Computational Wiretap Coding from Indistinguishability Obfuscation

Yuval Ishai
Technion
yuvali@cs.technion.ac.il

Aayush Jain
CMU
aayushja@andrew.cmu.edu

Paul Lou
UCLA
pslou@cs.ucla.edu

Amit Sahai
UCLA
sahai@cs.ucla.edu

Mark Zhandry
NTT Research
mzhandry@gmail.com

Abstract

A wiretap coding scheme for a pair of noisy channels $(\mathsf{ChB}, \mathsf{ChE})$ enables Alice to reliably communicate a message to Bob by sending its encoding over ChB, while hiding the message from an adversary Eve who obtains the same encoding over ChE.

A necessary condition for the feasibility of wiretap coding is that ChB is not a *degradation* of ChE, namely Eve cannot simulate Bob's view. While insufficient in the information-theoretic setting, a recent work of Ishai, Korb, Lou, and Sahai (Crypto 2022) showed that the non-degradation condition *is* sufficient in the computational setting, assuming idealized flavors of obfuscation. The question of basing a similar feasibility result on standard cryptographic assumptions was left open, even in simple special cases.

In this work, we settle the question for all discrete memoryless channels where the (common) input alphabet of ChB and ChE is *binary*, and with arbitrary finite output alphabet, under standard (sub-exponential) hardness assumptions: namely those assumptions that imply indistinguishability obfuscation (Jain-Lin-Sahai 2021, 2022), and injective PRGs. In particular, this establishes the feasibility of computational wiretap coding when ChB is a binary symmetric channel with crossover probability p and ChE is a binary erasure channel with erasure probability e, where $e > 2p$.

On the information-theoretic side, our result builds on a new polytope characterization of channel degradation for pairs of binary-input channels, which may be of independent interest.

1 Introduction

A primary focus of algorithmic coding theory is the construction of codes enabling efficient decoding of noisily perturbed codewords. Along the way, however, we often run into the hardness of recovering from different kinds of noise. For example, a random binary linear code of constant rate allows for efficient decoding of a codeword where each bit of the codeword is *erased* and replaced with a special $\perp$ symbol with some constant probability. In contrast, despite decades of study, we have no efficient algorithms for decoding random binary linear codes when each bit of the codeword can be flipped with any constant probability. Indeed, the conjectured hardness of this task is formalized as the Learning Parity with Noise (LPN) assumption [5]. On the flip side, the contrast between efficient decoding from one kind of noise and hardness of decoding from another serves as a useful basis for a variety of cryptographic primitives, including public-key encryption [2] and much more.

In this work we ask: *How general is this phenomenon?*

For example, can we turn things around and construct a specially-designed code (not a random binary linear code) that allows for efficient decoding from a constant probability p of bit flipping, but where any constant probability $e > 2p$ of erasures makes decoding computationally intractable? Note that if $e \leq 2p$, then the task becomes impossible, since a probability e erasure can be transformed into a probability p bit-flip by simply replacing $\perp$ symbols with random bits. At the same time, to make the question meaningful, we need to choose the parameters so that the erasure decoding problem is still information-theoretically possible. Note that *linear* codes do not suffice in this case, since their decoding can always be done in polynomial time by solving a system of linear equations.

As far as we know, even this very natural and simple question did not have any affirmative answers until very recently – and before this paper, this question had no affirmative answer where the hardness we seek can be reduced to well-studied hardness conjectures.

Wiretap Coding. An information-theoretic study of the above question, where computational hardness is replaced by information loss, was pioneered in the seminal work of Wyner [20] on wiretap channels. Wiretap coding enables secure message transmission using only *unidirectional* communication over noisy channels. This should be contrasted with the use of public-key cryptography for exchanging secret keys, which inherently requires bidirectional communication. In a sense, wiretap coding trades reduced interaction for physical assumptions. Wyner's work has spawned a large body of work in the borderline of information theory and cryptography, and serves as the basis of a research area known as physical layer security. See, e.g., [16] for a survey.

More concretely, given a pair of noisy channels $(\mathsf{ChB}, \mathsf{ChE})$ (here we only consider discrete memoryless channels), a wiretap coding scheme enables Alice to reliably send a message m to an honest Bob by sending a (randomized) encoding of m over ChB. Given the noisy encoding, Bob should be able to decode m with negligible failure probability. On the other hand, an adversary Eve who obtains the same encoding through the channel ChE, should learn essentially nothing about m.

For which pairs of channels $(\mathsf{ChB}, \mathsf{ChE})$ is wiretap coding at all possible? A simple necessary condition is captured by the following notion of (stochastic) *degradation*: We say that ChB is a degradation of ChE if there is a probabilistic function S such that, for every input x, the output of $\mathsf{ChB}(x)$ is identically distributed to $\mathsf{S}(\mathsf{ChE}(x))$. In such a case, it is possible for Eve to use S to perfectly simulate Bob's view, and wiretap coding is clearly impossible.

Is wiretap coding possible whenever ChB is *not* a degradation of ChE? Somewhat unexpectedly, the answer is no. This is implied by a general characterization due to Csiszár and Korner [8]. In the

special case where ChB is a binary symmetric channel BSC_p (flipping each bit with probability p) and ChE is a binary erasure channel BEC_e (erasing each bit with probability e), wiretap coding is possible if and only if $e > 4p(1-p)$ [15], whereas ChB is not a degradation of ChE whenever $e > 2p$.

Computational Wiretap. The above gap begs the following natural question: Is the non-degradation condition sufficient for *computational* wiretap coding, where all parties are computationally bounded? In particular, here security should only hold against a polynomial-time Eve. This question was studied in the recent work of Ishai, Korb, Lou, and Sahai [12], who showed that the non-degradation condition *is* sufficient in this setting, assuming *idealized flavors of obfuscation*. Concretely, rather than rely on the standard *indistinguishability obfuscation* ($i\mathcal{O}$) primitive, which can now be based on well-studied cryptographic assumptions [13, 14], the construction required Alice to send an obfuscated program over a channel, but its analysis treated the program as an oracle, relying on an idealized notion of "virtual black-box" obfuscation [4]. The question of basing a similar feasibility result on standard cryptographic assumptions was left open in [12], even in simple special cases such as the $(\mathsf{BSC}_p, \mathsf{BEC}_e)$ case.

While we now have a sophisticated toolbox of techniques to replace ideal obfuscation by $i\mathcal{O}$ [9, 17], these techniques apply in the context of obfuscating a cryptographic primitive, such as a pseudorandom function, building on the security properties of the primitive. In contrast, the constructions of [12] obfuscate non-cryptographic "evasive" functions [3], which poses a challenge to current techniques of leveraging $i\mathcal{O}$.

1.1 Our Contribution

Our main result settles the computational wiretap coding question, under the standard assumptions that $i\mathcal{O}$ and injective pseudorandom generators (PRGs) exist, for the case where the (common) input alphabet of ChB and ChE is binary. Here the output alphabets can be of any (finite) size.

Theorem 1. *Assuming the existence of $i\mathcal{O}$ and injective PRGs, there exists a computational wiretap coding scheme for any pair of binary-input channels $(\mathsf{ChB}, \mathsf{ChE})$ such that ChB is not a degradation of ChE.*

As a special case, under the same standard assumptions, there is a computational wiretap coding for $(\mathsf{BSC}_p, \mathsf{BEC}_e)$ if (and only if) $e > 2p$. In fact, this settles the broad coding question posed in the beginning of the introduction, with respect to *probabilistic* encodings, for the case of *binary* error-correcting codes with arbitrary channel noise.

On the information-theoretic side, a technical tool we develop for proving Theorem 1 is a complete *polytope characterization* of stochastic channel degradation for pairs of binary-input channels. To state this characterization, we will need the following definition.

Definition 1 (Channel Polytope)**.** Let $\mathbf{A}$ be a real-valued matrix of non-negative entries. We associate to $\mathbf{A}$ the following polytope, denoted $\mathcal{P}(\mathbf{A})$, which can be defined in either of the following equivalent ways:

- $\mathcal{P}(\mathbf{A})$ is the convex hull of all subset-sums of columns of $\mathbf{A}$.
- $\mathcal{P}(\mathbf{A}) = \{\mathbf{A} \cdot \mathbf{s} : \mathbf{0} \leq \mathbf{s} \leq \mathbf{1}\}$

Theorem 2. *Let* $\mathbf{B}, \mathbf{E}$ *be non-negative matrices with two rows that satisfy* $\mathbf{B} \cdot \mathbf{1} = \mathbf{E} \cdot \mathbf{1}$*, representing binary-input channels* ChB *and* ChE *respectively. Then* $\mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E})$ *if and only if* ChB *is a degradation of* ChE*.*

We also show that this characterization does not extend to general input alphabets of size greater than two. That is, we show an explicit counterexample for the case of $\mathbf{B}, \mathbf{E}$ with three rows (ternary input alphabets) where $\mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{B})$ yet ChB is not a degradation of ChE.

Perspective: Average-Case Complexity with Side-Information. One can also view our main result from the lens of average-case complexity in the presence of side-information. One way to design a computational wiretap coding scheme is by constructing hard average-case planted problems (e.g., a planted random CSP or a planted graph problem) with sharp algorithmic thresholds with respect to side information about the planted assignment. We can model such a problem as an inversion problem where we denote $y = \mathsf{P}_{e,p}(x)$, where a planted assignment $x \leftarrow \{0,1\}^n$ is chosen at random and the parameters e, p denote erasure and bit-flip probabilities, respectively. The properties we want are:

- If one is additionally given x' that is formed by erasing an e-fraction of the bits of x at random, recovering x from y should be hard.
- On the other hand, given x' that is formed by flipping a p-fraction of the bits of x at random, recovering x from y becomes easy.

We desire these properties to hold even when e is barely greater than $2 \cdot p$, thereby requiring a very sharp threshold. As an example, consider the case $e = 0.22$ and $p = 0.1$, in which case the x'_{flip} formed by flipping agrees with x on roughly 90% fraction of the bits whereas x'_{erasure} erases out a 22% fraction of the bits. By randomly guessing the erased bits we can come up with a string r_{erasure} that agrees with x on 89% of the bits of x, barely less than the agreement of x'_{flip} with x.

To our surprise, this seemingly very natural class of problems has not been very well studied. One notable example of such study is in the context of Goldreich's one-way functions, that have been shown to have a self-correction property. In particular, Goldreich's one-way functions satisfy the above property when $e = 1$ and $p = \frac{1}{2} - \epsilon$ for any constant $\epsilon > 0$ [6]. In this work, we show that relying on well-studied hardness assumptions we can construct a problem with these exact properties for any choice of parameters satisfying $e > 2p$.

Open Questions. Our work gives rise to several natural open questions.

- Can Theorem 1 be extended to an arbitrary pair of channels satisfying the non-degradation condition, removing the binary-input requirement? The failure of Theorem 2 to extend to this general case is the most immediate roadblock.
- Are strong cryptographic assumptions such as $i\mathcal{O}$, or even "public-key" assumptions, necessary? For instance, does computational wiretap coding for $(\mathsf{BSC}_{0.1}, \mathsf{BEC}_{0.3})$ imply secret key exchange in the plain model?
- Theorem 1 implies *randomized* encoding schemes which support efficient decoding of 0.1-fraction of random errors but cannot be efficiently decoded from a 0.3-fraction of random erasures (though inefficient decoding is possible). Can such codes be constructed more directly? Can the encoding function be made deterministic? See Section 4.1 for discussion.

- Can our technique for replacing ideal obfuscation of a "non-cryptographic" program by $i\mathcal{O}$ be extended to apply to other applications, such as secure computation over unidirectional noisy channels [1]?

2 Technical Overview

We will represent a channel ChB by a row-stochastic matrix $\mathbf{B} \in [0,1]^{2\times n_B}$ and ChE is by a row-stochastic matrix $\mathbf{E} \in [0,1]^{2\times n_E}$ where the (i,j)-th entry of the matrix gives the probability that the ith input alphabet symbols maps to the jth output alphabet symbol when passed through the channel. We are given that ChB is not a degradation of ChE: This means that there does not exist ChS (represented by a row stochastic matrix $\mathbf{S} \in [0,1]^{n_E\times n_B}$) such that $\mathsf{ChB} = \mathsf{ChS} \circ \mathsf{ChE}$ (equivalently $\mathbf{B} = \mathbf{E}\cdot\mathbf{S}$). Throughout this technical overview, we will refer to channels and their row-stochastic matrix representations interchangeably.

2.1 A Construction for BSC_p and BEC_e channels

We begin with a useful special case: the setting in which Bob's channel $\mathbf{B}$ is a BSC_p channel (a sent bit b is received as $1-b$ with probability p and received as b with probability $1-p$) and Eve's channel $\mathbf{E}$ is a BEC_e channel (a sent bit b is erased with probability e and received as b with probability $1-e$) for some channel parameters p and e. As we will see below, handling this case will be fundamental to handling the general case.

The Degradation Condition. In this setting, it is easy to see that ChB is *not* a degradation of ChE if and only if $e > 2p$. That is, Eve's best guessing strategy for each of her erasures to randomly guess a bit, so if $e > 2p$, then Eve cannot hope in expectation to produce a received string with a p error rate. And if $e \leq 2p$, Eve perfectly simulates receiving an output form ChB by randomly assigning a random bit for each of her received erasures (and depending on p, possibly introducing more intentional errors in her received non-erasures).

We now introduce some useful notation. Consider a randomly chosen string $x \in \{0,1\}^\lambda$ for a large (security) parameter λ passed through ChB and ChE. Let z^B denote the string that is received by Bob and z^E the string that is received by Eve. Denote by $\Delta_H(\star,\star)$ the Hamming distance function between two strings and by $\Delta_c(\star,\star)$ the function that outputs the number of indices on which the input strings agree. Since Bob's channel is BSC_p, we have that with high probability $\Delta_c(z^B, x) \approx (1-p)\lambda$ where $\approx$ subsumes $o(\lambda^{0.51})$ additive error given by a standard Chernoff bound. On the other hand, when x is passed through Eve's channel BEC_e we will receive a string $z^E \in \{0,1,\perp\}^\lambda$ where $\perp$ denote erasures. Let $S_\perp$ denote the indices where the erasures occur and $\overline{S_\perp}$ denote the remaining indices. For every index in $\overline{S_\perp}$, z^E will completely agree with x. On the other hand, we have no information about $x_{S_\perp}$ given z^E. Therefore the best we can do is produce a random string v which agrees with $x_{S_\perp}$ at roughly $\frac{|S_\perp|}{2}$ locations. This lets us come up with a string r^E for which $\Delta_c(x, r^E) \approx \frac{|S_\perp|}{2} + |\overline{S_\perp}|$. With high probability, this number is roughly $(1-\frac{e}{2})\cdot\lambda$.

When Bob's channel is not a degradation of Eve's channel we have $\frac{e}{2} > p$. As a consequence, Bob's received string z^B for which $\Delta_c(x, z^B) \approx (1-p)\cdot\lambda$ agrees with x at approximately $(1-p)\cdot\lambda - (1-\frac{e}{2})\cdot\lambda = (\frac{e}{2}-p)\cdot\lambda$ more locations than the best string r^E that Eve can construct for which $\Delta_c(x, r^E) \approx (1-\frac{e}{2})\cdot\lambda$. We want to make use of this fact to build our wiretap coding scheme.

$C(z)$:
Input: $z \in \{0,1\}^\lambda$
Hardwired: $x \in \{0,1\}^\lambda$

1. Check $\Delta_c(x, z) > (1-p) \cdot \lambda - \lambda^{0.9}$.
2. If the check passes, output m otherwise output $\perp$.

Figure 1: Circuit C

Prior Work: Using Ideal Obfuscation. How do we use this observation to construct a computational wiretap coding scheme for this case? Prior work [12] leverages an ideal obfuscation scheme and exactly exploits the above idea. Recall that in the ideal obfuscation model, one can obfuscate any circuit C to produce a functionally equivalent obfuscated circuit $\tilde{C} = \mathcal{O}(C)$, and any efficient adversary that obtains $\tilde{C}$ can be computationally simulated by a polynomial time algorithm Sim that gets oracle access to C but doesn't get $\tilde{C}$ itself. To encode a bit $m \in \{0,1\}$, the prior work suggested that we obfuscate using an ideal obfuscation scheme the circuit C given below. In the program, x is chosen at random from $\{0,1\}^\lambda$. The encoding algorithm sends out $\tilde{C}$ encoded using an appropriate error-correcting coding scheme for ChB, so that Bob can receive $\tilde{C}$ completely. Further, x is sent out through the channel ChB as is. At the end Bob receives $\tilde{C}$ and $\mathsf{ChB}(x) = z^B$. On the other hand, Eve will receive $\mathsf{ChE}(x) = z^E$, and for the security analysis, we assume that Eve is also able to recover $\tilde{C}$. Based on the previous insight, we now make the following observations:

- (Correctness for Bob): As described previously, with overwhelming probability $\Delta_c(x, z^B) > (1-p) \cdot \lambda - o(\lambda^{0.51})$. Therefore, the first check in the description of C will pass if we evaluate it on z^B. Thus, Bob can recover m by computing $\tilde{C}(z^B)$.
- (Security against Eve): We start with the observation that the best string that Eve can construct agrees at about constant fraction $(\frac{e}{2} - p)$ fewer number of indices than the required threshold $(1-p) \cdot \lambda - \lambda^{0.9}$. Furthermore, recall that Eve can be simulated by an efficient Sim that only has oracle access to C. Putting these two observations together, we observe that having oracle access to C is worthless: all queries to query C will produce an output of $\perp$. Thus, the message m is hidden from Eve.

In [12], the authors show that the same template described above can be extended to construct wiretap computational encoding scheme for arbitrary (multi-input/multi-output) channel pairs $(\mathsf{ChB}, \mathsf{ChE})$ satisfying the non-degradation condition.

Leveraging Indistinguishability Obfuscation. The above solution relies on ideal obfuscation, and uses it in a very interesting way. Our goal, however, is to try to solve the wiretap coding problem based on well-studied hardness conjectures, and unfortunately this type of ideal obfuscation is not known to exist under well-studied hardness conjectures. On the other hand, indistinguishability obfuscation $i\mathcal{O}$ has recently been achieved from well-studied hardness conjectures [13]. However, $i\mathcal{O}$ provides a fundamentally different kind of security guarantee compared to ideal obfuscation. $i\mathcal{O}$ guarantees that any two circuits C_0, C_1 of same size and identical input-output behavior must yield computationally indistinguishable obfuscations. What makes it hard to use $i\mathcal{O}$ is that in this case

circuit C in Figure 1 is not functionally equivalent to the always-$\perp$ circuit. In fact, these circuits differ at any points for which $\Delta_c(x, z) > (1-p) \cdot \lambda - \lambda^{0.9}$, which could be an exponential number of points.

