano studies at the Israel Conservatory of Music, Tel-Aviv.

GUY EVEN received his B. Sc. degree in 1988 in Mathematics and Computer Science from the Hebrew University. Received his M. Sc. and D. Sc. in Computer Science from the Technion in 1991 and 1994, respectively. Guy spent his post-doctorate in the University of the Saarland during 1995–1997 with Wolfgang Paul. Since 1997, he has been a faculty member in the School of Electrical Engineering in Tel-Aviv University. He is interested in algorithms and their applications in various fields. He has published papers on the theory of VLSI, approximation algorithms, computer arithmetic, online algorithms, frequency assignment, scheduling, packet-routing, linear-programming decoding of LDPC codes, and rateless codes. He is on the editorial board of "Theory of Computing Systems." Together with Moti Medina, Guy wrote a digital hardware textbook titled "Digital Logic Design: A Rigorous Approach" published in 2012 by Cambridge University Press.