Necessity and Sufficiency of $\log(n)$ Bits of Shared Randomness for Randomized Coding Capacity in Arbitrarily Varying Channels *

Sagnik Bhattacharya
Final Year Undergraduate
Department of Electrical Engineering, IIT Kanpur
Kanpur, India

Dr Amitalok Budkuley
Postdoctoral Researcher
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong

Dr Sidharth Jaggi
Assistant Professor
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong

November 2018

Abstract

We improve the bound of $\theta(\log n)$ due to Langberg on the amount of common randomness required to communicate reliably in the presence of an omniscient state-deterministic adversary to a tight $\log n$ bound. We give a scheme based on list decoding and polynomial hashing that achieves capacity with $(1+\epsilon)\log n$ bits of common randomness and show that $(1-\epsilon)\log n$ bits of common randomness are not enough to do so.

1 Introduction

Much of information theory involves studying the limits of reliable communication over a channel. When the noise is random (the Shannon model), there is a huge body of work starting with Shannon's original paper. In this case, we have a precise characterisation of the capacity of the channel, with any rate below capacity being achievable with probability of error going to zero asymptotically, and conversely, any rate above capacity implying probability of error bounded away from zero. We know the error exponents, computationally efficient codes coming close to capacity for many channels.

The problem becomes much harder for adversarial noise (the Hamming model). Even for binary input, binary output channels the best known upper (LP bound) and lower (GV bound) bounds on the achievable rates don't match. Also, using sphere-packing bounds like the Hamming bound, we know that rate equal to the random noise capacity is not possible in the adversarial noise setting.

Arbitrarily varying channels, which were introduced by Blackwell, model channels which can change adversarially with time and also taking into account constraints on the adversary and the input. They can be viewed as being intermediate between the Shannon and Hamming models

*to be submitted to ISIT 2019

mentioned earlier. In most cases, for deterministic or stochastic encoding and decoding, there is still a gap between the best achievable rate on AVC's and the stochastic noise capacity.

With the presence of common randomness, the problem of being able to communicate at rates close to the stochastic channel capacity becomes tractable. In the same paper, Blackwell showed that with unbounded common randomness at the encoder and decoder, it is possible to achieve the stochastic noise capacity, essentially proving that the randomized coding capacity is same as the stochastic noise capacity. Later on, achievability schemes that used less common randomness were also proposed, and the best schemes showed that $O(\log n)$ bits of common randomness is sufficient. There was also work done on lower bounds of the common randomness, and Langberg showed for the binary case that $O(\log n)$ bits of common randomness is also necessary.

In this paper, we give a precise characterisation of the amount of common randomness, by giving an achievability scheme using slightly more than $\log n$ bits of common randomness and showing that $\log n$ bits are necessary.

2 Notation and Problem Setup

2.1 Notation

Let $\mathbb{R}$ and $\mathbb{R}_+$ denote the sets of real numbers and non-negative real numbers respectively. Similarly, let $\mathbb{R}^n$ and $\mathbb{R}_+^n$, respectively, denote the sets of length-n real vectors and length-n real vectors with non-negative components. Let $\mathbb{N}$ denote the set of natural numbers. For any $K \in \mathbb{N}$, let $[K]$ denote the set $\{1, 2, \cdots, K\}$. We denote by $\mathbb{F}$ a finite field $\mathbb{F}_q$, where $q = p^m$ for some prime p and integer m. Upper case letters (e.g. X) are used to denote random variables, lower case letters (e.g. x) represent the values taken by them, and calligraphic letters (e.g. $\mathcal{X}$) denote their alphabet. We use boldface notation to represent random vectors (e.g. $\mathbf{X}$), and the values taken by them (e.g. $\mathbf{x}$). Unless specified otherwise, the length of the vectors is n (e.g. $\mathbf{X} = (X_1, X_2, \cdots, X_n)$) and corresponds to the *block length* of operation. Also, let $\mathbf{X}^i = (X_1, X_2, \cdots, X_i)$ and $\mathbf{x}^i = (x_1, x_2, \cdots, x_i)$, and similarly, let $\mathbf{X}_i^j = (X_i, X_{i_1}, \cdots, X_j)$ and $\mathbf{x}_i^j = (x_i, x_{i_1}, \cdots, x_j)$. The dot product between two vectors $\mathbf{x}$, $\mathbf{x}' \in \mathcal{X}^n$ is denoted by $\langle \mathbf{x}, \mathbf{x}' \rangle$. Alternately, we also denote this dot product by $\mathbf{x}^T \cdot \mathbf{x}'$, where $\mathbf{x}$, $\mathbf{x}'$ are viewed as column vectors and $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$. Let $\mathcal{P}(\mathcal{X})$ denote the set of all probability distributions defined over the set $\mathcal{X}$. Also, let $\mathcal{P}(\mathcal{X}|\mathcal{Y})$ denote the set of conditional distribution of a random variable with alphabet $\mathcal{X}$ conditioned on another random variable taking values in alphabet $\mathcal{Y}$. Let X and Y denote two random variables taking values in $\mathcal{X}$ and $\mathcal{Y}$ respectively. Then, we denote the distribution of X by $P_X(\cdot)$, the joint distribution of X and Y by $P_{X,Y}(\cdot,\cdot)$, and the conditional distribution of X given Y by $P_{X|Y}(\cdot|\cdot)$. Marginal distribution of X under the joint distribution $P_{X,Y}(\cdot,\cdot)$ is given by $[P_{X,Y}]_X(\cdot)$. The entropy of X is denoted by $H(X)$ and given as $H(X) := \sum_{x\in\mathcal{X}} P(x) \log \frac{1}{P(x)}$. The joint entropy of X and Y is denoted by $H(X,Y)$, and given as $H(X,Y) := \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} P(x,y) \log \frac{1}{P(x,y)}$. The conditional entropy of X given Y is denoted by $H(X|Y)$, where $H(Y|X) := \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} P_{XY}(x,y) \log \frac{1}{P_{Y|X}(y|x)}$. The mutual information between X and Y is denoted by $I(X;Y)$ and defined as $I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} P_{XY}(x,y) \log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)}$.

Given two probability distributions P_X and Q_X on alphabet $\mathcal{X}$, let $\mathbb{D}(P_X||Q_X)$ denote the Kullback-Liebler distance between P_X and Q_X. Here $\mathbb{D}(P_X||Q_X) := \sum_{x\in\mathcal{X}} P_X(x) \log \frac{P_X(x)}{Q_X(x)}$. The standard conventions that $0 \log 0 = 0 \log 0/0 = 0$ and $p \log p/0 = \infty$ for $p > 0$ are assumed.

Throughout this paper, all logarithms and exponentiations are with respect to (w.r.t.) base 2 unless stated otherwise.

Let $T_X(\cdot)$ denote a type of X, $T_{X,Y}(\cdot,\cdot)$ denote a joint type of (X,Y) and $T_{X|Y}(\cdot,\cdot)$ denote a conditional type of X given Y. Under certain scenarios, we sometimes also denote these quantities alternately as vectors $T_x(X)$, matrices $T_{x,y}(X,Y)$ and vectors $T_{x|y}(X|Y)$ respectively, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The corresponding sets of all such types are denoted by $\mathcal{T}^n(\mathcal{X})$, $\mathcal{T}^n(\mathcal{X},\mathcal{Y})$ and $\mathcal{T}^n(\mathcal{X}|\mathbf{y})$ respectively. Unless specified otherwise, all types are under block length n. We denote the type of $\mathbf{x}$ by $T_{\mathbf{x}}(\cdot)$, the joint type of $(\mathbf{x},\mathbf{y})$ by $T_{\mathbf{x},\mathbf{y}}(\cdot,\cdot)$, and the conditional type of $\mathbf{x}$ conditioned on $\mathbf{y}$ by $T_{\mathbf{x}|\mathbf{y}}(\cdot|\cdot)$. Let $N(x|\mathbf{x}) = |\{i \in [n] : x_i = x\}|$ be the number of occurrences of symbol $x \in \mathcal{X}$ in vector $\mathbf{x}$. Then, we have $T_{\mathbf{x}}(x) = \frac{N(x|\mathbf{x})}{n}$, $\forall x \in \mathcal{X}$. Note that $T_{\mathbf{x},\mathbf{y}}$ and $T_{\mathbf{x}|\mathbf{y}}$ can be similarly defined. Given $\epsilon > 0$, the set of ϵ-typical set of $\mathbf{x}$ sequences for a distribution X is denoted by $\mathcal{T}_\epsilon^n(P_X) := \{\mathbf{x} \in \mathcal{X}^n : \|T_{\mathbf{x}} - P_X\|_\infty \leq \epsilon\}$. Similarly, given a joint distribution $P_{X,Y}(\cdot,\cdot)$ and some $\mathbf{x} \in \mathcal{X}^n$, the set of ϵ-typical set of conditionally typical $\mathbf{y}$ sequences, conditioned on $\mathbf{x}$, for a distribution $P_{X,Y}$ is denoted by $\mathcal{T}_\epsilon^n(P_{X,Y}|\mathbf{x}) := \{\mathbf{y} \in \mathcal{Y}^n : \|T_{\mathbf{x},\mathbf{y}} - P_{X,Y}\|_\infty \leq \epsilon\}$.