Using $i\mathcal{O}$ instead of ideal obfuscation will require some new tools. We now elaborate:

Gadget: PRG with Self-Correction. Consider the following variant of an injective pseudorandom generator denoted by $\mathsf{SC\text{-}PRG}$. $\mathsf{SC\text{-}PRG}_\epsilon$ is indexed with a parameter $\epsilon \in (0, \frac{1}{2}]$ which is some constant. This PRG satisfies the following properties:

- (Polynomial Stretch and Pseudorandomness) Just like a regular PRG, $\mathsf{SC\text{-}PRG}_\epsilon : \{0,1\}^\lambda \to \{0,1\}^{p_\epsilon(\lambda)}$ maps λ bits to $p_\epsilon(\lambda)$ bits for some polynomial $p_\epsilon(\lambda) \gg 2 \cdot \lambda$ (that could depend on ϵ). Further, for a randomly chosen seed $\mathsf{Seed} \in \{0,1\}^\lambda$, $y = \mathsf{SC\text{-}PRG}_\epsilon(\mathsf{Seed})$ is computationally indistinguishable to a truly random string y'.
- (Self-Correction) There exists an efficient algorithm $\mathsf{SC\text{-}PRG}_\epsilon.\mathsf{Self\text{-}Correct}(y, \mathsf{Seed}')$ with the property that for overwhelming choices of $\mathsf{Seed} \in \{0,1\}^\lambda$ it holds that given $y = \mathsf{SC\text{-}PRG}_\epsilon(\mathsf{Seed})$ along with an arbitrary side-information string $\mathsf{Seed}' \in \{0,1\}^\lambda$ that agrees with Seed on at least $(\frac{1}{2} + \epsilon)$ fraction of bits — that is, such that $\Delta_c(\mathsf{Seed}, \mathsf{Seed}') > (\frac{1}{2} + \epsilon)\lambda$ — the algorithm will be able to successfully recover Seed itself.

In fact, the work of [6] showed that Goldreich PRGs (even with linear stretch $\Omega_\epsilon(\lambda)$) satisfy the self-correction property we are looking for. In our work (see below for more intuition), we show how to construct a PRG with self-correction from any injective PRG.

Using PRG with Self-Correction with $i\mathcal{O}$. We now describe how this new gadget can help us leverage the power of $i\mathcal{O}$ in our security proof. To encode $m \in \{0,1\}$, we give out $\tilde{C} = i\mathcal{O}(C)$ as an obfuscation of the circuit C, described in Figure 1, which is encoded using an appropriate coding scheme for Bob's channel ChB so that Bob can reconstruct it. Further x is transported to Bob without any encoding via ChB. In this case, Eve's view consist of $\tilde{C}$ and $\mathsf{ChE}(x) = z^E$. We now describe how we can switch computationally un-detectably an obfuscation of $\tilde{C}$ (even given z^E) from an obfuscation of the circuit in Figure 1 to an obfuscation of the always-$\perp$ circuit, from Eve's point of view.

Step 1: Hardwiring part of Eve's view into the ciruit. As a crucial first step, we observe that in the proof, we can have intermediate hyrbids where the circuit to be obfuscated actually depends on Eve's received string z^E, where z^E is a string in $\{0,1,\perp\}^\lambda$ and $\perp$ denotes an erasure. If $S_\perp$ is the set of erased locations and $\overline{S_\perp}$ is the set of non erased locations, then, we have that $z^E_{S_\perp} = \perp^{|S_\perp|}$ and $z^E_{\overline{S_\perp}} = x_{\overline{S_\perp}}$. In the first step, we replace $\tilde{C}$ to now be an $i\mathcal{O}$ obfuscation of the circuit $C^{(1)}$ described in Figure 2. Notice that circuits $C^{(1)}$ and C are functionally equivalent. This equivalence is because $\Delta_c(x, z) = \Delta_c(x_{S_\perp}, z_{S_\perp}) + \Delta_c(x_{\overline{S_\perp}}, z_{\overline{S_\perp}})$ since the set $S_\perp$ and $\overline{S_\perp}$ form a partition of $[\lambda]$. Therefore, due to $i\mathcal{O}$ security the change is computationally indistinguishable assuming we pad the circuits appropriately before computing the $i\mathcal{O}$ obfuscation.

Step 2: Using the $\mathsf{SC\text{-}PRG}_\epsilon$ scheme. So far, it may seem that we have not done anything interesting. The change was merely syntactical. But now, we observe that Eve has absolutely

$C^{(1)}(z)$:
Input: $z \in \{0,1\}^\lambda$
Hardwired: $x \in \{0,1\}^\lambda$, set $S_\perp$ and $\overline{S_\perp}$

1. Check $\Delta_c(x_{S_\perp}, z_{S_\perp}) + \Delta_c(x_{\overline{S_\perp}}, z_{\overline{S_\perp}}) > (1-p)\cdot\lambda - \lambda^{0.9}$.
2. If the check passes, output m otherwise output $\perp$.

Figure 2: Circuit $C^{(1)}$

zero information about $x_{S_\perp}$, while Bob does! This fact enables us to leverage the self-correcting properties of the $\mathsf{SC\text{-}PRG}$ scheme. We compute $\mathsf{SC\text{-}PRG}_\epsilon(x_{S_\perp}) = y$ where we describe how we set the constant $\epsilon > 0$ shortly. In the new circuit $C^{(2)}$ (Figure 3), we no longer hardwire x completely. This time, we will only hardwire the non-erased portion $x_{\overline{S_\perp}}$ and instead of hardwiring $x_{S_\perp}$, we hardwire $\mathsf{SC\text{-}PRG}_\epsilon(x_{S_\perp}) = y$. To maintain functional equivalence, the circuit $C^{(2)}$ will derive $x_{S_\perp}$ from y.

$C^{(2)}(z)$:
Input: $z \in \{0,1\}^\lambda$
Hardwired: $x_{\overline{S_\perp}} \in \{0,1\}^{\overline{S_\perp}}$, set $S_\perp$ and $\overline{S_\perp}$, $\epsilon = (\frac{1}{2} - \frac{p}{e}) \cdot \frac{1}{2} \in (0, \frac{1}{4}]$ and $y = \mathsf{SC\text{-}PRG}_\epsilon(x_{S_\perp})$,

1. Run $\mathsf{SC\text{-}PRG}_\epsilon.\mathsf{Self\text{-}Correct}(y, z_{S_\perp}) = s$ and check if $s = x_{S_\perp}$ by checking if $\mathsf{SC\text{-}PRG}_\epsilon(s) = y$. If the check fails output $\perp$. Else set $x_{S_\perp} = s$ and continue,
2. Check $\Delta_c(x_{S_\perp}, z_{S_\perp}) + \Delta_c(x_{\overline{S_\perp}}, z_{\overline{S_\perp}}) > (1-p)\cdot\lambda - \lambda^{0.9}$.
3. If the check passes, output m otherwise output $\perp$.

Figure 3: Circuit $C^{(2)}$

The idea is that if the following check passes on any given input z:

$$\Delta_c(x_{S_\perp}, z_{S_\perp}) + \Delta_c(x_{\overline{S_\perp}}, z_{\overline{S_\perp}}) > (1-p)\lambda - \lambda^{0.9}, \tag{1}$$

that means that:

$$\Delta_c(x_{S_\perp}, z_{S_\perp}) > (1-p)\cdot\lambda - \lambda^{0.9} - \Delta_c(x_{\overline{S_\perp}}, z_{\overline{S_\perp}}).$$

The maximum value of $\Delta_c(x_{\overline{S_\perp}}, z_{\overline{S_\perp}}) = |\overline{S_\perp}|$. With overwhelming probability, $|S_\perp| \in [e\cdot\lambda - \lambda^{0.9}, e\cdot\lambda + \lambda^{0.9}]$ and therefore $|\overline{S_\perp}| \in [(1-e)\cdot\lambda - \lambda^{0.9}, (1-e)\cdot\lambda + \lambda^{0.9}]$. Therefore, we have that with overwhelming probability over the size of $S_\perp$, for any z satisfying the check in Equation 1, we have:

$$\Delta_c(x_{S_\perp}, z_{S_\perp}) > (e-p)\cdot\lambda - 2\cdot\lambda^{0.9}.$$

Since with overwhelming probability over erasures, $|S_\perp| \in [e \cdot \lambda - \lambda^{0.9}, e \cdot \lambda + \lambda^{0.9}]$, the above equation can be rephrased as:

$$\Delta_c(x_{S_\perp}, z_{S_\perp}) > \underbrace{(1 - \frac{p}{e})}_{\frac{1}{2}+\epsilon' \ (\epsilon' = \frac{1}{2} - \frac{p}{e})} \cdot |S_\perp| - O(\lambda^{0.9}).$$

Therefore, with overwhelming probability over the number of erasures, for any z satisfying Equation 1 it must hold that $z_{S_\perp}$ agrees with $x_{S_\perp}$ at roughly $1 - \frac{p}{e} = \frac{1}{2} + \epsilon'$ fraction of indices (ignoring the $O(\lambda^{-0.9})$ term), where $\epsilon' = \frac{1}{2} - \frac{p}{e}$. Since $e > 2p$ because of the non-degradation condition, we have $\epsilon' > 0$. To be on the conservative side, we choose the self-correction threshold $\epsilon = \frac{\epsilon'}{2} > 0$.

Putting this all together, in the circuit $C^{(2)}$, we have $y = \mathsf{SC\text{-}PRG}_\epsilon(x_{S_\perp})$ hardwired. The circuit will takes input z, and uses it to "derive" $x_{S_\perp}$ by running $\mathsf{SC\text{-}PRG}_\epsilon.\mathsf{Self\text{-}Correct}(y, z_{S_\perp})$. Then it will perform all the checks as before. Due to self-correction property of $\mathsf{SC\text{-}PRG}$, with overwhelming probability over the locations $S_\perp$, $\overline{S_\perp}$ and the choice of $x_{S_\perp}$ the circuits $C^{(2)}$ and $C^{(1)}$ are functionally equivalent. For any input z that satisfies the check in $C^{(1)}$ given by Equation 1, we will also have that $z_{S_\perp}$ will agree with $x_{S_\perp}$ on at least $\frac{1}{2} + \epsilon$ fraction of indices as argued before. With high probability over $x_{S_\perp}$, it holds that $\mathsf{SC\text{-}PRG}_\epsilon.\mathsf{Self\text{-}Correct}(y, r) = x_{S_\perp}$ for every r that agrees with $x_{S_\perp}$ on at least $\frac{1}{2} + \epsilon$ fraction of indices. Therefore with overwhelming probability on the choice of $S_\perp$ and $x_{S_\perp}$, on input any such z that passes check in Equation 1, we can recover $x_{S_\perp}$ uniquely. Therefore, the two circuits are functionally identical; if the check in Equation 1 does not pass, both circuits output $\perp$ anyway. Because of this, we can appeal to $i\mathcal{O}$ security and argue computational indistinguishability for the change.

The final step: Exploiting Eve's ignorance. Recall that given z^E and conditioned on erased indices $S_\perp$, the distribution $x_{S_\perp}$ is identically uniform. Therefore, we can switch the $i\mathcal{O}$ obfuscation of the circuit $C^{(2)}$ with an obfuscation of $C^{(3)}$ described in Figure 4 where the only change is that now y is sampled as a uniformly random string in $\{0,1\}^{p_\epsilon(|S_\perp|)}$ as opposed to being $y = \mathsf{SC\text{-}PRG}_\epsilon(x_{S_\perp})$. With overwhelming probability $|S_\perp| = \Omega(\lambda)$ and therefore, this change is indistinguishable due to the security of $\mathsf{SC\text{-}PRG}$.

$C^{(3)}(z)$:
Input: $z \in \{0,1\}^\lambda$
Hardwired: $x_{\overline{S_\perp}} \in \{0,1\}^{\overline{S_\perp}}$, set $S_\perp$ and $\overline{S_\perp}$, $\epsilon = (\frac{1}{2} - \frac{p}{e}) \cdot \frac{1}{2} \in (0, \frac{1}{4}]$ and $y \leftarrow \{0,1\}^{p_\epsilon(|S_\perp|)}$,

1. Run $\mathsf{SC\text{-}PRG}_\epsilon.\mathsf{Self\text{-}Correct}(y, z_{S_\perp}) = s$ and check if $s = x_{S_\perp}$ by checking if $\mathsf{SC\text{-}PRG}_\epsilon(s) = y$. If the check fails abort. Else set $x_{S_\perp} = s$ and continue,
2. Check $\Delta_c(x_{S_\perp}, z_{S_\perp}) + \Delta_c(x_{\overline{S_\perp}}, z_{\overline{S_\perp}}) > (1 - p) \cdot \lambda - \lambda^{0.9}$.
3. If the check passes, output m otherwise output $\perp$.

Figure 4: Circuit $C^{(3)}$

Finally observe that if $\mathsf{SC\text{-}PRG}$ is sufficiently expanding, with overwhelming probability for any y that is chosen at random, there will not exist any $x_{S_\perp}$ such that $y = \mathsf{SC\text{-}PRG}_\epsilon(x_{S_\perp})$. Therefore,

with overwhelming probability over y, the circuit $C^{(3)}$ is functionally equivalent to the always-$\perp$ circuit. Thus using $i\mathcal{O}$ security, we can switch $\tilde{C}$ to be an obfuscation of the always-$\perp$ circuit. This finishes the overview of the proof of security of the wiretap scheme.

Constructing Self-Correcting PRGs from any Injective PRGs. One issue that we should address is that as far as we know the only currently known instantiations of such self-correcting PRGs are Goldreich PRGs [6]. We in fact show that any injective PRG suffices for constructing a self-correcting PRG.

Suppose we have an injective PRG $G : \{0,1\}^\lambda \to \{0,1\}^\kappa$ mapping λ bits to κ bits for some polynomial $\kappa(\lambda)$ that we will work out below. Additionally, we use a powerful tool from coding theory: a polynomial rate list-decodable code $\mathcal{C}_{\epsilon'} = (\mathrm{Enc}, \mathrm{Dec})$. The code is parameterized with any constant $\epsilon' \in (0, \frac{1}{2})$ and satisfies the following properties:

- (Polynomial rate) The encoding algorithm Enc is a polynomial deterministic algorithm mapping $\mathrm{Enc} : \{0,1\}^\lambda \to \{0,1\}^n$ for some polynomial $n(\lambda) > \lambda$.
- (List Decoding) The Dec algorithm is a polynomial time algorithm with the property that for any c such that $\Delta_H(c, \mathrm{Enc}(\alpha)) < (\frac{1}{2} - \epsilon') \cdot n$, the algorithm outputs a list of size at most $\mathsf{poly}(\lambda, \frac{1}{\epsilon'})$ of elements in $\{0,1\}^\lambda$ that contains α.

Such a coding scheme exists by [19] and many appropriate schemes have been well-explored [18, 11, 10]. Once we have both these ingredients, the function $\mathsf{SC\text{-}PRG}_\epsilon$ can be described as follows. On input a string $x = (x_1 \in \{0,1\}^n, x_2 \in \{0,1\}^\lambda)$, to compute $\mathsf{SC\text{-}PRG}_\epsilon(x) = y$ we evaluate $y_2 = G(x_2)$ and $r = \mathrm{Enc}(x_2)$. We then set $y_1 = r \oplus x_1$ and output $y = (y_1, y_2)$. If n and ϵ' satisfy some mild parameteric requirements that we arrive at below, we claim that this construction satisfies the properties we need.

Observe that $\mathsf{SC\text{-}PRG}_\epsilon$ is injective. Given $\mathsf{SC\text{-}PRG}_\epsilon(x_1, x_2) = (y_1, y_2)$, y_2 binds x_2 uniquely as $y_2 = G(x_2)$ and G is injective. As a consequence $y_1 = x_1 \oplus \mathrm{Enc}(x_2)$ binds x_1 once x_2 is determined. Similarly, $\mathsf{SC\text{-}PRG}_\epsilon(x_1, x_2) = (y_1, y_2)$ also satisfies pseudorandomness. This is because $y_1 = x_1 \oplus \mathrm{Enc}(x_2)$ and $y_2 = G(x_2)$. Since x_1 is random and independent of x_2, we have that y_1 hides x_2. As a consequence given y_1, it is the case that $y_2 = G(x_2)$ is pseudorandom due to the security of G. Therefore (y_1, y_2) is pseudorandom.

Most importantly, if the parameters are set appropriately, $\mathsf{SC\text{-}PRG}_\epsilon$ also satisfies the self-correction property. Imagine we have $(y_1, y_2) = \mathsf{SC\text{-}PRG}_\epsilon(x_1, x_2)$ and z such that $\Delta_H(z, (x_1, x_2)) < (\frac{1}{2} - \epsilon)|x| = (\frac{1}{2} - \epsilon)(\lambda + n)$. We want to show that such a z lets us recover x.

Note that $\Delta_H(z, (x_1, x_2)) < (\frac{1}{2} - \epsilon)(\lambda + n)$ means that $\Delta_H(z_1, x_1) < (\frac{1}{2} - \epsilon)n + \lambda$ where $z_1 \in \{0,1\}^n$ is the first n length sub-string of z. If $n \gg \lambda$, we have $\Delta_H(z_1, x_1) < (\frac{1}{2} - \epsilon')n$ for some constant ϵ' barely less than ϵ. If the list-decodable coding scheme can correct from $(\frac{1}{2} - \epsilon')$ fractions of errors and G is injective, given such a z we can derive x using the following steps:

- We first compute $c = y_1 \oplus z_1$ and then compute a list $L = \mathrm{Dec}(c)$ of polynomial size.
- Find $\alpha \in L$ such that $G(\alpha) = y_2$ (such an α must be equal to x_2).
- Finally, output x_1 computing $y_1 \oplus \mathrm{Enc}(x_2)$.