2.2 The Communication Setup

Consider the communication figure depicted in Fig. 1. Alice intends to reliably send a message

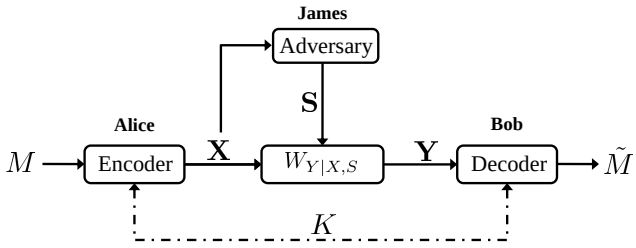

Figure 1: The communication setup

M to Bob over an arbitrarily varying channel (AVC) partially controlled by a jamming adversary James. The AVC has two inputs: sender's input $X \in \mathcal{X}$ and the jamming state $S \in \mathcal{S}$, and a single output $Y \in \mathcal{Y}$. Random variables X, S and Y take values in finite alphabets $\mathcal{X}, \mathcal{S}$ and $\mathcal{Y}$ respectively. The AVC behaviour is specified by the conditional distribution $W_{Y|X,S}$. Note that we assume a *state-deterministic* AVC $W_{Y|X,S}$, i.e., given any $x \in \mathcal{X}$ and $s \in \mathcal{X}$, $W_{Y|X,S}(y|x,s)$ equals 1 for a unique $y \in \mathcal{Y}$ and is zero otherwise.

Let us assume the standard block coding framework with a the block length n. Here X_i, S_i and Y_i denote the symbols associated with time instant i. For any $\mathbf{x} \in \mathcal{X}^n$ and $\mathbf{s} \in \mathcal{S}^n$, the probability of observing $\mathbf{y} \in \mathcal{Y}^n$ is given by

$$\mathbb{P}(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}, \mathbf{S} = \mathbf{s}) = \prod_{i=1}^{n} W_{Y|X,S}(y_i|x_i, s_i).$$

Alice encodes the message M into a codeword $\mathbf{X}$ and sends it on the channel. The encoder has a *input constraint* Λ_X, where $\Lambda_X \subseteq \mathcal{P}(\mathcal{X})$ is a convex set. In particular, this implies that $\mathbf{X} \in \mathcal{X}^n$

satisfies the input constraint Λ_X if $T_{\mathbf{X}} \in \Lambda_X^{(n)}$, where $\Lambda_X^{(n)} = \mathcal{T}^{(n)}(\mathcal{X}) \cap \Lambda_X$[1].

A noisy version $\mathbf{Y}$ of $\mathbf{X}$, corrupted by James' jamming input $\mathbf{S}$, is received at the decoder. Bob then outputs an estimate of $\hat{M}$ of the message M. We assume that Alice and Bob have access to randomness, denoted by $K \in \mathcal{K}$, unknown to the adversary. James is an *omniscient* jamming adversary, i.e., James can choose $\mathbf{S}$[2] under the knowledge of the coding scheme, the message M and the entire codeword $\mathbf{X}$. James has a *state constraint* Λ_S, where $\Lambda_S \subseteq \mathcal{P}(\mathcal{S})$ is a convex set. In particular, the chosen state $\mathbf{S}$ satisfies the state constraint if $T_{\mathbf{S}} \in \Lambda_S^{(n)}$, where $\Lambda_S^{(n)} = \mathcal{T}^{(n)}(\mathcal{S}) \cap \Lambda_S$[3]. For brevity, we introduce the following definition.

Definition 1 ($(\Lambda_X, \Lambda_S, W_{Y|X,S})$-AVC)**.** *An $(\Lambda_X, \Lambda_S, W_{Y|X,S})$-AVC denotes an AVC with a conditional distribution $W_{Y|X,S}$ under an input constraint $\Lambda_X \subseteq \mathcal{P}(\mathcal{X})$ and a state constraint $\Lambda_S \subseteq \mathcal{P}(\mathcal{S})$.*

An (n, R) *deterministic* code is a pair of fixed encoder-decoder mappings (ψ, ϕ). Here the encoder $\psi : \{1, 2, \cdots, 2^{nR}\} \to \mathcal{X}^n$, where $T_{\psi(m)} \in \mathcal{T}^n \cap \Lambda_X^{(n)}$, $\forall m$, and the decoder $\phi : \mathcal{Y}^n \to \{0, 1, 2, \cdots, 2^{nR}\}$, where the output 0 denotes a decoding error. We assume that 2^{nR} is an integer. An (n, R) randomized code is a random vector which takes values in the set of all (n, R) deterministic codes. If the randomized encoder-decoder pair almost surely takes values in a subset $\mathcal{K}$ of (n, R) deterministic codes, where $K \in \mathcal{K}$ denotes the randomly instantiated (n, R) deterministic code, then we call such a randomized code (n, R, K) randomized code. Thus, an (n, R, K) randomized code is a random variable $K \in \mathcal{K}^n$. Alternately, we also sometimes denote the randomized code by $K = (\Psi, \Phi)$, where (Ψ, Φ) denote the randomized encoder-decoder pair. The *key size* of the randomized code in given by $H(K)$. Recall that the actual realized key is shared between the encoder-decoder but is unknown to the adversary. However, the adversary may know the distribution of the (n, R, K) randomized code. For any given (n, R, K) randomized code, we define its probability of error by[4]

$$P_e^{(n)} := \max_m \max_{\mathbf{s}:\mathcal{X}^n \to \mathcal{S}^n} \mathbb{P}_{\mathbf{K}}(\Phi(\mathbf{Y}) \neq m | M = m)$$

where the probability is over the randomized code.
An (n, N, L) deterministic list code is a pair of mappings (ψ, ϕ), where encoder $\psi : \{1, 2, \cdots, N\} \to \mathcal{X}^n$ and decoder $\phi : \mathcal{Y}^n \to N^L$. The rate of this code $R = \frac{1}{n} \log(\frac{N}{L})$. For any (n, N, L) code, its

[1]Generally, an AVC is specified with an input constraint $\Lambda_X \in \mathbb{R}$ under some *cost function* $c_X : \mathcal{X} \to \mathbb{R}_+$, where $c_X(x) < \infty$, $\forall x \in \mathcal{X}$ (cf. [**?**]). This specifies the family of sets $\Lambda_X^{(n)} \in \mathbb{R}^n$, $\forall n$, where $\Lambda_X^{(n)} := \{\mathbf{x} : \frac{1}{n}\sum_{i=1}^n c(x_i) \leq \Lambda_X\}$. We can equivalently represent this as follows: $\Lambda_X^{(n)} = \{\mathbf{x} : \mathbf{c}^T T_{\mathbf{x}} \leq \Lambda_X\}$, where *cost function* $\mathbf{c} \in \mathbb{R}_+^{|\mathcal{X}|}$. Crucially, observe that any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^n$ such that $T_{\mathbf{x}'} = T_{\mathbf{x}}$ have identical costs , i.e., $\mathbf{c}^T \cdot T_{\mathbf{x}} = \mathbf{c}^T \cdot T_{\mathbf{x}'}$ under this definition. Our definition captures this fact more succinctly by defining the constraint set $\Lambda_X \subseteq \mathcal{P}(\mathcal{X})$, and hence, generalizes the notion of an *input constraint*.

[2]Even though James can privately randomize, we show later that randomized jamming strategies afford no benefit vis-à-vis the error performance. See footnote on the next page for detailed discussion.

[3]Similar to the input constraint, our definition of the *state constraint* differs from the usual definition [**?**]. For details, refer the discussion w.r.t. input constraint given earlier.

[4]There error is specified w.r.t. deterministic jamming strategies. However, this is not restrictive as randomized jamming strategies do not degrade the error performance any further. To see this, note that the error for message m under some randomized jamming strategy $P_{\mathbf{S}|\mathbf{X}=\mathbf{x}}$ (this may also depend on the distribution of the (n, R, K) randomized code) is $\sum_{\mathbf{s}:T_{\mathbf{s}} \in \Lambda_S^{(n)}} \mathbb{P}(\mathbf{S} = \mathbf{s}|\mathbf{x}) P_e^{(n)}(m, \mathbf{S} = \mathbf{s}) \leq \max_{\mathbf{s}:T_{\mathbf{s}} \in \Lambda_S} P_e^{(n)}(m, \mathbf{s})$. This implies that there exists a deterministic feasible jamming strategy with at least the same error performance as $P_{\mathbf{S}|\mathbf{X},m}$. Hence, even for the optimizing jamming strategy $P_{\mathbf{S}|\mathbf{X}}$, one can find a corresponding optimum determistic jamming strategy $\mathbf{s}(\mathbf{x})$.

error is given by

$$\mathbf{e}^{(n)} := \max_{m} \max_{\mathbf{s}:\mathcal{X}^n \to \mathcal{S}^n} \mathbb{I}_{\{\Phi(\mathbf{Y}) \neq m | M=m\}}$$

A rate R is *achievable* if for any $\epsilon > 0$, there exists an $n_0(\epsilon)$ such that for every $n \geq n_0$, there exists an (n, R) randomized code such that the corresponding probability of error $P_e^{(n)} \leq \epsilon$. The supremum of the achievable rates is defined as the *randomized capacity* of the AVC $W_{Y|X,S}$. This will be denoted by C_0. The definitions of an achievable rate and the capacity can be analogously given when (n, R, K) randomized codes are employed. We denote the capacity under (n, R, K) randomized codes by C_K. We study the capacity of the channel $W_{Y|X,S}$ under different assumptions on shared key K.

3 Main Results

Definition 2.

$$C_0 := \max_{P_X \in \Gamma_X} \min_{P_{S|X}:[P_X P_{S|X}]_S \in \Lambda_S} I(X;Y) \tag{1}$$

We now state the main results.

Theorem 1. *Given any $\epsilon > 0$, there exists an (n, R, K) randomized code with rate R given by*

$$R = C_0 - \epsilon, \tag{2}$$

key size

$$K \leq n^{1+\epsilon}, \tag{3}$$

and the maximum probability of error $P_e^{(n)} \leq c_1(\epsilon) n^{-c_2(\epsilon)}$, where $c_1(\epsilon), c_2(\epsilon) > 0$ and do not depend on n.

Theorem 2. *Given any AVC, if the zero-error capacity is not equal to the capacity then $\log(nR)$ bits of common randomness are necessary to communicate over the channel at capacity.*

4 Proof of Theorem 1

4.1 Achievability

Before we present the acheivability for randomized codes, we first show an important result on deterministic list codes for the AVC $\mathcal{A}$.

Theorem 3. *Given any $\epsilon > 0$, there exists a deterministic list code with rate R given by*

$$R = C_0 - \epsilon, \tag{4}$$

list size

$$L = \frac{2\log(|\mathcal{Y}|)}{\epsilon}, \tag{5}$$

and exponentially decaying maximum probability of error $P_e^{(n)} \leq 2^{-nc(\epsilon)}$, where $c(\epsilon) > 0$.

Proof. We use Lemma 4 to prove this result. Fix the optimizing distribution in (1), say P_X^*. We know from Lemma 4 that for every type $T_X \in \mathcal{T}^n(\mathcal{X}) \cap \Gamma_X$, there exists a list code with list size $L = \frac{2\log(|\mathcal{Y}|)}{\epsilon}$ and rate R given in (6). As the function $\min_{P_{S|X}:[P_X P_{S|X}]_S \in \Lambda_S} I(X;Y)$ is continuous in T_X and since the set of types $T_X \in \mathcal{T}^n(\mathcal{X}) \cap \Gamma_X$ is dense in the set of feasible input distributions $\Gamma_X \subseteq \mathcal{P}(\mathcal{X})$, it follows that for blocklength n sufficiently large, there exists a list code of list size L as given in (5) with rate $R = C_0 - \epsilon$. This completes the proof. □

Lemma 4. *Given any $\epsilon > 0$ and any type $T_X \in \mathcal{T}^n(\mathcal{X}) \cap \Gamma_X$, there exists for n sufficiently large, a list code comprising all codewords of type T_X, rate R such that*

$$R = \min_{P_{S|X}:[P_X P_{S|X}]_S \in \Lambda_S} I(X;Y) - \epsilon, \tag{6}$$

list size

$$L = \frac{2\log(|\mathcal{Y}|)}{\epsilon}. \tag{7}$$

and exponentially decaying maximum probability of error $P_e^{(n)} \leq 2^{-nc(\epsilon)}$, where $c(\epsilon) > 0$.

Proof. Fix arbitrary $\epsilon > 0$, and set the rate R as given in (6). We now describe the random list code construction along with the corresponding encoder and decoder.
Code construction: Generate 2^{nR} random codewords $\{\mathbf{X}_i\}_{i=1}^{2^{nR}}$, where each codeword is chosen uniformly at random from the set $\mathcal{T}^n(T_X)$ all $\mathbf{x}$ vectors with type T_X. We denote this collection as the random codebook $\mathbf{C}$. Note that $\mathbf{C}$ is also revealed to the adversary.
Encoder: Given message $m \in [2^{nR}]$, the encoder transmits codeword $\mathbf{X}_m$.
Decoder: The decoder receives channel output $\mathbf{y}$, and determines the following list of codewords

$$\mathcal{L}(\mathbf{y}) := \{i \in [2^{nR}] : T_{\mathbf{x}_i,\mathbf{y}} \in \mathcal{T}^n\left([T_X T_S W_{Y|X,S}]_{X,Y}\right) \text{ for some } T_S \in \mathcal{T}^n(\mathcal{S}) \cap \Lambda_S\}.$$

Next, if $|\mathcal{L}(\mathbf{y})| > L$ (we specify L as in (7) later; see discussion after (9)) then the decoder declares error and outputs $\hat{m} = 0$. Otherwise, the decoder outputs the list of estimated messages in $\mathcal{L}(\mathbf{y})$.
Error analysis for the list code: Let E_{err} denote error event. Recall that we are interested in the *maximum* probability of error, Hence, without loss of generality, let us fix the transmitted message

as $M = m$. Then, under any feasbile jamming state $\mathbf{s}$, we have

$$\begin{aligned}
\mathbb{P}_{\mathbf{C}}(E_{err}|M=m) &= \mathbb{P}\left(\exists \mathbf{y} \in \mathcal{Y}^n : |\mathcal{L}(\mathbf{y})| > L\right) \\
&\overset{(a)}{\leq} \sum_{\mathbf{y}\in\mathcal{Y}^n} \mathbb{P}(\exists \mathcal{B} \subseteq [2^{nR}], |\mathcal{B}| > L : \mathbf{X}_i \in \mathcal{L}(\mathbf{y}), i \in \mathcal{B}) \\
&\overset{(b)}{=} \sum_{\mathbf{y}\in\mathcal{Y}^n} \mathbb{P}(\exists \mathcal{B} \subseteq [2^{nR}], |\mathcal{B}| > L : \mathbf{X}_i \in \mathcal{L}(\mathbf{y}), i \in \mathcal{B}) \\
&\overset{(c)}{\leq} \sum_{\mathbf{y}\in\mathcal{Y}^n} \sum_{\substack{\mathcal{B}\subseteq[2^{nR}] \\ |\mathcal{B}|>L}} \mathbb{P}(\mathbf{X}_i \in \mathcal{L}(\mathbf{y}), i \in \mathcal{B}) \\
&\overset{(d)}{=} \sum_{\mathbf{y}\in\mathcal{Y}^n} \sum_{\substack{\mathcal{B}\subseteq[2^{nR}] \\ |\mathcal{B}|>L}} \left(\mathbb{P}(\mathbf{X}_i \in \mathcal{L}(\mathbf{y}))\right)^{|\mathcal{B}|} \\
&\overset{(e)}{=} \sum_{\mathbf{y}\in\mathcal{Y}^n} \sum_{i>L}^{2^{nR}} \binom{2^{nR}}{i} \left(\mathbb{P}(\mathbf{X}_i \in \mathcal{L}(\mathbf{y}))\right)^{i} \left(1 - \mathbb{P}(\mathbf{X}_i \in \mathcal{L}(\mathbf{y}))\right)^{2^{nR}-i} \\
&\leq \sum_{\mathbf{y}\in\mathcal{Y}^n} \sum_{i=L+1}^{2^{nR}} \binom{2^{nR}}{i} \left(\mathbb{P}(\mathbf{X}_i \in \mathcal{L}(\mathbf{y}))\right)^{i} \\
&\overset{(f)}{\leq} \sum_{\mathbf{y}\in\mathcal{Y}^n} \sum_{i=L+1}^{2^{nR}} 2^{nR(L+1)} \left(\mathbb{P}(\mathbf{X}_i \in \mathcal{L}(\mathbf{y}))\right)^{L+1} \\
&\overset{(g)}{\leq} \sum_{\mathbf{y}\in\mathcal{Y}^n} 2^{nR} 2^{nR(L+1)} \left(\mathbb{P}(\mathbf{X}_i \in \mathcal{L}(\mathbf{y}))\right)^{L+1}
\end{aligned}$$

We now make the following claim.

Claim 5. *For any $\mathbf{y} \in \mathcal{Y}^n$, we have*

$$\mathbb{P}_{\mathbf{C}}(\mathbf{X} \in \mathcal{L}(\mathbf{y})) \leq 2^{-n(\min_{P_S\in\Lambda_S} I(X;Y) - f(n))} \tag{8}$$

where $f(n) > 0$ and $\lim_{n\to\infty} f(n) = 0$.