The reason why this algorithm succeeds is that $y_1 = \mathrm{Enc}(x_2) \oplus x_1$ and if z_1 is such that $\Delta_H(z_1, x_1) < (\frac{1}{2} - \epsilon')n$, we have that $y_1 \oplus z_1$ satisfies $\Delta_H(y_1 \oplus z_1, \mathrm{Enc}(x_2)) < (\frac{1}{2} - \epsilon')n$. Therefore, due to the list decoding property of the code and the injectivity of G the second condition will produce $\alpha = x_2$. Since $y_1 = x_1 \oplus \mathrm{Enc}(x_2)$, we can derive x_1 correctly in the third step.

2.2 Tackling general channels with binary input alphabets: An Overview

Above we saw our construction for a simple pair of binary input channels based on $i\mathcal{O}$ and an injective PRG. We now need to build a computational wiretap coding scheme for pairs of general binary input channels. Before we give detailed technical ideas, we first provide a guide to the remainder of this technical overview.

Step 1: Coding scheme for BAC-BAEC pair. In the first step, we will construct a coding scheme for a case that is just a little more general than the case considered above. In this case, Bob's channel matrix $\mathbf{B}$ is an arbitrary binary channel in $[0,1]^{2\times 2}$ (such channels are called binary asymmetric channels BAC, see Definition 6). Such BAC's are a generalization of a binary symmetric channel. While there is a single parameter p that determines a binary symmetric channel BSC_p (the probability of flipping a bit b to $1-b$ is p independently of the bit b), there are two parameters p_0, p_1 for the binary asymmetric channel BAC_{p_0,p_1} that respectively define the probability of a 0 flipping and the probability of a 1 flipping, and these probabilities p_0 and p_1 may differ. Eve's channel is a generalization of the binary erasure channel, called a binary asymmetric erasure channel (BAEC, see Definition 7) whose row-stochastic matrix representation is in $[0,1]^{2\times 3}$. While a binary erasure channel BEC_e is parameterised by a single parameter e, where the probability of erasing any given bit $b \in \{0,1\}$ is e independently of b, there are two parameters e_0, e_1 for the asymmetric channel BAEC_{e_0,e_1} where the probability of erasure for any given bit b is e_b and these two parameters may differ. We will show how to build a computational wiretap encoding scheme that works as long as Bob's channel BAC_{p_0,p_1} is not a degradation of Eve's channel BAEC_{e_0,e_1}.

Step 2: Bootstrapping Step 1 to the general binary input case. Why should a computational wiretap coding scheme for the case when Bob's channel is of the form BAC_{p_0,p_1} and Eve's channel is of the form BAEC_{e_0,e_1} suffice for the general binary input case? In general, $\mathbf{B}$ and $\mathbf{E}$ can be completely arbitrary channels with arbitrary constant-sized input and output alphabets.

We show that as long as $\mathbf{B} \in [0,1]^{2\times n_B}$ and $\mathbf{E} \in [0,1]^{2\times n_E}$ are binary input channels (with potentially larger constant sized output alphabets), the above solution is fully general! To obtain this result, we show a series of implications:

- (Polytope characterization of non-degradation) For any channel $\mathbf{C} \in \mathbb{R}^{2\times n_C}$, we define $\mathcal{P}(\mathbf{C})$ as the bounded convex set:

$$\mathcal{P}(\mathbf{C}) = \left\{ \mathbf{C}\cdot\mathbf{u} \mid \mathbf{u} \in [0,1]^{n_c\times 1} \right\}.$$

  In simple words, this is a bounded convex-set in the two-dimensional plane $\mathbb{R}^2$ that is generated by the taking combinations of columns of $\mathbf{C}$ where the coefficients of each column in the combination are in $[0,1]$. Our main characterization theorem states that for any pair of binary-input channels (with potentially a different number of outputs) $\mathbf{B} \in \mathbb{R}^{2\times n_B}$ and $\mathbf{E} \in \mathbb{R}^{2\times n_E}$:

$$\mathbf{B} \text{ is a degradation of } \mathbf{E} \iff \mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E}).$$

  It turns out that this characterization does not extend to non-binary input alphabets. We give an explicit counter-example to the claim if the input alphabet is ternary.

- Using the simple characterization described above, for any binary input channel pair $(\mathbf{B}, \mathbf{E})$ where $\mathbf{B}$ is not a degradation of $\mathbf{E}$, we (efficiently) find two channels $\mathbf{B}' \in \mathbb{R}^{2\times 2}$ and $\mathbf{E}' \in \mathbb{R}^{2\times 3}$ such that the following properties hold:
  - $\mathbf{B}'$ is of the form BAC_{p_0,p_1} for some p_0, p_1 and $\mathbf{E}'$ is of the form BAEC_{e_0,e_1} for some e_0, e_1.
  - $\mathcal{P}(\mathbf{B}') \subseteq \mathcal{P}(\mathbf{B})$. In other words, using the characterization above, $\mathbf{B}'$ can be simulated by $\mathbf{B}$.
  - $\mathcal{P}(\mathbf{E}) \subseteq \mathcal{P}(\mathbf{E}')$. In other words, using the characterization above, $\mathbf{E}$ can be simulated by $\mathbf{E}'$.
  - Further, $\mathcal{P}(\mathbf{B}') \not\subseteq \mathcal{P}(\mathbf{E}')$. In other words, using the characterization above, $\mathbf{B}'$ is not a degradation of $\mathbf{E}'$.
- Using the observations above, we can use the computational wiretap coding scheme for the BAC-BAEC case to construct a computational wiretap coding scheme for the general binary input case where we use the base encoding scheme for the BAC-BAEC case effectively treating Bob's channel as $\mathbf{B}'$ and Eve's channel as $\mathbf{E}'$. In slightly more detail, while the physical channel to Bob is given by $\mathbf{B}$, Bob can simulate $\mathbf{B}'$ via a post-processing procedure allowing Bob to recover the message bit using the base encoding scheme. On the other hand, while the physical channel to Eve is given by $\mathbf{E}$, we show that an even more leaky BAEC channel $\mathbf{E}'$ which is enough to simulate $\mathbf{E}$ would not suffice to recover the message bit.

We describe the intuition behind both these steps next. In Section 2.3, we describe how we extend the above construction ideas to a computational wiretap coding scheme for the BAC-BAEC case. Finally, in Section 2.4, we discuss the polytope characterization for non-degradation of binary input channels and how use this polytope characterization to find channels $\mathbf{B}'$ and $\mathbf{E}'$ as described above to bootstrap our computational wiretap coding scheme for the BAC-BAEC case to a construction for *any* pair of non-degraded binary input channels.

2.3 Generalization to Asymmetric Erasures/Flips

To describe the ideas behind our base computational wiretap encoding scheme for the case when $\mathbf{B}$ is of the form BAC_{p_0,p_1} and $\mathbf{E}$ is of the form BAEC_{e_0,e_1}, let us understand for what parameter settings of p_0, p_1 and e_0, e_1, we have that BAC_{p_0,p_1} is not a degradation of BAEC_{e_0,e_1}. Without loss of generality, we can assume that p_0 is less than or equal to $\frac{1}{2}$. If this is not the case, we can post-process ChB with the channel given by the permutation matrix:

$$\mathbf{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

yielding Bob's channel to be $\mathsf{BAC}_{p'_0=1-p_0,p'_1=1-p_1}$ which satisfies our requirement $p'_0 \leq \frac{1}{2}$. This transformation also does not change the polytope for Bob as this transformation just swaps the columns of Bob's matrix.

2.3.1 Relation between Erasure/Flip probabilities for Non-Degradation

If $\mathbf{B}$ is a matrix corresponding to a BAC_{p_0,p_1} then, it can be expressed as:

$$\mathbf{B} = \begin{bmatrix} 1-p_0 & p_0 \\ p_1 & 1-p_1 \end{bmatrix}.$$

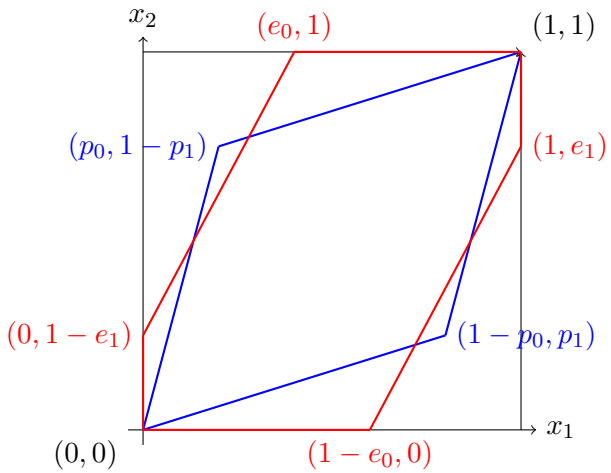

Figure 5: An example of polytope non-containment for binary asymmetric channels and binary asymmetric erasure channels. Here, x_1 and x_2 are indeterminates. The blue polytope is $\mathcal{P}(\mathsf{BAC}_{p_0,p_1})$ for parameters $p_0 = 1/5, p_1 = 1/4$. The red polytope is $\mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$ for parameters $e_0 = 2/5$, $e_1 = 3/4$.

Similarly, if $\mathbf{E}$ is a matrix corresponding to a BAEC_{e_0,e_1} then, it can be expressed as:

$$\mathbf{E} = \begin{bmatrix} 1-e_0 & 0 & e_0 \\ 0 & 1-e_1 & e_1 \end{bmatrix}.$$

Recall that as described above $\mathbf{B}$ is not a degradation of $\mathbf{E}$ if and only if $\mathcal{P}(\mathbf{B}) \not\subseteq \mathcal{P}(\mathbf{E})$.

One can draw these polytopes in $\mathbb{R}^2$, and a representative picture looks like the one we describe in Figure 5. The red polytope depicts the polytope $\mathcal{P}(\mathbf{E})$ and the blue polytope represents $\mathcal{P}(\mathbf{B})$. In order to show non-degradation, by the polytope criterion, we only need to show that the point $(p_0, 1-p_1)$ is not inside $\mathcal{P}(\mathbf{E})$. This translates to having the point $(p_0, 1-p_1)$ to be on the opposite side of the line joining $(e_0, 1)$ with $(0, 1-e_1)$ as the origin. When we compute this condition, we get the criterion:

$$e_0 \cdot e_1 > p_1 \cdot e_0 + p_0 \cdot e_1. \tag{2}$$

A formal proof can be found in Lemma 10.

2.3.2 Extending the Construction to the Asymmetric Setting

We now describe how to extend our construction ideas to handle the general (asymmetric) case when Bob's channel is BAC_{p_0,p_1} and Eve's channel is BAEC_{e_0,e_1} where the parameters p_0, p_1, e_0, e_1 are potentially differing values (conditioned on ChB not being a degradation of ChE). Below, we recall the mathematical formulation of the non-degradation condition described in Equation 2.

$$e_0 \cdot e_1 > p_1 \cdot e_0 + p_0 \cdot e_1.$$

Our construction in this case is largely similar to the construction for the BSC_p-BEC_e case with some important modifications.

$C(z)$:
Input: $z \in \{0,1\}^\lambda$
Hardwired: $x \in \{0,1\}^\lambda$

1. Check $\Delta_c(x,z) > (1 - \frac{(p_0e_1+p_1e_0)}{e_0+e_1})\lambda - \lambda^{0.9}$.
2. If the check passes, output m otherwise output $\perp$.

Figure 6: Circuit C

As before, to encode a message $m \in \{0,1\}$, as a first step we sample a string $x \in \{0,1\}^\lambda$. However, instead of sampling each bit of x uniformly at random from $\{0,1\}$, we sample each bit of x independently to be zero with probability $\eta = \frac{e_1}{e_0+e_1}$ and one with probability $1-\eta = \frac{e_0}{e_0+e_1}$ (therefore the distribution $\mathsf{Ber}^\lambda_{1-\eta}$). The second modification is the threshold condition in the circuit we will obfuscate. Once we have such an x, we compute $\tilde{C}$ which is now an obfuscation of the circuit C in Figure 6.

Notice that in the circuit C described in Figure 6, the threshold for the number of agreeing bits has changed to a constant fraction $(1 - \frac{(p_0e_1+p_1e_0)}{e_0+e_1})$.

We now describe why these two changes yield a computational wiretap encoding scheme for this case. The rationale behind this is that x is chosen so that each bit of x is zero with probability $\eta = \frac{e_1}{e_0+e_1}$. Further, Bob's channel ChB is BAC_{p_0,p_1} and flips a bit b with probability p_b. As a consequence, Bob's received string z^B will agree with x on an expected $(1-p_0)\eta + (1-p_1)(1-\eta) = \left(1 - \frac{(p_0e_1+p_1e_0)}{e_0+e_1}\right)$ fraction of bits, which is more than the threshold we set.

What can Eve do? Eve will receive a string $z^E \in \{0,1\}^\lambda$ that contains erasures and is in $\{0,1,\perp\}^\lambda$. As before, by let $z^E_{S_\perp}$ denote the erased part and $z^E_{\overline{S_\perp}}$ denote the rest of the string which is also equal to $x_{\overline{S_\perp}}$. For this distribution, the size $|\overline{S_\perp}|$ in expectation can be computed to be $((1-e_0)\eta + (1-e_1)(1-\eta))\lambda = \left(1 - \frac{2e_0e_1}{e_0+e_1}\right) \cdot \lambda$. What is also crucial for us and sheds a light on how η is chosen is that one can show using a simple probability analysis that conditioned on z^E (equivalently $x_{\overline{S_\perp}}$), the conditional distribution of the erased part $x_{S_\perp}$ is actually a uniform distribution. Therefore, to come up with a maximum number of agreeing bits, what Eve can essentially do is to use up every bit of z^E corresponding to the non-erased set $\overline{S_\perp}$ and make a random guess v corresponding to the set $S_\perp$. This lets Eve come up with a string r such that $r_{\overline{S_\perp}} = x_{\overline{S_\perp}}$ and $r_{S_\perp} = v$, which satisfies:

$$\Delta_c(r,x) \approx \frac{|S_\perp|}{2} + |\overline{S_\perp}|$$

In expectation, this value is a fraction $(1 - \frac{2e_0e_1}{e_0+e_1}) + \frac{e_0e_1}{e_0+e_1} = 1 - \frac{e_0e_1}{e_0+e_1}$.

This implies that Bob's string agrees with x at $\epsilon' = \left(1 - \frac{(p_0e_1+p_1e_0)}{e_0+e_1}\right) - \left(1 - \frac{e_0e_1}{e_0+e_1}\right) = \frac{e_0e_1-(p_0e_1+p_1e_0)}{e_0+e_1}$ more locations than Eve's best string r. The non-degradation condition that we work out in Equation 2 posits that $e_0e_1 > p_1e_0 + p_0e_1$ and this implies that $\epsilon' > 0$. For a successful Eve this means that it must come up with a string r such that $r_{S_\perp}$ agrees with $x_{S_\perp}$ on at least $(\frac{1}{2} + \gamma')|S_\perp|$ indices for some constant $\gamma' > 0$.

This rough intuition can be massaged into a proof. As before, we will make the following indistinguishable changes:

- As before, for the first change, we will program the string $z^E, S_\perp$ and $\overline{S_\perp}$ into the obfuscated circuit. We will replace the check $\Delta_c(x,z) > (1 - \frac{(p_0e_1+p_1e_0)}{e_0+e_1})\lambda - \lambda^{0.9}$ with a functionally equivalent check $\Delta_c(x_{S_\perp}, z_{S_\perp}) + \Delta_c(x_{\overline{S_\perp}}, z_{\overline{S_\perp}}) > \left(1 - \frac{(p_0e_1+p_1e_0)}{e_0+e_1}\right)\lambda - \lambda^{0.9}$.

- Then, just like in the symmetric case, instead of hardwiring $x_{S_\perp}$ in the circuit we will hardwire the value $y = \mathsf{SC\text{-}PRG}_\gamma(x_{S_\perp})$ where we set γ to be a constant barely less than γ'. Then, instead of using $x_{S_\perp}$ which we no longer have, the program will first derive $x_{S_\perp}$ using self-correction feature of $\mathsf{SC\text{-}PRG}_\gamma$ relying on y and a z that successfully pass our check. This circuit is indistinguishable because with high probability, for any input z that satisfies $\Delta_c(x_{S_\perp}, z_{S_\perp}) + \Delta_c(x_{\overline{S_\perp}}, z_{\overline{S_\perp}}) > \left(1 - \frac{(p_0e_1+p_1e_0)}{e_0+e_1}\right)\lambda - \lambda^{0.9}$, it must also hold that $\Delta_c(x_{S_\perp}, z_{S_\perp}) > \left(\frac{1}{2} + \gamma\right)|S_\perp|$ as argued above.

- Next, we will replace y with a truly random string. This change is indistinguishable due to the security of $\mathsf{SC\text{-}PRG}_\gamma$. Observe that because the conditional distribution $x_{S_\perp}$ given z^E is uniform and $|S_\perp| = \Omega(\lambda)$, $y = \mathsf{SC\text{-}PRG}_\gamma(x_{S_\perp})$ is pseudorandom.

- Once y is a random string, with high probability it will no longer have preimages with respect to $\mathsf{SC\text{-}PRG}$. Therefore, with high probability, the circuit under consideration is functionally equivalent to an all reject circuit. We can now use $i\mathcal{O}$ security to replace this circuit with an all reject circuit.

2.4 Reducing the General Binary Input Case to the Asymmetric Setting

We now describe how we construct a computational wiretap coding scheme for pairs of general non-degraded binary-input channels. To this extent, a reader might wonder why the polytope characterization below is both natural and useful for this purpose.

Theorem 3. *(Informal) Let $\mathbf{B} \in \mathbb{R}^{2\times n_B}$ and $\mathbf{E} \in \mathbb{R}^{2\times n_E}$ be arbitrary row-stochastic matrices. Then, $\mathbf{B} \neq \mathbf{E}\cdot\mathbf{S}$ for every row stochastic matrix $\mathbf{S}$ if and only if $\mathcal{P}(\mathbf{B}) \not\subseteq \mathcal{P}(\mathbf{E})$.*

The usefulness of this theorem is found by considering the following natural approach to construct a computational wiretap encoding scheme for a general binary input channel pair ($\mathbf{B} \in \mathbb{R}^{2\times n_B}, \mathbf{E} \in \mathbb{R}^{2\times n_E}$) such that $\mathbf{B}$ is not a degradation of $\mathbf{E}$:

- **Output Reduction for Bob**: Find a stochastic matrix $\mathbf{S}_B \in \mathbb{R}^{n_B\times 2}$ such that $\mathbf{B}' = \mathbf{B}\cdot\mathbf{S}_B$ is not a degradation of $\mathbf{E}$. In particular, at the end of this step, this yields us with a BAC channel $\mathbf{B}'$ such that there does not exist a stochastic matrix $\mathbf{S}$ such that $\mathbf{B}\cdot\mathbf{S}_B = \mathbf{B}' = \mathbf{E}\cdot\mathbf{S}$.

- **Simulating Eve's channel by a BAEC**: In the next step, we want to find an erasure channel BAEC_{e_0,e_1}, $\mathbf{E}' \in \mathbb{R}^{2\times 3}$ such that $\mathbf{E} = \mathbf{E}'\cdot\mathbf{S}_E$ for some stochastic matrix $\mathbf{S}_E$ (in other words Eve's channel is a degradation of $\mathbf{E}'$). Importantly, it must hold that $\mathbf{B}'$ must not be a degradation of $\mathbf{E}'$. That is, there should not exist any stochastic matrix $\mathbf{S}$ such that $\mathbf{B}' = \mathbf{E}'\cdot\mathbf{S}$.

- **Using a solution for BAC_{p_0,p_1}-BAEC_{e_0,e_1}:** Once we have $\mathbf{B}'$ and $\mathbf{E}'$ satisfying the criteria described above, we can leverage a computational wiretap scheme for the BAC_{p_0,p_1}-BAEC_{e_0,e_1} case. We will treat Bob's channel to be $\mathbf{B}'$ (which can be simulated by Bob) and Eve's channel to be $\mathbf{E}'$ (which can simulate Eve).

The question is: can such matrices $\mathbf{B}'$ and $\mathbf{E}'$ be found? We show that for the above approach to materialize, for the binary input channels the polytope condition in Theorem 3 is both necessary and sufficient.

The necessity can be seen just from the first condition. We want that there must exist a stochastic $\mathbf{S}_B \in \mathbb{R}^{n_B \times 2}$ such that there does not exist any stochastic matrix $\mathbf{S} \in \mathbb{R}^{n_E \times 2}$ for which it holds that:

$$\mathbf{B} \cdot \mathbf{S}_B = \mathbf{E} \cdot \mathbf{S}$$

Notice that $\mathbf{B}' = \mathbf{B} \cdot \mathbf{S}_B$ is of a very special form. It is of the form $\mathbf{B}' = [\mathbf{v}|\mathbf{v}']$ where due to the properties of stochastic matrices, the first column is some vector $\mathbf{v} \in \mathcal{P}(\mathbf{B})$, whereas the second column is simply $\mathbf{v}' = \mathbf{1} - \mathbf{v}$ where $\mathbf{1}$ is the all ones column matrix. When we have that $[\mathbf{v}|\mathbf{v}'] \neq \mathbf{E} \cdot \mathbf{S}$, then this must mean that $\mathbf{v} \notin \mathcal{P}(\mathbf{E})$. If it was not the case, then $\mathbf{v} = \mathbf{E}\mathbf{w}$ for a column vector $\mathbf{w} \in [0,1]^{n_E}$. Then, we can set $\mathbf{S} = [\mathbf{w}|\mathbf{1} - \mathbf{w}]$ which will satisfy $[\mathbf{v}|\mathbf{v}'] = \mathbf{E} \cdot \mathbf{S}$ giving us a contradiction.

2.4.1 Constructing $\mathbf{B}'$ and $\mathbf{E}'$

Assuming that Theorem 3 holds, how do we find such a $\mathbf{B}'$ and $\mathbf{E}'$ in finite time (we assume that channel description is "constant-sized").

Finding $\mathbf{B}'$. To find such a matrix $\mathbf{B}'$ one can find a vector $\mathbf{v} \in \mathcal{P}(\mathbf{B}) \setminus \mathcal{P}(\mathbf{E})$. For such a vector, $\mathbf{v} = \mathbf{B} \cdot \mathbf{a}$ for some $\mathbf{a} \in [0,1]^{n_B \times 1}$. We can set $\mathbf{B}' = [\mathbf{v} \mid \mathbf{1} - \mathbf{v}] = \mathbf{B} \cdot \mathbf{S}_B$ for the stochastic matrix $\mathbf{S}_B$ where $\mathbf{S}_B = [\mathbf{a} \mid \mathbf{1} - \mathbf{a}]$. Observe that $\mathbf{v} \in \mathcal{P}(\mathbf{B}') \setminus \mathcal{P}(\mathbf{E})$, therefore $\mathbf{B}'$ is not a degradation of $\mathbf{E}$ as per Theorem 3.

How do we find $\mathbf{v}$? Note that both $\mathcal{P}(\mathbf{B})$ and $\mathcal{P}(\mathbf{E})$ are convex bodies with finitely many extreme points. Since $\mathcal{P}(\mathbf{B})$ is not contained inside $\mathcal{P}(\mathbf{E})$, there must be an extreme point of $\mathcal{P}(\mathbf{B})$ not contained inside $\mathcal{P}(\mathbf{E})$. Furthermore, the set of extreme points of $\mathcal{P}(\mathbf{B})$ are contained inside the set $\{\mathbf{B} \cdot \mathbf{b} \mid \mathbf{b} \in \{0,1\}^{n_B}\}$. For each of these points, the non-containment can also be checked efficiently using a linear program.

Finding $\mathbf{E}'$. Perhaps what might seem really surprising is that we can actually find a channel matrix $\mathbf{E}'$ that is highly structured (of the form BAEC_{e_0,e_1}) so that it is powerful enough to simulate $\mathbf{E}$, but not enough to simulate $\mathbf{B}$, for any pair of channel matrices $\mathbf{B} \in \mathbb{R}^{2\times 2}$ and $\mathbf{E} \in \mathbb{R}^{2\times n_E}$ satisfying the non-degradation condition.

The equivalent polytope condition actually gives rise to a very intuitive geometric approach to show this. The idea is that $\mathcal{P}(\mathbf{E})$ is a bounded convex body in $[0,1]^2$, and there is a point $\mathbf{v} \in \mathcal{P}(\mathbf{B}') \setminus \mathcal{P}(\mathbf{E})$ so there exists a separating hyperplane that strictly separates $\mathbf{v}$ from $\mathcal{P}(\mathbf{E})$. This separating hyperplane, a line in two-dimensions, will form a facet of a new channel polytope that defines a binary asymmetric erasure channel. Since $(0,0)$ and $(1,1)$ are in both $\mathcal{P}(\mathbf{B}')$ and $\mathcal{P}(\mathbf{E})$ the line should stay "above" the line joining $(0,0)$ and $(1,1)$. This line will intersect the line $x_1 = 0$ at a point $(0, 1 - e_1)$ for some $e_1 > 0$ and the line $x_2 = 1$ at $(e_0, 1)$ some $e_0 > 0$. By two-fold rotational symmetry, we can find another parallel line intersecting $x_2 = 0$ at $(1 - e_0, 0)$ and $x_1 = 1$ at $(1, e_1)$

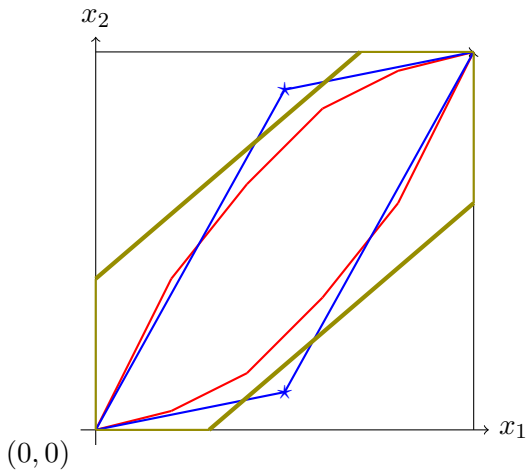

Figure 7: The blue polytope (for ChB) is not contained in the red polytope (for ChE), so ChB is not a degradation of ChE. Using separating hyperplanes (olive-colored lines) we can strictly separate the blue extreme points from the red polytope. The olive polytope corresponds to a binary asymmetric erasure channel ChE' that contains the red polytope but does not contain the blue polytope, i.e. ChE is a degradation of ChE' and ChB is not a degradation of ChE'.

that separates the point $\mathbf{1} - \mathbf{v}$ from $\mathcal{P}(\mathbf{E})$. The area formed that is between the two parallel lines inside $[0,1]^2$ can be represented by the channel matrix as required:

$$\mathbf{E}' = \begin{bmatrix} 1-e_0 & 0 & e_0 \\ 0 & 1-e_1 & e_1 \end{bmatrix}.$$

See Figure 7 for a visual depiction.

2.4.2 Proving the Polytope Characterization.

To show the polytope characterization theorem (Theorem 3), we first observe that one direction is straightforward. To show that if ChB is a degradation of ChE, then $\mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E})$, we simply open up all the definitions. By definition of the polytope formulation, for any point $\mathbf{x} \in \mathcal{P}(\mathbf{B})$ there exists a vector $\mathbf{s}$ such that $\mathbf{x} = \mathbf{B} \cdot \mathbf{s}$ where $\mathbf{0} \leq \mathbf{s} \leq \mathbf{1}$. From the definition of stochastic degradation, there is a row-stochastic matrix $\mathbf{S}$ such that $\mathbf{B} = \mathbf{E} \cdot \mathbf{S}$. Then $\mathbf{x} = \mathbf{E} \cdot (\mathbf{S} \cdot \mathbf{s}) = \mathbf{E} \cdot \mathbf{s}'$ where $\mathbf{0} \leq \mathbf{s}' \leq \mathbf{1}$ since $\mathbf{S}$ is stochastic.

Showing the converse that $\mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E})$ implies the existence of a row-stochastic matrix $\mathbf{S}$ such that $\mathbf{B} = \mathbf{E} \cdot \mathbf{S}$ is more involved. A natural approach is by induction on the number of columns of $\mathbf{B}$. For this induction approach, we will relax the row-stochastic condition on $\mathbf{B}$ and $\mathbf{S}$, which states that non-negative matrices $\mathbf{B}$ and $\mathbf{E}$ satisfy $\mathbf{B} \cdot \mathbf{1} = \mathbf{1} = \mathbf{E} \cdot \mathbf{1}$, and instead assume the more general condition $\mathbf{B} \cdot \mathbf{1} = \mathbf{E} \cdot \mathbf{1}$ for non-negative matrices $\mathbf{B}$ and $\mathbf{E}$.

1. In the base case, if $\mathbf{B}$ consists of one column, then $\mathbf{B} \cdot \mathbf{1} = \mathbf{B}$ so we can take the row-stochastic $\mathbf{S} = \mathbf{1}$ and observe that $\mathbf{E} \cdot \mathbf{1} = \mathbf{B} \cdot \mathbf{1} = \mathbf{B}$.

2. In the induction step, we consider a matrix $\mathbf{B}'$ which is constructed from $\mathbf{B}$ by removing a column of $\mathbf{B}$ so that $\mathbf{B} = \left[\mathbf{v} \mid \mathbf{B}'\right]$. Observe that $\mathbf{B}' \cdot \mathbf{1} = \mathbf{B} \cdot \mathbf{1} - \mathbf{v}$.

The induction hypothesis is that if $\mathcal{P}(\mathbf{B}') \subseteq \mathcal{P}(\mathbf{E}')$ for some matrix $\mathbf{B}'$ of fewer columns than $\mathbf{B}$, and some $\mathbf{E}'$ such that $\mathbf{B}' \cdot \mathbf{1} = \mathbf{E}' \cdot \mathbf{1}$, then there exists a row-stochastic matrix $\mathbf{S}'$ such that $\mathbf{B}' = \mathbf{E}' \cdot \mathbf{S}'$.

To see how we might apply the induction hypothesis, observe that $\mathcal{P}(\mathbf{B}') = \mathcal{P}(\mathbf{B}) \cap (\mathcal{P}(\mathbf{B}) - \mathbf{v})$ where we define the set $\mathcal{P}(\mathbf{B}) - \mathbf{v} \coloneqq \{\mathbf{u} - \mathbf{v} : \mathbf{u} \in \mathcal{P}(\mathbf{B})\}$. This immediately implies the following polytope containment relation

$$\mathcal{P}(\mathbf{B}') = \mathcal{P}(\mathbf{B}) \cap (\mathcal{P}(\mathbf{B}) - \mathbf{v}) \subseteq \mathcal{P}(\mathbf{E}) \cap (\mathcal{P}(\mathbf{E}) - \mathbf{v})$$

To apply the induction hypothesis, we need to find a matrix $\mathbf{E}'$ such that $\mathcal{P}(\mathbf{E}') = \mathcal{P}(\mathbf{E}) \cap (\mathcal{P}(\mathbf{E}) - \mathbf{v})$ such that $\mathbf{E}' \cdot \mathbf{1} = \mathbf{B}' \cdot \mathbf{1}$. To find this matrix $\mathbf{E}'$, we turn to the (two-dimensional) geometric view of the polytopes: $\mathcal{P}(\mathbf{E}) \cap (\mathcal{P}(\mathbf{E}) - \mathbf{v})$ is the intersection of a polytope and its translated polytope. This intersection, visually, is a polytope obtained by starting with the polytope $\mathcal{P}(\mathbf{E})$ and shrinking the length of its facets (lines) by some multiplicative factor in the interval $[0, 1]$. This geometric intuition is exactly captured by the existence of some diagonal matrix $\mathbf{D}$, whose entries are in the closed interval $[0, 1]$, such that $\mathcal{P}(\mathbf{E}) \cap (\mathcal{P}(\mathbf{E}) - \mathbf{v}) = \mathcal{P}(\mathbf{E} \cdot \mathbf{D})$. Thus, we set $\mathbf{E}' = \mathbf{E} \cdot \mathbf{D}$.

It remains to show that $\mathbf{E} \cdot \mathbf{D} \cdot \mathbf{1} = \mathbf{B}' \cdot \mathbf{1}$. To see why this is true, observe that by non-negativity $\mathbf{E} \cdot \mathbf{1}$ is the maximal element (in the ℓ_1-norm) of $\mathcal{P}(\mathbf{E})$ so $\mathbf{E} \cdot \mathbf{1} - \mathbf{v}$ is the maximal element of $\mathcal{P}(\mathbf{E}) - \mathbf{v}$. Then, $\mathbf{E} \cdot \mathbf{1} - \mathbf{v} \in \mathcal{P}(\mathbf{E})$ by definition of the polytope formulation ($\mathbf{v} = \mathbf{E} \cdot \mathbf{u}'$ for some $\mathbf{0} \leq \mathbf{u}' \leq \mathbf{1}$). Therefore, $\mathbf{E} \cdot \mathbf{1} - \mathbf{v}$ is the maximal element of $\mathcal{P}(\mathbf{E}) \cap (\mathcal{P}(\mathbf{E}) - \mathbf{v}) = \mathcal{P}(\mathbf{E} \cdot \mathbf{D})$. This fact implies that $\mathbf{E} \cdot \mathbf{D} \cdot \mathbf{1} = \mathbf{E} \cdot \mathbf{1} - \mathbf{v} = \mathbf{B} \cdot \mathbf{1} - \mathbf{v} = \mathbf{B}' \cdot \mathbf{1}$.

Applying the induction hypothesis, we now have a row-stochastic matrix $\mathbf{S}'$ such that $\mathbf{B}' = \mathbf{E} \cdot \mathbf{D} \cdot \mathbf{S}'$. To conclude the induction step, we set $\mathbf{S} = \left[\mathbf{1} - \mathbf{D} \cdot \mathbf{1} \;\middle|\; \mathbf{D} \cdot \mathbf{S}'\right]$ and observe that $\mathbf{E} \cdot \mathbf{S} = \mathbf{B}$ and $\mathbf{S} \cdot \mathbf{1} = \mathbf{1}$.

Counterexample for the many input-case. At first glance, it may seem to be without loss of generality to consider binary input channels, since Alice is honest and can anyway choose to use only binary inputs. However, there can exist channels with non-binary inputs where Bob's channel is not a degradation of Eve's channel, and yet every projection of those channels to only two inputs always yields channels where Bob's channel is a degradation of Eve's channel. Such pairs of channels, however, are not common. Nevertheless, if future work is to tackle the case of non-binary input channels, this issue will present a challenge.

One might wonder if our polytope characterization holds for channels with larger number of inputs $k > 2$. Such a claim would indeed be useful to extend our approach to handle to an arbitrary case when $\mathbf{B} \in \mathbb{R}^{k \times n_B}$ and $\mathbf{E} \in \mathbb{R}^{k \times n_E}$. Unfortunately it turns out that such a claim is untrue whenever $k \geq 3$. Intriguingly, we can come up with an explicit choice for stochastic matrices $\mathbf{B} \in \mathbb{R}^{3 \times 3}$ and $\mathbf{E} \in \mathbb{R}^{3 \times 4}$ such that $\mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E})$ and yet there does not exist any stochastic matrix such that $\mathbf{B} = \mathbf{E} \cdot \mathbf{S}$. We point the reader to Section 5.2 for our counterexample.

3 Preliminaries

Throughout this paper, we will use the notation $[n] = \{1, 2, 3, \ldots, n\}$. Let $\mathbf{1}$ denote the all-ones column vector whose length can be clearly inferred in the various contexts. We will use the usual

convention that rows are probability vectors. A row-stochastic matrix $\mathbf{M}$ is a matrix whose rows add up to 1; equivalently, $\mathbf{M}$ satisfies $\mathbf{M} \cdot \mathbf{1} = \mathbf{1}$.

Definition 2. Two probability ensembles $\{\mathcal{D}_{0,\lambda}\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{D}_{1,\lambda}\}_{\lambda\in\mathbb{N}}$ are computationally indistinguishable if there exists a negligible function $\mu : \mathbb{N} \to [0,1]$ such that for all $\lambda \in \mathbb{N}$, for all polynomial-time non-uniform algorithms $\mathcal{A}$,

$$\left| \Pr_{x\sim D_{0,\lambda}} \left[\mathcal{A}(1^\lambda, x) = 1\right] - \Pr_{x\sim D_{1,\lambda}} \left[\mathcal{A}(1^\lambda, x) = 1\right] \right| \leq \mu(\lambda).$$

We will use the shorthand notation to denote the existence of such a negligible function μ:

$$\{\mathcal{D}_{0,\lambda}\}_{\lambda\in\mathbb{N}} \approx_c \{\mathcal{D}_{1,\lambda}\}_{\lambda\in\mathbb{N}}$$

or we will use the following shorthand notation to denote computationally indistinguishability with an explict negligible function μ:

$$\{\mathcal{D}_{0,\lambda}\}_{\lambda\in\mathbb{N}} \approx_\mu \{\mathcal{D}_{1,\lambda}\}_{\lambda\in\mathbb{N}}.$$

3.1 Chernoff Bounds

We will use the following Chernoff bounds.