Proof of Claim. Fix any $\mathbf{y} \in \mathcal{Y}^n$. Then, we have

$$\begin{aligned}
\mathbb{P}_{\mathcal{C}}(\mathbf{X} \in \mathcal{L}(\mathbf{y})) &= \mathbb{P}\left(\mathbf{X} : T_{\mathbf{X},\mathbf{y}} \in \mathcal{T}^n\left([T_X T_S W_{Y|X,S}]_{X,Y}\right) \text{ for some } T_S \in \mathcal{T}^n(\mathcal{S}) \cap \Lambda_S\right) \\
&= \mathbb{P}\left(\mathbf{X} : T_{\mathbf{X},\mathbf{y}} \in \bigcup_{\substack{T_{X,Y} = [T_X T_S W_{Y|X,S}]_{X,Y} \\ T_S \in \mathcal{T}^n(\mathcal{S}) \cap \Lambda_S}} \mathcal{T}^n\left(T_{X,Y}\right)\right) \\
&\overset{(a)}{\leq} \sum_{\substack{T_{X,Y} = [T_X T_S W_{Y|X,S}]_{X,Y} \\ T_S \in \mathcal{T}^n(\mathcal{S}) \cap \Lambda_S}} \mathbb{P}\left(\mathbf{X} : T_{\mathbf{X},\mathbf{y}} \in \mathcal{T}^n\left(T_{X,Y}\right)\right) \\
&\overset{(b)}{\leq} \sum_{\substack{T_{X,Y} = [T_X T_S W_{Y|X,S}]_{X,Y} \\ T_S \in \mathcal{T}^n(\mathcal{S}) \cap \Lambda_S}} \frac{2^{nH_{[T_X T_S W_{Y|X,S}]_{X,Y}}(X|Y)}}{(n+1)^{-|\mathcal{X}|} 2^{nH_{T_X}(X)}} \\
&\leq \sum_{\substack{T_{X,Y} = [T_X T_S W_{Y|X,S}]_{X,Y} \\ T_S \in \mathcal{T}^n(\mathcal{S}) \cap \Lambda_S}} \frac{\max_{T_S \in \mathcal{T}^n(\mathcal{S}) \cap \Lambda_S} 2^{nH_{[T_X T_S W_{Y|X,S}]_{X,Y}}(X|Y)}}{(n+1)^{-|\mathcal{X}|} 2^{nH_{T_X}(X)}} \\
&\overset{(c)}{\leq} (n+1)^{|\mathcal{S}| \cdot |\mathcal{X}|} 2^{-n \min_{T_S \in \mathcal{T}^n(\mathcal{S}) \cap \Lambda_S} I(X;Y)} \\
&\leq (n+1)^{|\mathcal{S}| \cdot |\mathcal{X}|} 2^{-n \min_{P_S \in \Lambda_S} I(X;Y)} \\
&\leq 2^{-n\left(\min_{P_S \in \Lambda_S} I(X;Y) - f(n)\right)},
\end{aligned}$$

where $f(n) := \frac{|\mathcal{S}| \cdot |\mathcal{X}| \log(n+1)}{n} > 0$ and $\lim_{n\to\infty} f(n) = 0$. Here (a) follows from the union bound, while (b) and (c) follows from elementary facts about size of sets of types (cf. [**?**, pg. 17]). As there are at most a polynomial (in n) number of types T_S, we get (c). This completes the proof of the claim. □

Using the bound in Claim 5, we simplify (8) as follows

$$\begin{aligned}
\mathbb{P}_{\mathbf{C}}(E_{err}|M=m) &\leq \sum_{\mathbf{y} \in \mathcal{Y}^n} 2^{nR(L+1)} \left(2^{-n\left(\min_{P_S \in \Lambda_S} I(X;Y) - f(n)\right)}\right)^{L+1} \\
&\overset{(a)}{\leq} |\mathcal{Y}^n| 2^{nR(L+1)} 2^{-n(L+1)\left(\min_{P_S \in \Lambda_S} I(X;Y) - f(n)\right)} \\
&\overset{(b)}{=} 2^{n \log(|\mathcal{Y}|)} 2^{nR(L+1)} 2^{-n(L+1)(R+\epsilon-f(n))} \\
&= 2^{n \log(|\mathcal{Y}|)} 2^{-n(L+1)(\epsilon - f(n))} \\
&\overset{(c)}{\leq} 2^{-n\left((L+1)\frac{\epsilon}{2} - \log(|\mathcal{Y}|)\right)}
\end{aligned}$$

where (a) follows from noting that the RHS in Claim 5 does not depend on $\mathbf{y} \in \mathcal{Y}^n$. We get (b) by noting that $R = \bar{R} - \epsilon$ and (6), and (c) by choosing n large enough such that $f(n) \leq \epsilon/2$. Setting list size L as in (7), it follows that $\mathbb{P}(\mathcal{E}|M=1) \leq 2^{-nc(\epsilon)}$, where $c(\epsilon) > 0$, and hence, independent of m. Hence, $\mathbb{E}_{\mathbf{C}}[P_e^{(n)}] = \max_m \mathbb{P}_{\mathbf{C}}(E_{err}|M=m) \to 0$ as $n \to \infty$. This guarantees that with high probability, there exists a deterministic rate R list code with list size $L = \mathcal{O}(\frac{1}{\epsilon})$ and zero error, and completes the proof. □

Proof of Achievability for Theorem 1 Having established the existence of deterministic list codes (cf. Theorem 3), we now present the proof of achievability of our main result in Theorem 1. Toward this, we first describe the randomized codebook construction and the concomitant encoding and decoding maps, following which we analyse the probability of error for this code to establish the achievability result.

Let $\epsilon > 0$. Now fix a rate $R = C_0 - \epsilon$. Let P_X be the optimizing distribution in (1).
Key generation: Given $\epsilon > 0$, let $\Omega := (1+\epsilon)\log(n)$. Let $\beta \in \mathbb{R}^+$, where $\beta := \frac{3\epsilon}{4} < \epsilon$, and $k = \frac{2}{\epsilon}$. Now we define

$$\gamma := \left\lceil \frac{1+\beta}{1+k} \log(n) \right\rceil$$

Consider a field $\mathbb{F}$ of size 2^γ. Consider an integer $n \geq n(\epsilon) := 2^{\frac{4\epsilon+8}{\epsilon^2}}$. Now generate $(1+\beta)\log(n)$ independent and identically distributted (i.i.d.) bits, each generated using the distribution $Bern(1/2)$. We denote this collection of bits by Θ, where $\Theta \leq \Omega$ (see Appendix **??** for details), and $\Theta \to \Omega$ as $\epsilon \to 0$. We now split Θ into $(k+1)$ equal-sized chunks (see Fig. **??**),

i.e., $\Theta = (R_0, R_1, \cdots, R_k)$, where each chunk $R_i \in \mathbb{F}$, $i = 0, 1, \cdots, k$. Θ is now revealed to the encoder and decoder (but not to the adversary).
Codebook construction: Let

$$R' := R + \frac{(1+\beta)\log(n)}{(k+1)n}. \tag{9}$$

This choice of R' will be clear later in the description of the encoder. Note that $R < R' < C_0$, where $R' \to R$ as $n \to \infty$. Using this and from Lemma 4, it follows that there exists a deterministic list code with rate R' and list size $L = \frac{\log(|\mathcal{Y}|)}{\epsilon}$. Let $\mathcal{C}_L$ denote this list code, and let (ψ_L, ϕ_L) denote the corresponding has encoder-decoder pair. Using $\mathcal{C}_L$, we now describe the encoding and decoding in our randomized code in detail.
Encoder: The encoder observes the message m and key Θ (shared with decoder). It proceeds as follows:

- Given the binary representation of the message $m \in [2^{nR}]$, the encoder calculates its field-$\mathbb{F}$ representation $m^{(\mathbb{F})}$. Here $m^{(\mathbb{F})} := (m_1^{(\mathbb{F})}, m_2^{(\mathbb{F})}, \cdots, m_L^{(\mathbb{F})})$, where

$$L := \left\lceil \frac{nR}{\gamma} \right\rceil$$

  and $m_i^{(\mathbb{F})} \in \mathbb{F}$, $\forall i = 1, 2, \cdots, L$.

- Next, $m^{(\mathbb{F})} \in \mathbb{F}^L$ is arranged in a k-dimensional square matrix (say, $T(m)$) of dimension

$$\begin{aligned} D &:= \left\lceil L^{\frac{1}{k}} \right\rceil \\ &= \left\lceil \left\lceil \frac{nR}{\gamma} \right\rceil^{\frac{1}{k}} \right\rceil \end{aligned}$$

  Here each matrix entry is given by $m^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)}$, where $i_l \in [D]$ for $l = 1, 2, \cdots, k$.

- The encoder now calculates a hash value for the message m under Θ given by

$$H_\Theta(m) := R_0 + \sum_{i_1=1}^{D}\sum_{i_2=1}^{D}\cdots\sum_{i_k=1}^{D} R_1^{i_1}R_2^{i_2}\cdots R_k^{i_k} m^{(\mathbb{F})}_{(i_1,i_2,\cdots,i_k)}, \tag{10}$$

  using the unique collection $\{m^{(\mathbb{F})}_{(i_1,i_2,\cdots,i_k)}\}$ corresponding to message m. Here $H_\Theta(m) \in \mathbb{F}$, $\forall m$.

- The encoder now concatenates m and its hash $H_\Theta(m)$. Let this 'concatenated message' be $\bar{M} = (m, H_\Theta(m))$. Note that $\bar{M} \in [2^{nR'}]$, where R' is given in (9).

- Finally, the encoder transmits $\mathbf{X} = \psi_L(\bar{M})$ over the channel.

Decoder: The decoder knows the shared key Θ and observes channel output $\mathbf{y}$. It proceeds as follows:

- Using the list decoder $\phi_L : \mathcal{Y}^n \to [2^{nR'}]^L$, it outputs a list $L(\mathbf{y})$ of candidate codewords. Recall from earlier that ϕ_L outputs the following list

$$\mathcal{L}(\mathbf{y}) := \{i \in [2^{nR'}] : T_{\mathbf{x}_i,\mathbf{y}} \in \mathcal{T}^n\left([T_X T_S W_{Y|X,S}]_{X,Y}\right) \text{ for some } T_S \in \mathcal{T}^n(\mathcal{S}) \cap \Lambda_S\}.$$

- Let the collection $\{\hat{\bar{m}} = (\hat{m}, \hat{h})\}$ denote this list of pairs in $\mathcal{L}(\mathbf{y})$. If there exists a *unique* $\tilde{\bar{m}} = (\tilde{m}, \tilde{h}) \in \mathcal{L}(\mathbf{y})$, such that $(\tilde{m}, \tilde{h})$ is *consistent* w.r.t. the observed shared key $\Theta = (R_0, R_1.\cdots, R_k)$, i.e.,

$$\tilde{h} = H_\Theta(\tilde{m})$$

  then the decoder outputs the corresponding message $\tilde{m} \in [2^{nR}]$ as the estimate. Otherwise, it outputs $\tilde{m} = 0$ to declare error.

Probability of error analysis: Let $E := \{\hat{M} \neq M\}$ denote the error event. Recall that we are interested in the maximum probability of error criterion. Hence, let us fix $M = m$ w.l.o.g., and analyse the following conditional probability under any feasible state $\mathbf{s} \in \Lambda_S$ of the adversary:

$$\begin{aligned}
\mathbb{P}_\Theta(E|M=m) &= \mathbb{P}\left(\Phi(\mathbf{Y}) \neq m | M = m\right) \\
&\leq \mathbb{P}\left(\Phi(\mathbf{Y}) \neq m | M = m, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right) + \mathbb{P}\left(\mathbb{I}_{\{|\mathcal{L}(\mathbf{Y})| > \mathcal{O}(1/\epsilon)\}}\right) \\
&\overset{(a)}{\leq} \mathbb{P}\left(\exists \bar{M}' \in [|\mathcal{L}(\mathbf{Y})|] : M' \neq m, H_\Theta(m') = H_\Theta(m) | M = m, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right) + 2^{-nc} \\
&\overset{(b)}{\leq} \mathbb{P}\left(\exists \bar{M}' \in [\mathcal{O}(1/\epsilon)] : M' \neq m, H_\Theta(M') = H_\Theta(m) | M = m, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right) + 2^{-nc} \\
&\overset{(c)}{\leq} \sum_{j=1}^{\mathcal{O}(1/\epsilon)} \mathbb{P}\left(\bar{M}'_j : M'_j \neq m, H_\Theta(M'_j) = H_\Theta(m) | M = m, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right) + 2^{-nc} \\
&\overset{(d)}{\leq} \mathcal{O}(1/\epsilon)\mathbb{P}\left(\bar{M}' : M' \neq m, H_\Theta(M') = H_\Theta(m) | M = m, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right) + 2^{-nc} \\
&\overset{(e)}{\leq} \mathcal{O}(1/\epsilon)\mathbb{P}\left(\bar{m}' : m' \neq m, H_\Theta(m') = H_\Theta(m) | M = m, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right) + 2^{-nc}
\end{aligned}$$

Here (a) follows due to the list-size property and maximum probability of error of the list code $\mathcal{C}_L = (\psi_L, \phi_L)$. We get (b) by noting the conditional event $\{|\mathcal{L}(\mathbf{y})| \leq \mathcal{O}(1/\epsilon)\}$. The union bound

gives us (c), while fixing the concomitant optimizing $\bar{M}'$ gives (d). The conditional distribution of $\{\bar{M}'_j\}$ in (c), and hence, of $\bar{M}'$ in (d) is fairly complicated; so we instead allow the adversary to fix *any* arbitrary $\bar{m}' \in [2^{nR'}]$ in (e). Note that such an action is a *strengthening* of the adversary, and can only degrade the probability of error. Thus, it gives us the upper bound in (e).

Lemma 6. *Conditioned on $M = m$, $|\mathcal{L}(\mathbf{y})| \leq \mathcal{O}(1/\epsilon)$, we have*

$$\mathbb{P}\left(\bar{m}' : m' \neq m, H_\Theta(m') = H_\Theta(m) | M = m, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right) \leq \frac{kD}{2^\gamma}, \tag{11}$$

under any feasbile jamming state $\mathbf{s} \in \Lambda_S$ chosen by the adversary.

Proof. We first prove the conditional version of this lemma. Toward this, let us first condition on the events $H_\Theta(m) = h$ and $\mathbf{X} = \mathbf{x}$. Recall that the adversary already knows the message $M = m$ and the transmitted codeword $\mathbf{X} = \mathbf{x}$. Thus, the conditioning on $H_\Theta(m) = h$ is additional knowledge which makes the adversary stronger and can only increase the probability in (11), thereby upper bounding the LHS in (11). Under such a stronger adversary, we now analyse the following:

$$\begin{aligned}
&\mathbb{P}\quad \left(\bar{m}' : m' \neq m, H_\Theta(m') = H_\Theta(m) = h | M = m, H_\Theta(m) = h, \mathbf{X} = \mathbf{x}, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right) \\
&\quad = \mathbb{P}\Bigg(\bar{m}' : m' \neq m, R_0 + \sum_{i_1, i_2, \cdots, i_k} R_1^{i_1} R_2^{i_2} \cdots R_k^{i_k} m'^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)} \\
&\qquad\qquad = R_0 + \sum_{i_1, i_2, \cdots, i_k} R_1^{i_1} R_2^{i_2} \cdots R_k^{i_k} m^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)} = h \Bigg| M = m, H_\Theta(m) = h, \mathbf{X} = \mathbf{x}, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\Bigg) \\
&\quad = \mathbb{P}\left(\bar{m}' : m' \neq m, \sum_{i_1, i_2, \cdots, i_k} R_1^{i_1} R_2^{i_2} \cdots R_k^{i_k} \left(m'^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)} - m^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)}\right) = 0 \middle| M = m, H_\Theta(m) = h, \mathbf{X} = \mathbf{x}, |\mathcal{L}(\mathbf{Y}\right) \\
&\quad = \mathbb{P}\left(\bar{m}' : m' \neq m, \sum_{i_1, i_2, \cdots, i_k} R_1^{i_1} R_2^{i_2} \cdots R_k^{i_k} m''^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)} = 0 \middle| M = m, H_\Theta(m) = h, \mathbf{X} = \mathbf{x}, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right)
\end{aligned}$$

where the last step follows by defining $m''^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)} := m'^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)} - m^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)}$. As $m' \neq m$, it follows that the term

$$\sum_{i_1, i_2, \cdots, i_k} R_1^{i_1} R_2^{i_2} \cdots R_k^{i_k} m''^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)}$$

is not trivially zero. To proceed, we need two results. The first is the following important claim.

Claim 7. *The conditional distribution of $(R_1, R_2, \cdots, R_k)$, conditioned on the adversary's view of the message m, the hash value h, the transmitted codeword $\mathbf{x}$ and the list encoder-decoder (ψ_L, ϕ_L), is a uniform distribution.*

Proof of Claim. Recall from the encoder description earlier that we calculate the hash value for any message, say $m \in [2^{nR}]$, as

$$H_\Theta(m) = R_0 + \sum_{i_1, i_2, \cdots, i_k} R_1^{i_1} R_2^{i_2} \cdots R_k^{i_k} m^{(\mathbb{F})}_{(i_1, i_2, \cdots, i_k)}.$$

Note that conditioned on the message $M = m$ and the hash value $H_\Theta(m) = h$, the distribution of $(R_1, R_2, \cdots, R_k)$ depends only on R_0. Conditioning on $R_0 = r_0$, it immediately follows that the conditional distribution of $(R_1, R_2, \cdots, R_k)$ is uniform over values which are compatible with $R_0 = r_0$ when message and hash value are fixed to m and h respectively in (12). As this is true for every r_0, the overal distribution of $(R_1, R_2, \cdots, R_k)$ is uniform as required. This proves the claim. □

Remark 1. *Interestingly, however, this is not the case for the conditional distribution for R_0. This is because, conditioned on the message $M = m$ and the hash value $H_\Theta(m) = h$, for every value $(R_1, R_2, \cdots, R_k) = (r_1, r_2, \cdots, r_k)$, the equation in* (12) *completely determines R_0.*

The second useful result is the well-known Schwarz-Zippel lemma [**?**]. Next, we recapitulate this result.