Lemma 1. *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables taking values in $\{0,1\}$, and let $X = \sum_{i=1}^n X_i$ and $\mathbb{E}[X] = \mu$. Then for $0 < \delta < 1$,*

$$\Pr\left[X \geq (1+\delta)\mu\right] \leq \exp\left(\frac{-\delta^2\mu}{3}\right)$$

and

$$\Pr\left[X \leq (1-\delta)\mu\right] \leq \exp\left(\frac{-\delta^2\mu}{2}\right).$$

Lemma 2. *Let $X_1, \ldots, X_n$ be independent Bernoulli 0/1 random variables . Let $X = \sum_{i=1}^n X_i$ and let $p = \frac{\mathbb{E}[X]}{n}$. Then for all $\varepsilon \geq 0$:*

$$\Pr\left[\frac{1}{n}X \geq p + \varepsilon\right] \leq e^{-2\varepsilon^2 n}$$

and for all $0 \leq \varepsilon < p$,

$$\Pr\left[\frac{1}{n}X \leq p - \varepsilon\right] \leq e^{-2\varepsilon^2 n}.$$

3.2 Channels and Wiretap Coding

Definition 3 (Discrete Memoryless Channel)**.** A discrete memoryless channel (DMC) $\mathsf{ChW} : \mathcal{X} \to \mathcal{Y}$ is a randomized function from input alphabet $\mathcal{X}$ to output alphabet $\mathcal{Y}$. Let $p_W(y \mid x)$ denote the probability that we observe $y \in \mathcal{Y}$ after sending $x \in \mathcal{X}$ through ChW. For $x \in \mathcal{X}$, we use $\mathsf{ChW}(x)$ to denote a random variable over $\mathcal{Y}$ such that for $y \in \mathcal{Y}$,

$$\Pr[\mathsf{ChW}(x) = y] = p_W(y \mid x).$$

We associate ChW with its row-stochastic matrix

$$\mathbf{W} = [p_W(y \mid x)]_{x \in \mathcal{X}, y \in \mathcal{Y}}$$

so that $\mathbf{W} \cdot \mathbf{1} = \mathbf{1}$. For $n \in \mathbb{N}$ and $r = (r_1, \ldots, r_n) \in \mathcal{X}^n$, we define

$$\mathsf{ChW}(r) = \mathsf{ChW}(r_1) \ldots \mathsf{ChW}(r_n).$$

For two channels $\mathsf{ChW} : \mathcal{X} \to \mathcal{Y}$ and $\mathsf{ChV} : \mathcal{Y} \to \mathcal{Z}$, we use $\mathsf{ChV} \circ \mathsf{ChW}$ to denote their concatenation $\mathsf{ChV}(\mathsf{ChW}(\cdot))$. Whenever we discuss channels in the context of efficient algorithms, we assume all channels have finite description size with constant alphabet size and rational probabilities.

Definition 4 (Binary Symmetric Channel (BSC))**.** A binary symmetric channel with crossover probability p, denoted as BSC_p is a DMC with binary input and binary output such that on input bit b, it outputs $1 - b$ with probability p and b otherwise.

Definition 5 (Binary Erasure Channel (BEC))**.** A binary erasure channel with erasure probability e, denoted as BEC_e, is a DMC with binary input and output $\{0, 1, \perp\}$ such that on input bit b, it outputs $\perp$ (i.e. erases the bit) with probability e and b otherwise.

Definition 6 (Binary Asymmetric Channel (BAC))**.** A binary asymmetric channel with crossover probabilities (p_0, p_1), denoted as BAC_{p_0,p_1}, is a DMC with binary input and binary output such that on input bit b, the channel outputs $1 - b$ with probability p_b and b with probability $1 - p_b$. The associated row-stochastic matrix is given by

$$\begin{bmatrix} 1 - p_0 & p_0 \\ p_1 & 1 - p_1 \end{bmatrix}.$$

Definition 7 (Binary Asymmetric Erasure Channel (BAEC))**.** A binary asymmetric channel with erasure probabilities (e_0, e_1), denoted as BAEC_{e_0,e_1}, is a DMC with binary input and ternary output in $\{0, 1, \perp\}$ such that on input bit b, the channel outputs $\perp$ with probability e_b and b with probability $1 - e_b$. The associated row-stochastic matrix is given by

$$\begin{bmatrix} 1 - e_0 & 0 & e_0 \\ 0 & 1 - e_1 & e_1 \end{bmatrix}.$$

Remark 1. In the symmetric case, we can assume that a channel BSC_p has $p \leq 1/2$ without loss of generality because the receiver can always flip its interpretation of the received bit. In the asymmetric setting, by the same reasoning we can assume without loss of generality that $p_0 \leq 1/2$ (but not both p_0 and p_1).

If one channel can be used to simulate another channel, we say that the latter is a degradation of the former. More formally, we recall the well-established notion of stochastic channel degradation.

Definition 8 (Stochastic Degradation)**.** We say that channel ChB is a degradation of channel ChE if there exists a channel ChS such that $\mathsf{ChB} = \mathsf{ChS} \circ \mathsf{ChE}$. Equivalently, ChB is a degradation of ChE if there exists a row-stochastic matrix $\mathbf{S}$ such that

$$\mathbf{B} = \mathbf{E} \cdot \mathbf{S},$$

where $\mathbf{B}$ is the row-stochastic matrix of ChB and $\mathbf{E}$ is the row-stochastic matrix of ChE.

Definition 9 (Wiretap Channel)**.** A wiretap channel is a pair of DMCs $(\mathsf{ChB}, \mathsf{ChE})$ where $\mathsf{ChB} : \mathcal{X} \to \mathcal{Y}$ and $\mathsf{ChE} : \mathcal{X} \to \mathcal{Z}$ share the same input alphabet $\mathcal{X}$.

We now recall the definition of wiretap coding schemes in the setting of a computationally bounded adversary.

Definition 10 (Computational wiretap coding [12])**.** A pair of PPT algorithms $\Pi = (\mathrm{Enc}, \mathrm{Dec})$ is a *computational secure wiretap coding scheme* for wiretap channel $(\mathsf{ChB}, \mathsf{ChE})$ and message space $\mathcal{M} = \{0,1\}$, if there exists a negligible function $\epsilon(\lambda)$ such that

- **Correctness**: For every message $m \in \{0,1\}$,

$$\Pr[\mathrm{Dec}(1^\lambda, \mathsf{ChB}(\mathrm{Enc}(1^\lambda, m))) = m] \geq 1 - \epsilon(\lambda)$$

- **Security**: For all polynomial-time non-uniform adversaries $\mathcal{A}$,

$$\Pr[\mathcal{A}(1^\lambda, \mathsf{ChE}(\mathrm{Enc}(1^\lambda, b))) = b] \leq \frac{1}{2} + \epsilon(\lambda)$$

  where b is uniformly distributed over $\{0,1\}$.

3.3 Indistinguishability Obfuscation

Definition 11 (Indistinguishability Obfuscation ($i\mathcal{O}$) for Circuits, Imported from [13])**.** A PPT algorithm $i\mathcal{O}$ is an indistinguishability obfuscator for (polynomial-sized) circuits if the following holds:

- **Completeness:** For every $\lambda \in \mathbb{N}$, every circuit C with input length n, every input $x \in \{0,1\}^n$ we have that

$$\Pr\left[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(1^\lambda, C)\right] = 1$$

- **Indistinguishability:** For every two ensembles $\{C_{0,\lambda}\}, \{C_{1,\lambda}\}$ of polynomial-sized circuits that have the same size, input length, and output length, and are functionally equivalent (in the sense that for all $\lambda \in \mathbb{N}$, $C_{0,\lambda}(x) = C_{1,\lambda}(x)$ for every input x), the following distributions are computationally indistinguishable:

$$\{i\mathcal{O}(1^\lambda, C_{0,\lambda})\} \approx_c \{i\mathcal{O}(1^\lambda, C_{1,\lambda})\}$$

  That is, for all polynomial-time non-uniform algorithms $\mathcal{A}$, there exists a negligible function $\mu : \mathbb{N} \to [0,1]$ such that for all λ,

$$\left|\Pr\left[\mathcal{A}(1^\lambda, i\mathcal{O}(1^\lambda, C_{0,\lambda})) = 1\right] - \Pr\left[\mathcal{A}(1^\lambda, i\mathcal{O}(1^\lambda, C_{1,\lambda})) = 1\right]\right| \leq \mu(\lambda).$$

3.4 Error-Correcting Codes

For any two binary strings x and y of the same length n, let $\Delta_H(x,y)$ denote their Hamming distance and let $\delta_H(x,y) = \frac{\Delta_H(x,y)}{n}$ denote their relative Hamming distance.

Definition 12. A q-ary code of block length n and dimension k is given by a function $C : \mathcal{M} \to \subseteq [q]^n$ where $|\mathcal{M}| = q^k$ and $\mathcal{M}$ is the message space and $[q]$ is the alphabet of C. Such a code is also referred as a $(n,k)_q$ code.

Definition 13. An ensemble of codes $\{C_\lambda : \mathcal{M}_\lambda \to [q_\lambda]^{n_\lambda}\}$ is $(p(\cdot), L(\cdot))$-list decodable, where $p : \mathbb{N} \to (0,1)$ and let $L : \mathbb{N} \to \mathbb{N}$ are functions in some parameter λ, if there is a polynomial-time algorithm $\mathsf{ListDec}$ such that for all $\lambda \in \mathbb{N}$, for all $y \in [q_\lambda]^{n_\lambda}$, $\mathsf{ListDec}(\lambda, y)$ outputs a list S of size at most $L(\lambda)$ messages such that S contains all $m \in \mathcal{M}_\lambda$ such that $\delta_H(C_\lambda(m), y) \leq p(\lambda)$.

Lemma 3 (Implicit in [19], also Theorem 9 in [18])**.** *For every ε, k, if $n \geq \mathsf{poly}\left(k, \frac{1}{\varepsilon}\right)$, there exists an $(n,k)_2$ code with a polynomial time list-decoding algorithm for up to $(\frac{1}{2} - \varepsilon) \cdot n$ errors.*

3.5 Separating Hyperplane Theorem

For $x \in \mathbb{R}^n$ and $S \subseteq \mathbb{R}^n$, we define $\langle S, x\rangle = \{\langle v, x\rangle : v \in S\}$.

Theorem 4 (Strict Separating Hyperplane Theorem [7])**.** *Let $C, K \subseteq \mathbb{R}^n$ be nonempty convex sets with $C \cap K = \emptyset$. If C is closed and K compact, then there exists $\psi \in \mathbb{R}^n$ with $\inf\langle C, \psi\rangle > \sup\langle K, \psi\rangle$.*

4 The BSC-BEC Case

We first consider the simpler setting when Bob's channel is BSC_p and Eve's channel is BEC_e where Bob's channel is not a degradation of Eve's (which happens exactly when $e > 2p$). In this setting, we present a simple construction of a computational wiretap coding scheme based on the existence of $i\mathcal{O}$ and an injective one-way function. Our "code-offset" based construction for this setting also motivates our construction for the general binary input wiretap channels that will be presented next.

Theorem 5 (Computational wiretap for the BSC-BEC case)**.** *Assuming the existence of $i\mathcal{O}$ and an injective one-way function, there exists a computational wiretap coding scheme for a wiretap channel of the form $(\mathsf{BSC}_p, \mathsf{BEC}_e)$ if and only if $e > 2p$.*

The "only if" direction follows from the fact that when $e \leq 2p$, Bob's channel BSC_p is a degradation of Eve's channel BEC_e. In fact, this direction holds unconditionally. We thus focus on constructing a computational wiretap coding scheme when $e > 2p$ using $i\mathcal{O}$ and an injective one-way function. The construction is described in the following figure.

Coding Scheme 1. (**Computational Wiretap Coding Scheme for** $(\mathsf{BSC}_p, \mathsf{BEC}_e)$)

For the construction, we will use an $i\mathcal{O}$ scheme and any error-correcting coding scheme $\mathcal{C}_B = (\mathcal{C}_B.\mathrm{Enc}, \mathcal{C}_B.\mathrm{Dec})$ for the channel BSC_p ($p < 1/2$) such that for all $x \in \{0,1\}^*$,

$$\Pr[\mathcal{C}_B.\mathrm{Dec}(1^\lambda, \mathsf{BSC}_p(\mathcal{C}_B.\mathrm{Enc}(1^\lambda, x))) = x] \geq 1 - \varepsilon(\lambda)$$

for some negligible function ε. For example, even a simple repetition code of block length λ suffices.
$\mathrm{Enc}(1^\lambda, b)$:

1. Let $\delta_{th} = \lambda^{-0.1}$.
2. Sample $r \leftarrow \{0,1\}^\lambda$ uniform randomly.

3. Construct a circuit (whose size is determined by λ) for the function $f : \{0,1\}^\lambda \to \{0,1,\perp\}$ defined as follows:

   $f_\lambda(x)$:
   Input: $x \in \{0,1\}^\lambda$.
   Hardwired constants: r, b.

   1. If the Hamming distance $\Delta_H(x, r) < (p + \delta_{th}) \cdot \lambda$, then output b. Else, output $\perp$.

4. Output $(\mathcal{C}_B.\mathrm{Enc}(1^\lambda, i\mathcal{O}(f_\lambda)), r)$ where f_λ is padded to be the maximum circuit size of itself and $\{f_\lambda^{(i)}\}_{i\in[4]}$ which are described below.

$\mathrm{Dec}(1^\lambda, \hat{f}, z)$:

1. Let $f \leftarrow \mathcal{C}_B.\mathrm{Dec}(1^\lambda, \hat{f})$.
2. Output $f(z)$.

Having explained the intuition for the above coding scheme in the technical overview (Section 2), we proceed to the formal proofs.

Lemma 4 (Correctness of the Computational Wiretap Encoding Scheme)**.** *There exists a negligible function $\varepsilon : \mathbb{N} \to [0,1]$ such that for every message bit $b \in \{0,1\}$,*

$$\Pr[\mathrm{Dec}(1^\lambda, \mathsf{ChB}(\mathrm{Enc}(1^\lambda, b))) = b] \geq 1 - \varepsilon(\lambda).$$

Proof. We will use (f, r) to denote random variables representing the output of $\mathrm{Enc}(1^\lambda, b)$ and we will use $(\tilde{f}, \tilde{r})$ to denote random variables representing the output of the channel $\mathsf{ChB}(f, r)$. Using the notation $\Delta_H(\cdot,\cdot)$ to denote Hamming distance (non-relative), we note that the expected value of the Hamming distance of $\tilde{r}$ from r is given as $\mathbb{E}_{\mathsf{ChB},\mathrm{Enc}}[\Delta_H(\tilde{r}, r)] = p \cdot \lambda$. Then the probability over the channel randomness and the coins used by the encoding algorithm $\mathrm{Enc}(\cdot,\cdot)$ that the received string $\tilde{r}$ fails the statistical check is given by an additive Chernoff (Lemma 2) bound:

$$\Pr_{\mathsf{ChB},\mathrm{Enc}}[\Delta_H(\tilde{r}, r) \geq (p + \delta_{th}) \cdot \lambda] \leq \exp\left(-2 \cdot \delta_{th}^2 \cdot \lambda\right) = \exp\left(-2 \cdot \lambda^{0.8}\right)$$

which is negligible in λ.

□

Lemma 5. *Let ChB be a BSC_p channel and let ChE be a BEC_e channel such that $e > 2p$. For all polynomial-time non-uniform algorithms $\mathcal{A}$, there exists a negligible function $\mu : \mathbb{N} \to [0,1]$ such that*

$$\Pr[\mathcal{A}(1^\lambda, \mathsf{ChE}(\mathrm{Enc}(1^\lambda, b))) = b] \leq \frac{1}{2} + \mu(\lambda)$$

where b is uniformly distributed over $\{0,1\}$.

Proof. We will proceed through the following series of hybrids (experiments) that model Eve's view. We will show that what Eve receives from this encoding process is computationally indistinguishable

from Eve receiving a null circuit, thereby rendering Eve unable to recover the message bit b except with negligible advantage. In each of the following hybrids, each function (viewed as a circuit) is padded to be the maximum circuit size of the circuits in $\{f_\lambda\} \cup \left\{f_\lambda^{(i)}\right\}_{i\in[4]}$ where f_λ is defined in the construction and $f_\lambda^{(i)}$ are defined in each of the below hybrids.

1. $H_0(1^\lambda)$: In the real world, Alice sends $\mathrm{Enc}(1^\lambda, b) = (\mathcal{C}.\mathrm{Enc}(i\mathcal{O}(f)), r)$ through ChE and Eve receives the output of the channel, $\mathsf{ChE}((\mathcal{C}.\mathrm{Enc}(i\mathcal{O}(f))), r)$. We assume that Eve successfully recovers $i\mathcal{O}(f)$, since such an assumption only gives Eve more information. The output of the experiment is $(i\mathcal{O}(f), \mathsf{ChE}(r))$.

2. $H_1(1^\lambda)$: In this hybrid, we consider a slight variation of the above experiment. Let $r \leftarrow \{0,1\}^\lambda$ be chosen as in $\mathrm{Enc}(1^\lambda, b)$ where each bit r_i is independently identically sampled uniform randomly. Then let $\hat{r} \coloneqq \mathsf{ChE}(r) \in \{0,1,\perp\}^\lambda$. Let $S_\perp \subseteq [\lambda]$ be the set of indices for which $\hat{r}_i = \perp$ and let $\overline{S_\perp} \coloneqq [\lambda] \setminus S_\perp$. Let $\kappa \coloneqq |S_\perp|$. Define the finite subsequence (a string) $r_{S_\perp} \coloneqq (r_{i_j})_{i_j \in S_\perp \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^\kappa$ consisting of the bits from the indices from $S_\perp$ and analogous finite subsequence $r_{\overline{S_\perp}} \coloneqq (r_{i_j})_{i_j \in \overline{S_\perp} \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^{\lambda-\kappa}$ of the bits from the indices from $\overline{S_\perp}$.