Lemma 8 (Schwarz-Zippel lemma [**?**])**.** *Let $P = \mathbb{F}[z_1, z_2, \cdots, z_l]$ be a non-zero polynomial of total degree $d \geq 0$ over some field $\mathbb{F}$. Let $\mathcal{S} \subseteq \mathbb{F}$ such that $|\mathcal{S}| < \infty$, and let $\mathbf{Z} = (Z_1, Z_2, \cdots, Z_l)$ be selected uniformly at random from $\mathcal{S}$. Then,*

$$\mathbb{P}\left(P(Z_1, Z_2 \cdots, Z_l) = 0\right) \leq \frac{d}{|\mathcal{S}|}.$$

Coming back to the proof of this lemma, we now use Claim 7 and Lemma 8, to directly simplify (12) to

$$\mathbb{P} \quad \left(\bar{m}' : m' \neq m, H_\Theta(m') = H_\Theta(m) = h | M = m, H_\Theta(m) = h, \mathbf{X} = \mathbf{x}, |\mathcal{L}(\mathbf{Y})| \leq \mathcal{O}(1/\epsilon)\right) \leq \frac{kD}{2^\gamma} \quad (13)$$

where we make the correspondence $\mathcal{S} \leftarrow 2^\gamma$ and $d \leftarrow kD$ in Lemma 8. This completes the proof of the conditional version of the lemma. As the RHS in (12) does not depend on the conditioned random variables, the unconditioned version follows directly, thereby proving the lemma. □

We now return to the probability of error analysis. Using Lemma 6, we can simplify (11) as

$$\begin{aligned}
\mathbb{P}_\Theta(E|M = m) &\leq \mathcal{O}(1/\epsilon)\frac{kD}{2^\gamma} + 2^{-nc} \\
&\leq 2\mathcal{O}(1/\epsilon)\frac{kD}{2^\gamma} \qquad \text{for large enough } n \\
&\leq c_1 n^{-c_2},
\end{aligned}$$

where c_1, $c_2 > 0$. The final step follows from careful though elementary simplifications.

From (13), it follows that $P_e^{(n)} = \max_m \mathbb{P}(E|M = m) \leq c_1 n^{-c_2}$, and hence, $P_e^{(n)} \to 0$ as $n \to \infty$. This shows that the rate R is achievable. As $\epsilon > 0$ was arbitrary, any rate arbitrarily close to C_0 is achievable. This completes the proof of achievability.

4.2 Converse

In the following, we prove a converse result under an *average* probability of error criterion instead of the *maximum* probability of error criterion. Given an (n, R, K) randomized code, we define the corresponding average probability of error by

$$P_e^{(n)} := \frac{1}{2^{nR}} \sum_{m=1}^{2^{nR}} \max_{\mathbf{s}:\mathcal{X}^n \to \mathcal{S}^n} \mathbb{P}_K(\phi(\mathbf{Y}) \notin m | M = m).$$

For the rest of this section, we assume $P_e^{(n)}$ to denote average probability of error.

Consider any sequence of (n, R, K) codes with corresponding $P_e^{(n)} \to 0$ as $n \to \infty$. We need to show that for this sequence of codes (a) $R < C$ and (b) $\Theta \geq \log(n)$. We now prove the first part.
Upper bound on the rate R:
The proof starts along the lines of the converse of the standard discrete memoryless channel [**?**]. From Fano's inequality, it immediately follows that for the given sequence of (n, R, L) randomized codes, we have $H(M|\mathbf{Y}, \Theta) \leq n\epsilon_n$, where $\epsilon_n \to 0$ as $n \to \infty$. We now note that the rate R can be bounded as

$$\begin{aligned}
R &= H(M) \\
&= I(M, \mathbf{Y}, \Theta) + H(M|\mathbf{Y}, \Theta) \\
&\leq I(M; \mathbf{Y}, \Theta) + n\epsilon_n \\
&\leq I(M; \Theta) + I(M; \mathbf{Y}|\Theta) + n\epsilon_n \\
&= I(M; \mathbf{Y}|\Theta) + n\epsilon_n \\
&\leq I(M, \Theta; \mathbf{Y}) + n\epsilon_n \\
&\leq I(\mathbf{X}; \mathbf{Y}) + n\epsilon_n \\
&\leq \sum_{i=1}^{n} I(X_i; Y_i) + n\epsilon_n \\
&= n\left(\sum_{i=1}^{n} \frac{1}{n} I(X_i; Y_i)\right) + n\epsilon_n \\
&= n\left(\sum_{i=1}^{n} \mathbb{P}(Q = i) I(X_i; Y_i | Q = i)\right) + n\epsilon_n \\
&= nI(X_Q; Y_Q|Q) + n\epsilon_n \\
&= nI(X_Q, Q; Y_Q) + n\epsilon_n \\
&= nI(X_Q; Y_Q) + n\epsilon_n \\
&= nI(X; Y) + n\epsilon_n,
\end{aligned}$$

under every feasible $P_{S|X}$. As $n \to \infty$, we thus, get for any $P_X \in \Gamma_X$,

$$R \leq \min_{P_{S|X}:[P_X P_{S|X}]_S \in \Lambda_S} I(X; Y).$$

As this is true for every $P_X \in \Gamma_X$, we can further bound it as

$$R \leq \max_{P_X \in \Gamma_X} \min_{P_{S|X}:[P_X P_{S|X}]_S \in \Lambda_S} I(X; Y).$$

This completes the proof of the first part.

5 Proof of Theorem 2

For the second part, we need to prove a lower bound of $\log(n)$ on the shared randomness. This proof is via a contradiction argument. Assume for the contraction, assmume that there exists an (n, R, K) code with rate $R < C_0$, $P_e^{(n)} \leq \delta$, $\delta > 0$ and uses shared randomness $K = (1-\epsilon)\log(n)$, for some $\epsilon > 0$. We first give the dependence of r on message-length nR and alphabet size $|\mathcal{X}|$, for constant δ. We use a proof by contradiction - let the amount of common randomness r be $\leq (1-\epsilon)\log nR + \log\log|\mathcal{X}|$ bits for some $\epsilon > 0$. For $|\mathcal{X}| \ll nR$, we can absorb the second term into the first term by suitably modifying the value of ϵ, so essentially this shows the necessity of $\log(nR)$ bits of common randomness.

Assume that there exists an $[n, R, r]$ private code (ψ, ϕ) of block-length n, rate R which uses r bits of common randomness that is privately decodable with probability $1 - \delta > \frac{1}{2}$ within error parameter p' greater than $p - \epsilon_p$. We construct an adversary $\mathcal{S}$ with error parameter p' so that the average decoding error of any private decoder ϕ for this adversary is $\geq \delta$. This will contradict the assumption about the decoding error of the code (ψ, ϕ), proving the claim.

5.1 Proof Overview

We find 'large' balls of codewords which have a large number of pairs of codewords that are close together. For each such pair, the adversary pushes them to the middle of the two. This results in a decoding error for a large fraction of transmitted codewords.

5.2 Some Definitions

Let $\mathcal{M}$ notation correct? be the set of all messages, $|\mathcal{M}| = |\mathcal{X}|^{nR}$. Let $\mathcal{R}$ notation again be the set of all vectors of common randomness, $|\mathcal{R}| = 2^r$. The encoder ψ maps pairs (m, r) to codewords $\mathbf{x}$. Let the image of ψ be the set

$$\mathcal{X} \coloneqq \{\psi(w, r) \mid m \in \mathcal{M}, r \in \mathcal{R}\} \tag{13}$$

Clearly, $|\mathcal{X}| \leq |\mathcal{X}|^{nR} \cdot 2^r$.

Assume for now that there are no collisions in the code, that is, there does not exist codewords $\mathbf{x} \in \mathcal{X}$ such that $\psi(m_1, r) = \psi(m_2, r) = x$ and $m_1 \neq m_2$.

For each $\mathbf{x} \in \mathcal{X}$ define the sets

$$R_{\mathbf{x}} = \{r \in \mathcal{R} \mid \exists m \in \mathcal{M}\ \psi(m, r) = \mathbf{x}\} \tag{14}$$

Therefore, all pre-images of $\mathbf{x} \in \mathcal{X}$ are pairs (m, r) such that $r \in R_{\mathbf{x}}$.

For each of $R \in \{R_{\mathbf{x}}\}$, define the subset of $\mathcal{X}$ consisting of codewords $\mathbf{x}$ such that $R_{\mathbf{x}} = R$ as the set $\mathcal{X}_R$. Clearly, the sets $\mathcal{X}_R$ form a partition of the set $\mathcal{X}$.

Let the sets $R_{\mathbf{x}}$ correspond to some set of vertices in a graph and the codewords $\mathbf{x}$ correspond to another (disjoint) set of vertices. From each R-vertex, draw an edge to an $\mathbf{x}$-vertex if and only if $R = R_{\mathbf{x}}$ (or, equivalently, $\mathbf{x} \in \mathcal{X}_R$). It is clear that this constructs a bipartite graph, with the sets R in one part and codewords $\mathbf{x}$ on the other.

We define a size threshold on the degree of the R-vertices. That is, we call the degree small if it is smaller than

$$\alpha(n) = nR - (nR)^{1-\epsilon} + log_{|\mathcal{X}|}(1-2\delta) \tag{15}$$

and large if it is $\geq \alpha(n)$. We call the R-sets 'large' or 'small' depending on their degrees. The motivation for this choice of $\alpha(k)$ will be given later.

5.3 Main Proof

5.3.1 Counting sets and images

We consider all of the small R-sets and take the union of all sets $\mathcal{X}_R$ corresponding to these R-sets, calling this union A. Note that $A \subset \mathcal{X}$. Taking an union-bound over all possible *subsets* of $\mathcal{R}$ (there are total 2^r vectors in $\mathcal{R}$, and therefore a maximum of 2^{2^r} subsets of R) and the maximum value of the out-degree of each of the small R-sets, we get that

$$|A| < 2^{2^r} \cdot |\mathcal{X}|^{\alpha(n)} \tag{16}$$

From the assumed value of r and the size-threshold $\alpha(n)$, we get that

$$|A| < |\mathcal{X}|^{nR+\log_{|\mathcal{X}|}(1-2\delta)} \tag{17}$$

In addition to the bipartite graph constructed above, it may be helpful to visualise the map ψ as $\psi : \mathcal{M} \times \mathcal{R} \to \mathcal{X}$ for the next few arguments.

By assumption, there are no collisions, so if we fix $\mathbf{x}$, then corresponding to every $r \in R_{\mathbf{x}}$ we get a unique $m \in \mathcal{M}$. In the worst case, the set $R_{\mathbf{x}}$ can be the whole set R and we get $|\mathcal{R}|$ (m, r) pairs for each $\mathbf{x} \in A$. Using this, we can bound the size of the inverse image of the set A as follows:

$$|\psi^{-1}(A)| < 2^r \cdot |\mathcal{X}|^{nR+\log_{|\mathcal{X}|}(1-2\delta)} \tag{18}$$

The remaining (m, r) pairs then have to map to A^c, which is just the union of the sets $\mathcal{X}_R$ corresponding to large R-sets. Therefore, we get

$$|\psi^{-1}(A^c)| \geq 2^r \cdot |\mathcal{X}|^{nR} - 2^r \cdot |\mathcal{X}|^{nR+\log_{|\mathcal{X}|}(1-2\delta)} \tag{19}$$

We want this to be large because the proposed adversary will work only for the codewords mapped to by the large sets, which is precisely the set A^c. This is directly related to the choice of $\alpha(n)$, and the requirement that the probability of error be at least δ gives an upper bound on the value of $\alpha(n)$.

5.3.2 Large R-sets

Fix a large R-set and consider the set $\mathcal{X}_R$. Note that for large k and $0 < \epsilon_R < 1$, using the definition of $\alpha(n)$,

$$|\mathcal{X}_R| \gg |\mathcal{X}|^{nR(1-\epsilon_R)} \tag{20}$$

Therefore, expurgation of all codewords outside $\mathcal{X}_R$ leads to just an ϵ_R loss of rate. If such a code were possible, we would have constructed an an $(\epsilon_R, 2\epsilon_p)$-perfect code for the channel. This is not the case because by assumption, the list decoding capacity and the zero-error capacity are different. Also, note that the existence of a gap between the list-decoding capacity and zero-error

capacity means that we need to remove exponentially many codewords from any code near the list-decoding capacity line to reach the zero-error line. Thus, almost every codeword in $\mathcal{X}$ is close to some other codeword in $\mathcal{X}$, the two codewords forming a 'bad' pair. More formally, we claim via this argument that

Theorem 9. *For any fixed $0 < \epsilon < 1$, for sufficiently large n, $\frac{|\mathcal{X}_R|}{2}(1-\epsilon)$ such pairs can be found in the large set $\mathcal{X}_R$.*

Note that for this argument to work, it is necessary that $|\mathcal{X}_R| \gg |\mathcal{X}|^{nR(1-\epsilon_R)}$. This gives a lower bound on $\alpha(k)$.

5.3.3 The Adversary

Fix one of the bad pairs defined above, and let the two codewords be $\mathbf{x}_1$ and $\mathbf{x}_2$. The adversary chooses a vector $\mathbf{x}$ equidistant from $\mathbf{x}_1$ and $\mathbf{x}_2$, and pushes both to this $\mathbf{x}$. The previous arguments show that the adversary has enough budget to do so for a large number of codewords (if the adversary did not have this budget, then the capacities involved would be different)

Therefore, when $\mathbf{x}_1$ or $\mathbf{x}_2$ is transmitted, the received vector $\mathbf{y} = \mathbf{x}$. The decoder cannot distinguish between which of the codewords $\mathbf{x}_1$ or $\mathbf{x}_2$ was sent, because the common randomness for both of these is also same by construction.

5.3.4 Probability of Error

We know from equation (19) a lower bound on the number of (m, r) pairs that map to codewords that are in large sets. We also know from the adversary action section 5.3.3 and the pairing argument in section 5.3.2 that the decoder has no information to distinguish between the codewords in almost $\frac{|\mathcal{X}_R|}{2}$ pairs.

Therefore, for any fixed $0 < \epsilon < 1$ (the same ϵ as in theorem 9) the probability of error for any decoder is bounded from below by

$$\mathbb{P}_{err} \geq \left[\frac{2^r \cdot |\mathcal{X}|^k - 2^r \cdot |\mathcal{X}|^{k+\log_{|\mathcal{X}|}(1-2\delta)}}{|\mathcal{X}|^k \cdot 2^r}\right] \cdot \frac{1}{2}(1-\epsilon) \tag{21}$$

Therefore,

$$\mathbb{P}_{err} \geq \delta(1-\epsilon) \tag{22}$$

for any $0 < \epsilon < 1$ and this can be made arbitrarily close to δ. This proves the result under the assumption of no collisions.

5.3.5 Handling collisions

Intuitively, the existence of collisions (as defined in section 5.2) should increase the probability of error. Of each colliding pair, the decoder must fail on at least one of the messages because it has no information as to which of the messages was transmitted and the adversary does not need to do anything for such an error to take place. We now formalise this argument.

Note that if there are exactly $|\mathcal{X}|^m \cdot 2^r$ codewords in the set $\mathcal{X}$, then no collisions can exist. If there are less than that many codewords in the set, then fix some error codeword $\hat{\mathbf{x}}$ and add that to the set $\mathcal{X}$. For every set of colliding (m, r) pairs in the code ψ, map all but one of them to $\hat{\mathbf{x}}$ and

leave one to map to the colliding codeword. Whenever the decoder sees $\hat{\mathbf{x}}$, the decoder declares an error. This creates a new code ψ'. As mentioned above, the adversary doesn't need to do anything for an error to occur when $\hat{\mathbf{x}}$ is transmitted, and by construction, the same number of codewords result in error in ψ' as in ψ. Therefore, the code ψ' has the same error performance as the original code ψ.

Also note that the code ψ' has exactly the same set $\mathcal{R}$ as the original code ψ because none of the r-vectors are removed in the construction. We shall now give a lower-bound the probability of error for ψ'.

We reconstruct the sets as in section 5.2 but we do not include the error codeword $\hat{\mathbf{x}}$ while defining the sets. We do not include edges mapping to the codeword $\hat{\mathbf{x}}$ while calculating the degrees of the R-vertices and retain the same size-threshold as before. Let the total number of (m, r) pairs that map to $\hat{\mathbf{x}}$ be M.

The set A (section 5.3.1) is constructed again, but with $\hat{\mathbf{x}}$ excluded. The same upper bound on the size of the set A follows.

$$|A| < |\mathcal{X}|^{nR+\log_{|\mathcal{X}|}(1-2\delta)} \tag{23}$$

Since by definition the set A does not include $\hat{\mathbf{x}}$, the size of it's pre-image is not affected by the number of collisions. Also, by construction of ψ', $|\mathcal{R}|$ remains the same as before. Therefore, $|\psi^{-1}(A)|$ is bounded from above by the same number as before using the same argument (equation (18).