   We now give an alternate encoding method where instead of constructing the function f as in Coding Scheme 1, Alice instead uses the following function $f_\lambda^{(1)}$:

   > $f_\lambda^{(1)}(x)$:
   > **Input**: $x \in \{0,1\}^\lambda$
   > **Hardwired constants**: $r_{S_\perp}, r_{\overline{S_\perp}}, b, e_0, e_1, p_0, p_1, S_\perp$.
   >
   > 1. If the Hamming distances satisfy $\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) \leq (p + \delta_{th}) \cdot \lambda$, then output b. Else, output $\perp$.

   The output of the experiment is $(i\mathcal{O}(f_\lambda^{(1)}), \hat{r})$.

3. $H_2(1^\lambda)$: Let $r \leftarrow \{0,1\}^\lambda$ be chosen as in $\mathrm{Enc}(1^\lambda, b)$ where each bit r_i is sampled uniform randomly. Then let $\hat{r} \coloneqq \mathsf{ChE}(r) \in \{0,1,\perp\}^\lambda$. Let $S_\perp \subseteq [\lambda]$ be the set of indices for which $\hat{r}_i = \perp$ and let $\overline{S_\perp} \coloneqq [\lambda] \setminus S_\perp$. Let $\kappa \coloneqq |S_\perp|$. Define the finite subsequence (a string) $r_{S_\perp} \coloneqq (r_{i_j})_{i_j \in S_\perp \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^\kappa$ consisting of the bits from the indices from $S_\perp$ and analogous finite subsequence $r_{\overline{S_\perp}} \coloneqq (r_{i_j})_{i_j \in \overline{S_\perp} \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^{\lambda-\kappa}$ of the bits from the indices from $\overline{S_\perp}$.

   We now give an alternate encoding method where instead of constructing the function $f^{(1)}$ as in $H_1(1^\lambda)$, Alice will do the following in order to construct a different function $f^{(2)}$ which we will define shortly:

   (a) Let $\varepsilon = \frac{1}{4} - \frac{p}{2e}$. Let $C_{LD,\kappa} : \{0,1\}^{\kappa^d} \to \{0,1\}^\kappa$ be a code from a $(1/2 - \varepsilon, q(\kappa, 1/\varepsilon))$-list decodable ensemble of binary codes for some constant $0 < d < 1$ and some polynomial $q(\kappa, 1/\varepsilon)$. We will use $C_{LD,\kappa}.\mathsf{ListDec}(\cdot)$ to denote an efficient list-decoding function for $C_{LD,\kappa}$.

   (b) Sample $\alpha \in \{0,1\}^{\kappa^d}$ uniform randomly and set $c \leftarrow C_{LD,\kappa}(\alpha)$ so $c \in \{0,1\}^\kappa$.

(c) Let $z = c \oplus r_{S_\perp}$.

(d) Let $G : \{0,1\}^{\kappa^d} \to \{0,1\}^{3\cdot\kappa^d}$ be a length-tripling injective PRG.

$f_\lambda^{(2)}(x)$:
Input: $x \in \{0,1\}^\lambda$
Hardwired constants: $r_{\overline{S_\perp}}, , z, G(\alpha), b, e_0, e_1, p_0, p_1, S_\perp$.

1. Let $D \leftarrow C_{LD,\kappa}.\mathsf{ListDec}(z \oplus x_{S_\perp})$. D is a list of at most $q(\kappa, 1/\varepsilon)$ many elements in $\{0,1\}^{\kappa^d}$.
2. If $G(s) \neq G(\alpha)$ for all strings $s \in D$, output $\perp$. Otherwise, set α' to be the string s such that $G(s) = G(\alpha)$.
3. Set $r_{S_\perp} \leftarrow C_{LD,\kappa}(\alpha') \oplus z$.
4. If the Hamming distances satisfy $\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) \leq (p + \delta_{th}) \cdot \lambda$, then output b. Else, output $\perp$.

The output of the experiment is $(i\mathcal{O}(f_\lambda^{(2)}), \hat{r})$.

4. $H_3(1^\lambda)$: Let $r \leftarrow \{0,1\}^\lambda$ be chosen as in $\mathrm{Enc}(1^\lambda, b)$ where each bit r_i is independently identically sampled uniform randomly. Then let $\hat{r} := \mathsf{ChE}(r) \in \{0,1,\perp\}^\lambda$. Let $S_\perp \subseteq [\lambda]$ be the set of indices for which $\hat{r}_i = \perp$ and let $\overline{S_\perp} := [\lambda] \setminus S_\perp$. Let $\kappa := |S_\perp|$. Define the finite subsequence (a string) $r_{S_\perp} := (r_{i_j})_{i_j \in S_\perp \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^\kappa$ consisting of the bits from the indices from $S_\perp$ and analogous finite subsequence $r_{\overline{S_\perp}} := (r_{i_j})_{i_j \in \overline{S_\perp} \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^{\lambda-\kappa}$ of the bits from the indices from $\overline{S_\perp}$.

We now give an alternate encoding method where instead of constructing the function $f^{(1)}$ as in $H_1(1^\lambda)$, Alice will do the following in order to construct a different function $f^{(2)}$ which we will define shortly:

(a) Let $\varepsilon = \frac{1}{4} - \frac{p}{2e}$. Let $C_{LD,\kappa} : \{0,1\}^{\kappa^d} \to \{0,1\}^\kappa$ be a code from a $(1/2 - \varepsilon, q(\kappa, 1/\varepsilon))$-list decodable ensemble of binary codes for some constant $0 < d < 1$ and some polynomial $q(\kappa, 1/\varepsilon)$. We will use $C_{LD,\kappa}.\mathsf{ListDec}(\cdot)$ to denote an efficient list-decoding function for $C_{LD,\kappa}$.

(b) Sample $\alpha \in \{0,1\}^{\kappa^d}$ uniform randomly and set $c \leftarrow C_{LD,\kappa}(\alpha)$ so $c \in \{0,1\}^\kappa$.

(c) Let $z \leftarrow c \oplus r_{S_\perp}$.

(d) Let $G : \{0,1\}^{\kappa^d} \to \{0,1\}^{3\cdot\kappa^d}$ be a length-tripling injective PRG.

(e) Let R be a string sampled uniform randomly from $\{0,1\}^{3\cdot\kappa^d}$.

$f_\lambda^{(3)}(x)$:
Input: $x \in \{0,1\}^\lambda$
Hardwired constants: $r_{\overline{S_\perp}}, z, R, b, e_0, e_1, p_0, p_1, S_\perp$.

1. Let $D \leftarrow C_{LD,\kappa}.\mathsf{ListDec}(z \oplus x_{S_\perp})$. D is a list of at most $q(\kappa, 1/\varepsilon)$ many elements in $\{0,1\}^{\kappa^d}$.

2. If $G(s) \neq R$ for all strings $s \in D$, output $\perp$. Otherwise, set α' to be the string s such that $G(s) = G(\alpha)$.
3. Set $r_{S_\perp} \leftarrow C_{LD,\kappa}(\alpha') \oplus z$.
4. If the Hamming distances satisfy $\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) \leq (p + \delta_{th}) \cdot \lambda$, then output b. Else, output $\perp$.

The output of the experiment is $(i\mathcal{O}(f_\lambda^{(3)}), \hat{r})$.

5. $H_4(1^\lambda)$: We now consider when Eve simply gets the $i\mathcal{O}$ of a null circuit. Let $r \leftarrow \{0,1\}^\lambda$ be chosen as in $\mathrm{Enc}(1^\lambda, b)$ where each bit r_i is independently uniform randomly sampled. Then let $\hat{r} \coloneqq \mathsf{ChE}(r) \in \{0,1,\perp\}^\lambda$.

$f_\lambda^{(4)}(x)$:
Input: $x \in \{0,1\}^\lambda$

1. Output $\perp$.

The output of the experiment is $(i\mathcal{O}(f_\lambda^{(4)}), \hat{r})$.

We now make the following claims:

1. $H_0(1^\lambda) \approx_c H_1(1^\lambda)$: First, $\hat{r}$ is sampled identically as $\mathsf{ChE}(r)$. Then, observe that for any subset $S_\perp \subseteq [\lambda]$, the function $f_\lambda^{(1)}(\cdot)$ is functionally equivalent to f_λ because for any string $x \in \{0,1\}^\lambda$,
$$\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) = \Delta_H(x, r).$$
Therefore, the claim follows by the indistinguishability of the $i\mathcal{O}$ scheme.

2. $H_1(1^\lambda) \approx_c H_2(1^\lambda)$: We claim that $f_\lambda^{(2)}$ is functionally equivalent to $f_\lambda^{(1)}$ with overwhelming probability over the coins used in generation of $\hat{r} \in \{0,1,\perp\}^\lambda$.

   For the functional equivalence to hold we require that on inputs $x \in \{0,1\}^\lambda$, that if $\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) \leq (p + \delta_{th}) \cdot \lambda$, then the list decoding algorithm is able to recover α. The list decoding algorithm recovers α when $\Delta_H(x_{S_\perp}, r_{S_\perp}) \leq \left(\frac{1}{2} - \varepsilon\right) \cdot \kappa$. Now viewing κ as a random variable, a sufficient condition for this implication to occur is, therefore, that $(p + \delta_{th}) \cdot \lambda \leq \left(\frac{1}{2} - \varepsilon\right) \cdot \kappa$. A standard Chernoff argument shows that κ satisfies this inequality with overwhelming probability for our choice of parameters.

   In detail, let κ_i be a 0/1 indicator random variable for the event that $\hat{r}_i = \perp$, and let $\kappa \coloneqq \sum_{i \in [\lambda]} \kappa_i$. Note that $\mathbb{E}[\kappa] = e \cdot \lambda$. By a standard additive Chernoff (Lemma 2), we have
$$\Pr\left[\kappa < \left(e - \lambda^{-0.1}\right) \cdot \lambda\right] \leq \exp\left(-2 \cdot \lambda^{0.8}\right)$$
   Recall that our objective is to show that $\left(\frac{1}{2} - \varepsilon\right) \cdot \kappa \geq (p + \delta_{th}) \cdot \lambda$ with high probability so that on inputs of small Hamming distance less that $(p + \delta_{th}) \cdot \lambda$, our new function $f_\lambda^{(2)}$

successfully recovers α via a list decoding procedure. For there to exist a setting of ε such that the following probability is overwhelming:

$$\Pr\left[\left(\frac{1}{2}-\varepsilon\right)\cdot\kappa \geq (p+\delta_{th})\cdot\lambda\right]$$

it suffices to choose a constant ε such that

$$\left(\frac{1}{2}-\varepsilon\right)^{-1}\cdot(p+\delta_{th})\cdot\lambda \leq \left(e-\lambda^{-0.1}\right)\cdot\lambda.$$

To see why, observe that if this inequality holds, then the same Chernoff above implies that

$$\Pr\left[\kappa < \left(\frac{1}{2}-\varepsilon\right)^{-1}\cdot(p+\delta_{th})\cdot\lambda\right] \leq \exp(-2\cdot\lambda^{0.8})$$

Rearranging the above inequality, we obtain an equivalent inequality:

$$\varepsilon \leq \frac{1}{2} - \frac{p\cdot\lambda^{0.1}+1}{e\cdot\lambda^{0.1}-1}.$$

Then observe that the degradation condition guarantees that $p < \frac{e}{2}$, so by choosing any constant $\varepsilon \in \left[0, \frac{1}{2}-\frac{p}{e}\right]$, the above inequality holds for sufficiently large $\lambda \in \mathbb{N}$. Therefore, we conclude that by choosing any constant $\varepsilon \in \left[0, \frac{1}{2}-\frac{p}{e}\right]$, for sufficiently large λ,

$$\Pr\left[\left(\frac{1}{2}-\varepsilon\right)\cdot\kappa > (p+\delta_{th})\cdot\lambda\right] \geq 1-\exp(-2\cdot\lambda^{0.8})$$

Conditioning on the event that $\kappa > \left(\frac{1}{2}-\varepsilon\right)^{-1}\cdot(p+\delta_{th})\cdot\lambda = \Omega(\lambda)$, we can analyze the behavior of $f_\lambda^{(2)}$:

(a) If the input x satisfies that $\Delta_H(x_{S_\perp}, r_{S_\perp}) \leq (p+\delta_{th})\cdot\lambda$, then the Hamming weight of $x_{S_\perp} \oplus r_{S_\perp}$ satisfies

$$\mathsf{wt}_H(x_{S_\perp}\oplus r_{S_\perp}) \leq (p+\delta_{th})\cdot\lambda \leq \left(\frac{1}{2}-\varepsilon\right)\cdot\kappa.$$

The $(1/2-\varepsilon, q(\kappa, 1/\varepsilon))$-list decodable property, implies that the preimage α of the randomly chosen codeword c will be recovered in the list D. By the injectivity of the PRG, there is a unique such preimage α and therefore, the function $f_\lambda^{(2)}$ can correctly compute $C_{LD,\kappa}(\alpha)$ and recover $r_{S_\perp}$. Finally, Step 3 is exactly computing $f_\lambda^{(1)}(r')$.

(b) If the input x instead satisfies the complement relation $\Delta_H(x_{S_\perp}, r_{S_\perp}) > p+\delta_{th}$, then observe that either $f_\lambda^{(2)}$ will output $\perp$ either due to Step 2, or due to Step 3. This is exactly the output behavior of $f_\lambda^{(1)}$.

Therefore, conditioning on the event that $\kappa > \left(\frac{1}{2}-\varepsilon\right)^{-1}\cdot(p+\delta_{th})\cdot\lambda = \Omega(\lambda)$, we have that $f_\lambda^{(2)}$ has the same input-output behavior as $f_\lambda^{(1)}$ on all inputs and we appeal to the indistinguishability of the $i\mathcal{O}$ scheme to show that $(i\mathcal{O}(f^{(1)}), \hat{r}) \approx_c (i\mathcal{O}(f^{(2)}), \hat{r})$. This event occurs with all but negligible probability, so the two hybrids are computationally indistinguishable: $H_1(1^\lambda) \approx_c H_2(1^\lambda)$.

3. $H_2(1^\lambda) \approx_c H_3(1^\lambda)$: We will use the computationally indistinguishability of the PRG G to show this statement. Again, we condition on the event that the number of observed erasures, κ, satisfies $\kappa > \left(\frac{1}{2} - \varepsilon\right)^{-1} \cdot (p + \delta_{th}) \cdot \lambda = \Omega(\lambda)$. This event occurs with all but negligible probability in λ where the probability is over the the coins used in the generation of $\hat{r}$.

   If there is a polynomial-time non-uniform algorithm $\mathcal{A}$ that can distinguish between $(i\mathcal{O}(f^{(2)}), \hat{r})$ and $(i\mathcal{O}(f^{(3)}), \hat{r})$, then we can construct a polynomial-time non-uniform algorithm $\mathcal{B}$ that can distinguish between the output of G on a random string of length κ^d and a uniform random string of length $3 \cdot \kappa^d$. Namely, $\mathcal{B}$ on input $R_{Challenge}$ follows the construction template of the experiment $H_2(1^\lambda)$ and uses $R_{Challenge}$ in-place of $G(\alpha)$ to obtain some output $(\hat{f}, \hat{r})$. Crucially, observe that z is independently sampled from $G(\alpha)$ because $r_{S_\perp}$ is uniform randomly sampled and $r_{S_\perp}$ is not a hardcoded constant in the function. If $R_{Challenge}$ is sampled from the distribution $G(U_{p(\lambda)})$, where $U_{p(\lambda)}$ is the uniform distribution on $p(\lambda)$ bits, then $\mathcal{B}$ has exactly sampled $(\hat{f}, \hat{r})$ from the distribution of $H_3(1^\lambda)$'s output. Otherwise, $R_{Challenge}$ is sampled from the distribution $U_{3 \cdot p(\lambda)}$, where $U_{3 \cdot p(\lambda)}$ is the uniform distribution on $3 \cdot p(\lambda)$ bits, then $\mathcal{B}$ has exactly sampled $(\hat{f}, \hat{r})$ from the distribution of $H_4(1^\lambda)$'s output. Then $\mathcal{B}$ passes $(\hat{f}, \hat{r})$ as input, as well as the appropriate advice string, to $\mathcal{A}$ who distinguishes between the two with non-negligible probability in λ. Therefore, $\mathcal{B}$ breaks the security of the PRG with non-negligible probability in λ.

4. $H_3(1^\lambda) \approx_c H_4(1^\lambda)$: Again, we condition on the event that $\kappa > \left(\frac{1}{2} - \varepsilon\right)^{-1} \cdot (p + \delta_{th}) \cdot \lambda = \Omega(\lambda)$. Observe that in $H_3(1^\lambda)$, with all but negligible probability in λ, the uniform randomly chosen string R is not in the image of G. Then with all but negligible probability in λ, $f_\lambda^{(3)}$ always outputs $\perp$ so $f_\lambda^{(3)}$ is identical to the null circuit $f_\lambda^{(4)}$ that always outputs $\perp$. Then, by the indistinguishability property of the $i\mathcal{O}$ scheme, we have that $H_3(1^\lambda) \approx_c H_4(1^\lambda)$.

This series of hybrids show that Eve's view is computationally indistinguishable from receiving a null circuit. Therefore, there cannot exist any polynomial-time non-uniform algorithm that is able to recover b efficiently from the real output of the coding scheme with non-negligible advantage.

□

4.1 Application: Codes with Easy Error Correction and Hard Erasure Correction

In any error-correcting code, correcting t erasures is (by definition) no harder than correcting t errors. But suppose we allow the error bound t to be smaller than the erasure bound v, while still insisting that erasure-decoding is information-theoretically possible. Then we have a fundamental coding-theoretic complexity question, unexplored before [12] and our work: Can we design an (efficiently encodable, binary) error-correcting code for which t errors can be corrected in polynomial time whereas correcting v erasures requires super-polynomial time?

What makes the problem challenging is the fact that most useful classes of error-correcting codes that support efficient decoding are *linear*. For linear codes, if correcting v erasures is information-theoretically possible, then it can also be done in polynomial time by solving a system of linear equations. Thus, a solution to the above question must inherently rely on efficiently decodable *nonlinear* codes, for which fewer natural examples exist.

A simple corollary of Theorem 5 gives a solution to this problem where the encoding function is *probabilistic* and the noise pattern is *random* (for both errors and erasures). This is captured by the

following theorem.

Corollary 1 (Easy-hard codes)**.** *Suppose $i\mathcal{O}$ and injective one-way functions exist. Then, for every $p, e \in (0,1)$ such that $2p < e < 4p(1-p)$, there exists a PPT encoding algorithm $E : \{0,1\}^k \to \{0,1\}^{n(k)}$ such that the following holds:*

- **Easy p-error correction.** *There is a polynomial-time decoder D and a negligible ϵ such that for all $x \in \{0,1\}^k$ we have $\Pr[D(\tilde{y}_p) \neq x] \leq \epsilon(k)$, where $\tilde{y}_p$ is obtained by first computing $y \leftarrow E(x)$ and then flipping each bit of y with probability p.*
- **Hard e-erasure correction.** *For every non-uniform polynomial-time decoder D^* there is a negligible μ such that for a uniformly random $x \in \{0,1\}^k$, $\Pr[D^*(\tilde{y}_e) = x] \leq \mu(k)$ where $\tilde{y}_e$ is obtained by first computing $y \leftarrow E(x)$ and then erasing each bit of y with probability e.*
- **Nontriviality.** *There exists a computationally unbounded decoder D^∞ and a negligible ϵ such that for a random $x \in \{0,1\}^k$ we have $\Pr[D^\infty(\tilde{y}_e) = x] \geq 1 - \epsilon(k)$, where $\tilde{y}_e$ is obtained by first computing $y \leftarrow E(x)$ and then erasing each bit of y with probability e.*

Proof. Let $(\mathrm{Enc}, \mathrm{Dec})$ be as guaranteed by Theorem 5 for the given e and p (which exist since $e > 2p$), and assume without loss of generality that the encoding length is λ^c for some positive integer c. For $x \in \{0,1\}^k$, let $E(x) = \mathrm{Enc}(1^k, x_1) \circ \cdots \circ \mathrm{Enc}(1^k, x_k)$, and let $D(\tilde{y}^1 \circ \cdots \circ \tilde{y}^k) = \mathrm{Dec}(1^k, \tilde{y}^1) \circ \cdots \circ \mathrm{Dec}(1^k, \tilde{y}^k)$ (where the length regularity assumption guarantees unique parsing).

The easiness requirement is immediate, and the hardness follows by a standard hybrid argument (in fact, hardness holds not only for a random message x but also for distinguishing between any two messages x of the same length).

Finally, nontriviality follows from the impossibility of information-theoretic wiretap coding when $e < 4p(1-p)$ [8, 15]. Indeed, if nontriviality does not hold, then an information-theoretic wiretap coding can be obtained via the amplification techniques in [8] (see also [12]) □

A natural question is whether it is possible to prove a variant of Corollary 1 in which the encoding function E is *deterministic*. Note that if we use a random oracle to determine the randomness for E based on the message, then the above proof still applies. This gives rise to a heuristic solution using a cryptographic hash function to replace the random oracle. We leave open the question of eliminating the random oracle by relying on cryptographic or derandomization assumptions.

Finally, an intriguing question is whether instances of similar "easy-hard codes" can be obtained (even heuristically) via a natural construction, without relying on the power of general-purpose obfuscation.

5 Characterization of Degraded Binary Channels

We present a novel polytope (polygon in two dimensions) formulation for DMCs that exactly characterizes when a binary input channel ChB is a degradation of a binary input channel ChE. In the case of larger constant-sized input alphabets, this characterization breaks down; nevertheless, we can show that polytope non-containment for *any* pair of channels $(\mathsf{ChB}, \mathsf{ChE})$ enables a reduction to the binary input alphabet and output alphabet setting.

Definition 14 (Channel Polytope)**.** Let $\mathbf{A}$ be a matrix of non-negative entries. We associate to $\mathbf{A}$ the following polytope, denoted $\mathcal{P}(\mathbf{A})$, which can be defined in either of the following equivalent ways:

- $\mathcal{P}(\mathbf{A})$ is the convex hull of all subset-sums of columns of $\mathbf{A}$.
- $\mathcal{P}(\mathbf{A}) = \{\mathbf{A} \cdot \mathbf{s} : \mathbf{0} \leq \mathbf{s} \leq \mathbf{1}\}$

For a channel ChA, we denote by $\mathcal{P}(\mathsf{ChA})$ the polytope $\mathcal{P}(\mathbf{A})$, where $\mathbf{A}$ is the row-stochastic matrix associated to ChA.

5.1 Characterization of Degraded Channels with Binary Input

We now characterize when a pair of channels $(\mathsf{ChB}, \mathsf{ChE})$, both with binary input alphabet, satisfies the relation that ChB is a degradation of ChE.

Theorem 6. *Let $\mathbf{B}, \mathbf{E}$ be two non-negative matrices with two rows that satisfy $\mathbf{B} \cdot \mathbf{1} = \mathbf{E} \cdot \mathbf{1}$. Then $\mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E})$ if and only if there exists a row-stochastic matrix $\mathbf{S}$ such that $\mathbf{B} = \mathbf{E} \cdot \mathbf{S}$.*

In other words, for any finite constant-sized alphabets $\mathcal{Y}, \mathcal{Z}$, and any two binary input channels $\mathsf{ChB} : \{0,1\} \to \mathcal{Y}$ and $\mathsf{ChE} : \{0,1\} \to \mathcal{Z}$ with associated matrices $\mathbf{B}$ and $\mathbf{E}$ respectively, the polygon containment $\mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E})$ exactly characterizes when ChB is a degradation of ChE.

Proof of Theorem 6. Proving one direction of Theorem 6 is straightforward: Suppose there is a row-stochastic matrix $\mathbf{S}$ such that $\mathbf{B} = \mathbf{E} \cdot \mathbf{S}$. Now consider any point $\mathbf{x} \in \mathcal{P}(\mathbf{B})$, meaning $\mathbf{x} = \mathbf{B} \cdot \mathbf{s}$ where $\mathbf{0} \leq \mathbf{s} \leq \mathbf{1}$. Then $\mathbf{x} = \mathbf{E} \cdot (\mathbf{S} \cdot \mathbf{s}) = \mathbf{E} \cdot \mathbf{s}'$ where $\mathbf{0} \leq \mathbf{s}' \leq \mathbf{1}$ since $\mathbf{S}$ is stochastic.

The converse direction, namely showing that if $\mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E})$, then there exists a stochastic $\mathbf{S}$ such that $\mathbf{B} = \mathbf{E} \cdot \mathbf{S}$, will proceed by induction on the number of columns of $\mathbf{B}$. In the base case that $\mathbf{B}$ has a single column $\mathbf{v}$, we note that $\mathbf{v} = \mathbf{E} \cdot \mathbf{1}$ since the column sums of $\mathbf{B}, \mathbf{E}$ are assumed to be the same. Therefore, we can let $\mathbf{S} = \mathbf{1}$ and then $\mathbf{B} = \mathbf{v} = \mathbf{E} \cdot \mathbf{S}$.

To prove the inductive step, let $\mathbf{v}$ be the first column of $\mathbf{B}$ and $\mathbf{B}'$ be the result of deleting the first column, so that $\mathbf{B} = \left[\mathbf{v} \mid \mathbf{B}'\right]$. Define the shifted polygon $\mathcal{P}_{\mathbf{v}}(\mathbf{E}) = \mathcal{P}(\mathbf{E}) - \mathbf{v} = \{\mathbf{u} - \mathbf{v} : \mathbf{u} \in \mathcal{P}(\mathbf{E})\}$. We will utilize the following lemma:

Lemma 6. *For any matrix $\mathbf{E}$ of non-negative entries, and any $\mathbf{v} \in \mathcal{P}(\mathbf{E})$, there exists a* diagonal *matrix $\mathbf{D}$, $\mathbf{0} \leq \mathbf{D} \leq \mathbf{1}$, such that $\mathcal{P}(\mathbf{E}) \cap \mathcal{P}_{\mathbf{v}}(\mathbf{E}) = \mathcal{P}(\mathbf{E} \cdot \mathbf{D})$.*

Using Lemma 6, we can finish the inductive proof. Since $\mathbf{v} \in \mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E})$, $\mathbf{v}$ satisfies the conditions in Lemma 6. Then let $\mathbf{D}$ be the non-negative diagonal matrix guaranteed by Lemma 6. Now, note that $\mathcal{P}(\mathbf{B}') \subseteq \mathcal{P}(\mathbf{B}) \subseteq \mathcal{P}(\mathbf{E})$. Likewise, $\mathcal{P}(\mathbf{B}') + \mathbf{v} \subseteq \mathcal{P}(\mathbf{B})$, and so $\mathcal{P}(\mathbf{B}') \subseteq \mathcal{P}_{\mathbf{v}}(\mathbf{B}) \subseteq \mathcal{P}_{\mathbf{v}}(\mathbf{E})$. Therefore, $\mathcal{P}(\mathbf{B}') \subseteq \mathcal{P}(\mathbf{E}) \cap \mathcal{P}_{\mathbf{v}}(\mathbf{E}) = \mathcal{P}(\mathbf{E} \cdot \mathbf{D})$. Moreover, recall that $\mathbf{E} \cdot \mathbf{1}$ is the maximal element on $\mathcal{P}_{\mathbf{v}}(\mathbf{E})$. Note that $(\mathbf{E} \cdot \mathbf{1}) - \mathbf{v} \in \mathcal{P}(\mathbf{E}) \cap \mathcal{P}_{\mathbf{v}}(\mathbf{E})$, and therefore $(\mathbf{E} \cdot \mathbf{1}) - \mathbf{v}$ is the maximal element in $\mathcal{P}(\mathbf{E}) \cap \mathcal{P}_{\mathbf{v}}(\mathbf{E}) = \mathcal{P}(\mathbf{E} \cdot \mathbf{D})$, meaning $\mathbf{B}' \cdot \mathbf{1} = (\mathbf{B} \cdot \mathbf{1}) - \mathbf{v} = (\mathbf{E} \cdot \mathbf{1}) - \mathbf{v} = \mathbf{E} \cdot \mathbf{D} \cdot \mathbf{1}$.

Since $\mathbf{B}' \cdot \mathbf{1} = (\mathbf{E} \cdot \mathbf{D}) \cdot \mathbf{1}$ and $\mathbf{B}'$ has one less column than $\mathbf{B}$, we can apply the inductive hypothesis: There exists a row-stochastic $\mathbf{S}'$ such that $\mathbf{B}' = (\mathbf{E} \cdot \mathbf{D}) \cdot \mathbf{S}'$. Let $\mathbf{w} = \mathbf{1} - \mathbf{D} \cdot \mathbf{1}$. Note that since the diagonal entries of $\mathbf{D}$ are in the closed interval $[0, 1]$, we have $\mathbf{0} \leq \mathbf{w} \leq \mathbf{1}$. Moreover, $\mathbf{E} \cdot \mathbf{w} = (\mathbf{E} \cdot \mathbf{1}) - (\mathbf{E} \cdot \mathbf{D} \cdot \mathbf{1}) = (\mathbf{B} \cdot \mathbf{1}) - (\mathbf{B}' \cdot \mathbf{1}) = \mathbf{v}$.

Let $\mathbf{S} = \left[\mathbf{w} \mid \mathbf{D} \cdot \mathbf{S}'\right]$. Then $\mathbf{E} \cdot \mathbf{S} = \mathbf{B}$. Moreover, $\mathbf{S}$ is non-negative, and $\mathbf{S} \cdot \mathbf{1} = \mathbf{w} + (\mathbf{D} \cdot \mathbf{S}' \cdot \mathbf{1}) = \mathbf{w} + \mathbf{D} \cdot \mathbf{1} = \mathbf{1}$. Therefore, $\mathbf{S}$ is row-stochastic, as desired. This completes the proof of Theorem 6, assuming Lemma 6. □

Proof of Lemma 6 For the remainder of the subsection, we focus on proving Lemma 6. To prove this lemma, we will use a geometric viewpoint to understand the intersection of a polygon and its shifted copy. We will define an "enclosing path" which encloses a solid.

Definition 15 (Solid of an Enclosing Path)**.** For a finite ordered set of points $P = (p_1, \ldots, p_m) \subseteq [0,1] \times [0,1]$ such that $p_m = p_1$, the solid $\mathcal{S}(P) \subseteq [0,1] \times [0,1]$ is defined to be the closed polygon whose facets are the directed line segments $\{\overline{p_i p_{i+1}}\}$ for $i \in \{1, \ldots, m-1\}$ and whose interior is given by the counterclockwise orientation with respect to each directed line segment.

For any non-negative matrix $\mathbf{E} \in \mathbb{R}^{2\times n}_{\geq 0}$, we can now describe $\mathcal{P}(\mathbf{E})$ by an enclosing path. Let $\mathbf{m}_i$ is the ith column of $\mathbf{E}$ and assume without loss of generality that columns $\mathbf{m}_i$ are sorted in monotonically increasing order of their polar coordinate angles. That is, $\mathbf{m}_i$ can be written as (r_i, θ_i) for $r_i \geq 0$ and $\theta_i \in [0, \pi/2]$, and we sort the columns such that $\theta_{i+1} \geq \theta_i$. If there are ties, $\theta_i = \theta_{i+1}$, we decide on an arbitrary order for the colliding columns.

Lemma 7. *For any matrix $\mathbf{E} \in \mathbb{R}^{2\times n}_{\geq 0}$ whose columns are sorted in monotonically increasing order of their polar coordinate angles,*

$$\mathcal{P}(\mathbf{E}) = \mathcal{S}\left((\mathbf{0}, \mathbf{m}_1, \mathbf{m}_1 + \mathbf{m}_2, \ldots, \sum_{i=1}^{n-1} \mathbf{m}_i, \sum_{i=1}^{n} \mathbf{m}_i, \sum_{i=2}^{n} \mathbf{m}_i, \ldots, \mathbf{m}_{n-1} + \mathbf{m}_n, \mathbf{m}_n, \mathbf{0})\right)$$

Proof. We proceed by induction on the number of columns n. The proof is effectively a proof by picture.

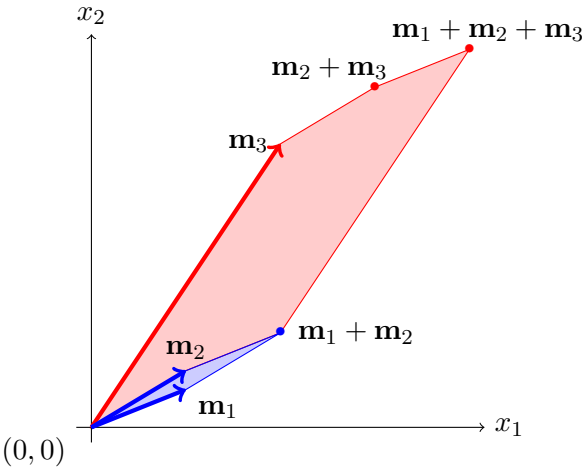

Figure 8: We visually depict the induction step. The vectors $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ are sorted by their angle in a monotonically increasing order. The induction hypothesis is given by the polygon outlined in blue. The shaded red region represents the addition of $\alpha \cdot \mathbf{m}_3$ for $\alpha \in [0,1]$. The induction step is completed by following the path $(\mathbf{0}, \mathbf{m}_1, \sum_{i=1}^{2} \mathbf{m}_i, , \sum_{i=1}^{3} \mathbf{m}_i, \sum_{i=2}^{3} \mathbf{m}_i, \mathbf{m}_3, \mathbf{0})$.

In the base case, $n = 1$ and $\mathbf{E} = \mathbf{e}_1 \in \mathbb{R}^2_{\geq 0}$ is a column and the polygon is given by the points of the form $\alpha \cdot \mathbf{m}_1$, for scalar $\alpha \in [0,1]$, so we have $\mathcal{P}(\mathbf{E}) = \mathcal{S}((\mathbf{0}, \mathbf{m}_1, \mathbf{0}))$.