The pre-image of A^c does change, because the number of (m, r) pairs that map to $\hat{\mathbf{x}}$ must be excluded from this count. The new bound (changing from the bound in equation (19)) is

$$|\psi^{-1}(A^c)| \geq 2^r \cdot |\mathcal{X}|^k - 2^r \cdot |\mathcal{X}|^{k+\log_{|\mathcal{X}|}(1-2\delta)} - M \tag{24}$$

Since the size-threshold remains the same, the argument presented in section 5.3.2 still works, and theorem 9 still holds. The action of the adversary on the non-error codewords remains the same, and the adversary does not do anything when the transmitted codeword is $\hat{\mathbf{x}}$. This means that if a (m, r) pair is in the set $|\psi^{-1}(A^c)|$ and does not map to $\hat{\mathbf{x}}$, then the conditional probability of error is greater than $\frac{1}{2}(1-\epsilon)$ for any $0 < \epsilon < 1$. For the (m, r) pairs that do map to $\hat{\mathbf{x}}$, all of them result in a decoding error. Therefore, we can bound the probability of error from below as follows:

$$\mathbb{P}_{err} \geq \frac{\left(2^r \cdot |\mathcal{X}|^k - 2^r \cdot |\mathcal{X}|^{k+\log_{|\mathcal{X}|}(1-2\delta)} - M\right) \cdot \frac{1}{2}(1-\epsilon) + M}{|\mathcal{X}|^k \cdot 2^r} \tag{25}$$

Simplifying, we get that

$$\mathbb{P}_{err} \geq \delta(1-\epsilon) + \frac{M(1+\epsilon)}{2 \cdot |\mathcal{X}|^k \cdot 2^r} \tag{26}$$

Since M is a non-negative number, the second term is non-negative, and we have thus obtained the desired result.

6 Auxiliary Results

6.1 Capacity under private randomization at encoder/decoder/adversary

Lemma 10. *Randomized jamming strategies do not worsen the error performance.*

Proof. There error is specified w.r.t. deterministic jamming strategies. However, this is not restrictive as randomized jamming strategies do not degrade the error performance any further. To see this, note that the error for message m under some randomized jamming strategy $P_{\mathbf{S}}$ is $\sum_{\mathbf{s}:T_{\mathbf{s}}\in\Lambda_S} \mathbb{P}(\mathbf{S}=\mathbf{s})P_e^{(n)}(m,\mathbf{S}=\mathbf{s}) \leq \max_{\mathbf{s}:T_{\mathbf{s}}\in\Lambda_S} P_e^{(n)}(m,\mathbf{s})$. This implies that there exists a deterministic feasible jamming strategy with the same error performance.

More generally, this communication problem can be viewed as a *min-max communication game* over $P_e^{(n)}$ for the code, where Alice-Bob pair comprise the minimizing player and James comprises the maximizing player. Now, it can be easily seen that for any fixed coding strategy of the minimizing player, whether deterministic or randomized, randomized strategies for the maximizing player (James is the maximizing player here) do not degrade the error performance any more than deterministic strategies. Note however, that the above contention above does not hold when the roles of the players' are reversed; randomized coding strategies may perform better than deterministic ones under any given jamming strategy. Thus, without loss of generality, the error criterion can be specified w.r.t. deterministic jamming strategies. □

Lemma 11. *Capacity does not increase if randomized encoding with a deterministic decoder is allowed.*

Proof. Our proof proceeds by showing that the error ${P_e^{(n)}}_s$ for any given code with a stochastic encoder and a deterministic decoder can be attained by a deterministic code ${P_e^{(n)}}_d$. Hence, it will follow that $\min_{\mathcal{C}_s} {P_e^{(n)}}_s \geq \min_{\mathcal{C}_d} {P_e^{(n)}}_d$. As we already know that $\min_{\mathcal{C}_s} {P_e^{(n)}}_s \leq \min_{\mathcal{C}_d} {P_e^{(n)}}_d$, the result follows.

□

Lemma 12. *Capacity does not increase if randomized decoding with a deterministic encoder is allowed.*

Proof. Let C_{det} and C_{stoc} denote the capacity when stochastic decoding is allowed and when only deterministic decoding is allowed. We assume that the encoder is deterministic under both scenarios.

Note that $C_{det} \leq C_{stoc}$, and hence, the result follows if we show that $C_{stoc} \leq C_{det}$. To establish $C_{stoc} \leq C_{det}$, it is sufficient to show that given any (n, R) code with a stochastic decoder, say $\mathcal{C}$, such that $P_e^{(n)}(\mathcal{C}) \leq \epsilon$, there exists, for n sufficiently large, a code with a deterministic decoder, say $\mathcal{C}'$, such that the corresponding probability of error $P_e^{(n)}(\mathcal{C}') \leq a\epsilon$, where $a > 0$ is an absolute constant.

Recall from earlier that the capacity under maximum and average probability of error is identical (cite relevant lemma). Thus, for the rest of the discussion, we assume the maximum probability of error.

Fix message m. Note that this fixes $\mathbf{x} = \psi(m)$. Then, the probability of error for message m under a feasible adversarial strategy $\mathbf{s} = \mathbf{s}(\mathbf{x})$, where $T_{\mathbf{s}} \in \Lambda_S$, is

$$\begin{aligned} P_e^{(n)}(\mathcal{C}, m) &= \sum_{k=1}^{K} \mathbb{P}(K = k) \sum_{\mathbf{y}:\phi_k(\mathbf{y}) \neq m} \mathbb{P}_W(\mathbf{Y} = \mathbf{y} | \psi(m) = \mathbf{x}, \mathbf{s}) \\ &\geq \min_k \sum_{\mathbf{y}:\phi_k(\mathbf{y}) \neq m} \mathbb{P}_W(\mathbf{Y} = \mathbf{y} | \psi(m) = \mathbf{x}, \mathbf{s}). \end{aligned}$$

Hence, Let the stochastic decoder choose a fixed decoding map $\phi_k : \mathcal{Y}^n \to \{1, 2, \ldots, 2^{nR}\}$ with probability $\mathbb{P}(K = k)$, where $k = 1, 2, \ldots, K$. Let $P_{e,s-d}^{(n)}(m)$ denote the probability of error for message m under this code with a stochastic decoder $\mathcal{C}_{dec}$. Then,

$$\begin{aligned} P_{e,s-d}^{(n)}(m) &= \max_{\mathbf{s}(\mathbf{x}):T_{\mathbf{s}} \in \Lambda_S} \sum_{k=1}^{K} \mathbb{P}(K = k) \sum_{\mathbf{y}:\phi_k(\mathbf{y}) \neq m} \mathbb{P}_W(\mathbf{Y} = \mathbf{y} | \psi(m) = \mathbf{x}, \mathbf{s}) \\ &\geq \max_{\mathbf{s}} \min_k \sum_{\mathbf{y}:\phi_k(\mathbf{y}) \neq m} \mathbb{P}_W(\mathbf{Y} = \mathbf{y} | \psi(m) = \mathbf{x}, \mathbf{s}) \end{aligned}$$

□

6.2 Invariance of capacity under maximum error and average error

Lemma 13. *Capacity is identical under both probability of error criteria, viz., average error and maximum error.*

Let C_{max} and C_{avg} denote the capacity under the maximum and average probability of error criterion respectively. As $C_m \leq C_{avg}$, the result follows once we show that $C_{avg} \leq C_m$. This will follow if we show that for any (n, R) code $\mathcal{C}$ with ${P_e^{(n)}}_{avg}(\mathcal{C}) \leq \epsilon$, $\epsilon > 0$, there exists, for n sufficiently large, an (n, R) code $\mathcal{C}'$ with ${P_e^{(n)}}_{max}(\mathcal{C}') \leq a\epsilon$, where $a > 0$ is an absolute constant.

We use the *expurgation approach* (cf. [**?**, pg. 203], for instance). Let $\mathcal{C}$ be an (n, R) code with ${P_e^{(n)}}_{avg}(\mathcal{C}) \leq \epsilon$, $\epsilon > 0$. Now order the codewords in $\mathcal{C}$ according to their individual probabilities of error and expurgate the worst (in terms of probability of error) half of the codewords. This results in a codebook, say $\mathcal{C}'$, where total number of codewords in $\mathcal{C}'$ is 2^{nR-1}, and $\forall m \in \{1, 2, \cdots, 2^{nR-1}\}$, the corresponding error probability $P_e^{(n)}(\mathcal{C}', m) \leq 2\epsilon$ (otherwise, this would have violated the condition ${P_e^{(n)}}_{avg}(\mathcal{C}) \leq \epsilon$). Thus, the maximum probability of error ${P_e^{(n)}}_{max}(\mathcal{C}') \leq 2\epsilon$. Since the rate R' of codebook $\mathcal{C}'$ is $R' = R - \frac{1}{n}$, and $R' \to R$ as $n \to \infty$, the result now follows.