Now consider a matrix $\mathbf{E} \in [0,1]^{2\times n+1}$ whose columns are sorted in monotonically increasing order of their polar coordinate angles. Define the matrix $\mathbf{E}' \in [0,1]^{2\times n}$ so that

$$\mathbf{E}' \coloneqq \left[\mathbf{m}_1 \;\middle|\; \cdots \;\middle|\; \mathbf{m}_n\right]$$
$$\mathbf{E} \coloneqq \left[\mathbf{E}' \;\middle|\; \mathbf{m}_{n+1}\right].$$

Then the inductive hypothesis gives the following characterization:

$$\mathcal{P}(\mathbf{E}') = \mathcal{S}\left((\mathbf{0}, \mathbf{m}_1, \mathbf{m}_1 + \mathbf{m}_2, \ldots, \sum_{i=1}^{n-1} \mathbf{m}_i, \sum_{i=1}^{n} \mathbf{m}_i, \sum_{i=2}^{n} \mathbf{m}_i, \ldots, \mathbf{m}_{n-1} + \mathbf{m}_n, \mathbf{m}_n, \mathbf{0})\right).$$

Moreover, we have

$$\mathcal{P}(\mathbf{E}) = \mathcal{P}(\mathbf{E}') + \{\alpha \cdot \mathbf{m}_{n+1} : \alpha \in [0,1]\}.$$

Since $\mathbf{m}_{n+1}$ is of the largest angle $\theta_{n+1} \in [0, \pi/2]$, $\theta_{n+1} \geq \theta_i$ for $i \in [n]$, we have

$$\begin{aligned}
&\mathcal{P}(\mathbf{E}') + \{\alpha \cdot \mathbf{m}_{n+1} : \alpha \in [0,1]\} \\
&\quad = \mathcal{S}\left((\mathbf{0}, \mathbf{m}_1, \mathbf{m}_1 + \mathbf{m}_2, \ldots, \sum_{i=1}^{n} \mathbf{m}_i, \sum_{i=1}^{n+1} \mathbf{m}_i, \sum_{i=2}^{n+1} \mathbf{m}_i, \ldots, \mathbf{m}_n + \mathbf{m}_{n+1}, \mathbf{m}_{n+1}, \mathbf{0})\right).
\end{aligned}$$

For a visual depiction of this step, see Figure 8. This step concludes the proof as we have established the desired characterization of $\mathcal{P}(\mathbf{E})$. □

Having established a characterization of $\mathcal{P}(\mathbf{E})$ in terms of an enclosing path, we can now complete the proof of Lemma 6. Using the above characterization, we define the "forward path" of $\mathcal{P}(\mathbf{E})$ to be

$$\left(0, \mathbf{m}_1, \mathbf{m_1} + \mathbf{m}_2, \ldots, \sum_{i=1}^{n} \mathbf{m}_i\right)$$

and the "returning path" of $\mathcal{P}(\mathbf{E})$ to be

$$\left(\sum_{i=1}^{n} \mathbf{m}_i, \sum_{i=2}^{n} \mathbf{m}_i, \ldots, \mathbf{m}_{n-1} + \mathbf{m}_n, \mathbf{m}_n, \mathbf{0}\right).$$

For any $\mathbf{v} \in \mathcal{P}(\mathbf{E})$, the shifted polygon $\mathcal{P}_v(\mathbf{E})$ satisfies the translation

$$\begin{aligned}
\mathcal{P}_v(\mathbf{E}) = \mathcal{S}\Big((\mathbf{0} - \mathbf{v}, \mathbf{m}_1 - \mathbf{v}, \mathbf{m}_1 + \mathbf{m}_2 - \mathbf{v}, \ldots, \sum_{i=1}^{n-1} \mathbf{m}_i, \\
\sum_{i=1}^{n} \mathbf{m}_i - \mathbf{v}, \sum_{i=2}^{n} \mathbf{m}_i - \mathbf{v}, \ldots, \mathbf{m}_{n-1} + \mathbf{m}_n - \mathbf{v}, \mathbf{m}_n - \mathbf{v}, \mathbf{0} - \mathbf{v})\Big).
\end{aligned}$$

To characterize the forward path of the intersection $\mathcal{P}(\mathbf{E}) \cap \mathcal{P}_{\mathbf{v}}(\mathbf{E})$, we make the following observation. The forward path of the intersection $\mathcal{P}(\mathbf{E}) \cap \mathcal{P}_{\mathbf{v}}(\mathbf{E})$ starts by following the forward path of $\mathcal{P}(\mathbf{E})$ and intersects the forward path of $\mathcal{P}_{\mathbf{v}}(\mathbf{E})$ at some point or at an overlapping line segment. Exactly one such intersection must occur because the maximal element of $\mathcal{P}_{\mathbf{v}}(\mathbf{E})$, given by $(\sum_{i=1}^{n} \mathbf{m}_i) - \mathbf{v}$, lies in $\mathcal{P}(\mathbf{E})$. The reason why this maximal element lies in $\mathcal{P}(\mathbf{E})$ is because our polytope formulation satisfies the symmetry condition that $\mathbf{v} \in \mathcal{P}(\mathbf{E})$ if and only if $(\sum_{i=1}^{n} \mathbf{m}_i) - \mathbf{v} \in \mathcal{P}(\mathbf{E})$ (express $\mathbf{v} = \mathbf{E} \cdot \mathbf{u}$ for some vector $\mathbf{u} \in [0,1]^n$ and express $\sum_{i=1}^{n} \mathbf{m}_i = \mathbf{E} \cdot \mathbf{1}$). Therefore, there exists some

index $i^* \in [n]$ and index $j^* > i^*$, such that the forward path of $\mathcal{P}(\mathbf{E}) \cap \mathcal{P}_{\mathbf{v}}(\mathbf{E})$ is given by

$$\Bigg(\mathbf{0}, \mathbf{m}_1, \mathbf{m}_1 + \mathbf{m}_2, \ldots, \sum_{i=1}^{i^*-1} \mathbf{m}_i, \qquad \text{(initial forward path of } \mathcal{P}(\mathbf{E})\text{)}$$

$$\alpha_{i^*} \cdot \mathbf{m}_{i^*} + \sum_{i=1}^{i^*-1} \mathbf{m}_i, \qquad \text{(the two forward paths meet)}$$

$$\alpha_{i^*} \cdot \mathbf{m}_{i^*} + \alpha_{j^*} \mathbf{m}_{j^*} + \sum_{i=1}^{i^*-1} \mathbf{m}_i \qquad \text{(begin on the forward path of } \mathcal{P}_{\mathbf{v}}(\mathbf{E})\text{)}$$

$$\alpha_{i^*} \cdot \mathbf{m}_{i^*} + \alpha_{j^*} \mathbf{m}_{j^*} + \sum_{i=1}^{i^*-1} \mathbf{m}_i + \mathbf{m}_{j^*+1}, \ldots, \qquad \text{(continue on the forward path of } \mathcal{P}_{\mathbf{v}}(\mathbf{E})\text{)}$$

$$\alpha_{i^*} \cdot \mathbf{m}_{i^*} + \alpha_{j^*} \mathbf{m}_{j^*} + \sum_{i=1}^{i^*-1} \mathbf{m}_i + \sum_{i=j^*+1}^{n} \mathbf{m}_i \Bigg)$$

for some scalars $\alpha_{i^*} \in [0,1]$ and $\alpha_{j^*} \in [0,1]$. Note that the returning path is given by symmetry. Then the solid formed by the forward and returning path exactly describe the polygon $\mathcal{P}(\mathbf{E} \cdot \mathbf{D})$ where the matrix $\mathbf{E} \cdot \mathbf{D} \in [0,1]^{2 \times n}$ has columns given by

$$\mathbf{E} \cdot \mathbf{D} = \left[\mathbf{m}_1 \;\middle|\; \cdots \;\middle|\; \mathbf{m}_{i^*-1} \;\middle|\; \alpha_{i^*} \cdot \mathbf{m}_{i^*} \;\middle|\; \mathbf{0} \;\middle|\; \cdots \;\middle|\; \mathbf{0} \;\middle|\; \alpha_{j^*} \cdot \mathbf{m}_{j^*} \;\middle|\; \mathbf{m}_{j^*+1} \;\middle|\; \cdots \;\middle|\; \mathbf{m}_n \right].$$

Therefore, we have a diagonal matrix

$$\mathbf{D} = \mathsf{diag}\left(\underbrace{1, \ldots, 1}_{i^*-1 \text{ times}}, \alpha_{i^*}, \underbrace{0, \ldots, 0}_{j^*-i^*-1 \text{ times}}, \alpha_{j^*}, \underbrace{1, \ldots, 1}_{n-j^* \text{ times}} \right) \in [0,1]^{n \times n}$$

such that $\mathcal{P}(\mathbf{E}) \cap \mathcal{P}_{\mathbf{v}}(\mathbf{E}) = \mathcal{P}(\mathbf{E} \cdot \mathbf{D})$. This completes the proof of Lemma 6.

5.2 Counterexample for Ternary Input Alphabets

We now show that polytope containment does not imply degradation when the input alphabet $\mathcal{X}$ is ternary. We construct two row-stochastic matrices $\mathbf{B}^* \in [0,1]^{3 \times 3}$ and $\mathbf{E}^* \in [0,1]^{3 \times 4}$ such that $\mathcal{P}(\mathbf{B}^*) \subseteq \mathcal{P}(\mathbf{E}^*)$ yet there does not exist any row-stochastic matrix $\mathbf{S} \in [0,1]^{4 \times 3}$ such that $\mathbf{B}^* = \mathbf{E}^* \cdot \mathbf{S}$.

Consider any set of four column vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4 \in [0,1]^{3 \times 1}$ that satisfy the following three conditions:

1. $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are linearly independent.
2. $\sum_{i=1}^{4} \mathbf{v}_i = \mathbf{1}$.
3. $\mathbf{v}_4 = \mathbf{v}_1/5 + \mathbf{v}_2/5 + \mathbf{v}_3/5$.

As a concrete example, consider $\mathbf{v}_1 = \begin{bmatrix} 5/6 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 5/6 \\ 0 \end{bmatrix}$, $\mathbf{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 5/6 \end{bmatrix}$. Then define the column vectors

$$\begin{aligned} \mathbf{u}_1 &\coloneqq \frac{3}{5} \cdot \mathbf{v}_1 + \frac{1}{5} \cdot \mathbf{v}_2 \\ \mathbf{u}_2 &\coloneqq \frac{3}{5} \cdot \mathbf{v}_1 + \frac{1}{5} \cdot \mathbf{v}_3 \\ \mathbf{u}_3 &\coloneqq \mathbf{v}_2 + \mathbf{v}_3 \end{aligned}$$

Define the matrix $\mathbf{E}^*$ to be

$$\mathbf{E}^* \coloneqq \left[\mathbf{v}_1 \mid \mathbf{v}_2 \mid \mathbf{v}_3 \mid \mathbf{v}_4 \right]$$

and define the matrix $\mathbf{B}^*$ to be

$$\mathbf{B}^* \coloneqq \left[\mathbf{u}_1 \mid \mathbf{u}_2 \mid \mathbf{u}_3 \right]$$

Lemma 8. $\mathcal{P}(\mathbf{B}^*) \subseteq \mathcal{P}(\mathbf{E}^*)$.

Proof. It suffices to show that all the extreme points of $\mathcal{P}(\mathbf{B}^*)$, given by the 0/1-combinations of the columns of $\mathbf{B}^*$, are in $\mathcal{P}(\mathbf{E}^*)$. Note that $\mathbf{0} \in \mathcal{P}(\mathbf{E}^*)$ by definition of the polytope formulation. Then the columns themselves, $\mathbf{u}_i$, are in $\mathcal{P}(\mathbf{E}^*)$ by the construction of $\mathbf{u}_i$. Then we have the following equivalences for the remaining 0/1-combinations:

$$\begin{aligned} \mathbf{u}_1 + \mathbf{u}_2 &= \mathbf{v}_1 + \mathbf{v}_4 \\ \mathbf{u}_2 + \mathbf{u}_3 &= \frac{2}{5} \cdot \mathbf{v}_1 + \frac{4}{5} \cdot \mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_4 \\ \mathbf{u}_1 + \mathbf{u}_3 &= \frac{2}{5} \cdot \mathbf{v}_1 + \mathbf{v}_2 + \frac{4}{5}\mathbf{v}_3 + \mathbf{v}_4 \\ \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{u}_3 &= \mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_4 = \mathbf{1}. \end{aligned}$$

Since all the extreme points are of $\mathcal{P}(\mathbf{B}^*)$ are contained in $\mathcal{P}(\mathbf{E}^*)$, the convexity of $\mathcal{P}(\mathbf{B}^*)$ and $\mathcal{P}(\mathbf{E}^*)$ implies that $\mathcal{P}(\mathbf{B}^*) \subseteq \mathcal{P}(\mathbf{E}^*)$. □

Lemma 9. *The does not exist a row-stochastic matrix* $\mathbf{S} \in [0,1]^{4\times 3}$ *such that* $\mathbf{B}^* = \mathbf{E}^* \cdot \mathbf{S}$.

Proof. Let $\mathbf{S}$ be any matrix with real entries such that $\mathbf{B}^* = \mathbf{E}^* \cdot \mathbf{S}$. Let $\mathbf{S} \coloneqq \left[\mathbf{s}_1 \mid \mathbf{s}_2 \mid \mathbf{s}_3 \right]$ where $\mathbf{s}_i$ denotes the ith column of $\mathbf{S}$. Then we have $\mathbf{u}_1 = \mathbf{E}^* \cdot \mathbf{s}_1$, $\mathbf{u}_2 = \mathbf{E}^* \cdot \mathbf{s}_2$, $\mathbf{u}_3 = \mathbf{E}^* \cdot \mathbf{s}_3$. By the linear independence of the vectors $\{\mathbf{v}_i\}_{i\in[3]}$, the only matrix $\mathbf{S}$ that satisfies the above relations is one in which $\mathbf{s}_1 = \begin{bmatrix} 3/5 \\ 1/5 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{s}_2 = \begin{bmatrix} 3/5 \\ 0 \\ 1/5 \\ 0 \end{bmatrix}$ and $\mathbf{s}_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$. But observe that this matrix $\mathbf{S}$ is not row-stochastic.

□

Therefore, we have shown that polytope containment does not imply stochastic channel degradation for channels of ternary alphabet.

6 Computational Wiretap Coding for Asymmetric BSC/BEC

In this section, we generalize the results of Section 4 from the symmetric BSC/BEC case to the asymmetric case.

Our characterization of when two binary input channels are degraded gives a clean formula for when a binary asymmetric channel is not a degradation of a binary asymmetric erasure channel. We will use the formula to obtain a computational wiretap coding scheme from indistinguishability obfuscation for wiretap channels of the form $(\mathsf{BAC}_{p_0,p_1}, \mathsf{BAEC}_{e_0,e_1})$ for choices of p_0, p_1, e_0, e_1 that satisfy the following formula.

Lemma 10. *Channel* BAC_{p_0,p_1} *is not a degradation of channel* BAEC_{e_0,e_1} *if and only if* $e_0e_1 > p_1e_0 + p_0e_1$.

Proof. By Theorem 6, BAC_{p_0,p_1} is not a degradation of BAEC_{e_0,e_1} if and only if $\mathcal{P}(\mathsf{BAC}_{p_0,p_1}) \not\subseteq \mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$. Therefore it suffices to characterize for what values of e_0, e_1, p_0, p_1 we have that $\mathcal{P}(\mathsf{BAC}_{p_0,p_1}) \not\subseteq \mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$. These two polygons sit inside the unit square $[0,1]^2$ and are easy to visualize.

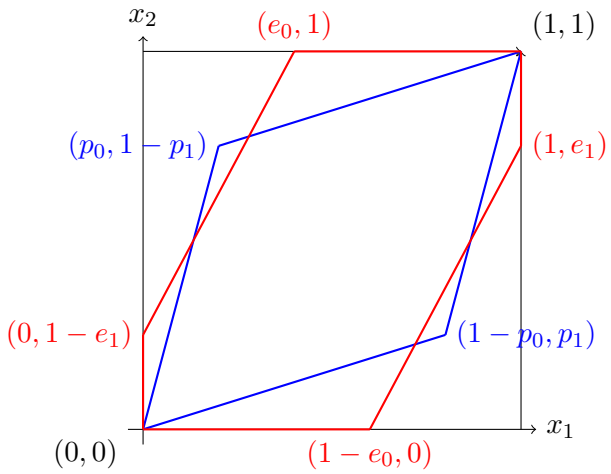

Figure 9: An example of polygon non-containment for binary asymmetric channels and binary asymmetric erasure channels. Here, x_1 and x_2 are indeterminates. The blue polygon is $\mathcal{P}(\mathsf{BAC}_{p_0,p_1})$ for parameters $p_0 = 1/5, p_1 = 1/4$. The red polygon is $\mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$ for parameters $e_0 = 2/5$, $e_1 = 3/4$.

By convexity, if $\mathcal{P}(\mathsf{BAC}_{p_0,p_1}) \not\subseteq \mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$, then there exists an extreme point of $\mathcal{P}(\mathsf{BAC}_{p_0,p_1})$ that does not belong to $\mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$. There are only four extreme points of $\mathcal{P}(\mathsf{BAC}_{p_0,p_1})$: $(0,0), (1,1), (1-p_0, p_1), (p_0, 1-p_1)$. Moreover, both $\mathcal{P}(\mathsf{BAC}_{p_0,p_1})$ and $\mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$ satisfy two-fold symmetry, namely a point $\mathbf{v}$ is in the polygon if and only if $\mathbf{1} - \mathbf{v}$ is in the polygon. This symmetry tells us that $(1-p_0, p_1) \notin \mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$ if and only if $(p_0, 1-p_1) \notin \mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$. We have shown, then, it suffices to characterize when $(p_0, 1-p_1) \notin \mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$.

Assume without loss of generality that $p_0 \leq \frac{1}{2}$ (see Remark 1). Then, by the slope-intercept formula, we see that $(p_0, 1-p_1) \notin \mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$ if and only if $p_0 < e_0$ and $e_0e_1 > p_1e_0 + p_0e_1$. Since $p_1 \in [0,1]$, the condition $e_0e_1 > p_1e_0 + p_0e_1$ implies that $p_0 < e_0$. Then, removing this redundant condition gives the statement that $(p_0, 1-p_1) \notin \mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$ if and only if $e_0e_1 > p_1e_0 + p_0e_1$. Then, we conclude that $\mathcal{P}(\mathsf{BAC}_{p_0,p_1}) \not\subseteq \mathcal{P}(\mathsf{BAEC}_{e_0,e_1})$ if and only if $e_0e_1 > p_1e_0 + p_0e_1$.

□

Theorem 7. *Assuming the existence of $i\mathcal{O}$ and injective PRGs, there exists a computational wiretap coding scheme for any wiretap channel $(\mathsf{BAC}_{p_0,p_1}, \mathsf{BAEC}_{e_0,e_1})$ such that $e_0e_1 > e_0p_1 + e_1p_0$.*

In addition to $i\mathcal{O}$ and injective PRGs for the main construction, we will use any efficient error correcting code for BAC_{p_0,p_1}, given by $\mathcal{C}_B = (\mathcal{C}_B.\mathrm{Enc}, \mathcal{C}_B.\mathrm{Dec})$ such that for all $x \in \{0,1\}^*$,

$$\Pr[\mathcal{C}_B.\mathrm{Dec}(1^\lambda, \mathsf{BAC}_{p_0,p_1}(\mathcal{C}_B.\mathrm{Enc}(1^\lambda, x))) = x] \geq 1 - \varepsilon(\lambda)$$

for a negligible ε. As an example of a concrete instantiation, one can consider a simple repetition code [1] where the encoding of a bit b is given by $\mathcal{C}.\mathrm{Enc}(1^\lambda, b) = b^\lambda = \underbrace{bb\ldots b}_{\lambda \text{ times}}$ and the encoding of a string is done bit-by-bit. The rate of this code is $1/\lambda$.

Coding Scheme 2. (**Computational Wiretap Coding Scheme for** $(\mathsf{BAC}_{p_0,p_1}, \mathsf{BAEC}_{e_0,e_1})$)

We now describe our wiretap encoder-decoder pair $(\mathrm{Enc}, \mathrm{Dec})$ that depends on both BAC_{p_0,p_1} and BAEC_{e_0,e_1}. The encoder takes a bit $b \in \{0,1\}$ and we only note that the construction readily extends to taking strings as input.

$\mathrm{Enc}(1^\lambda, b)$:
Constants: p_0, p_1, e_0, e_1.

1. Let $\eta = \frac{e_1}{e_0+e_1}$, $t = \frac{e_0p_1+e_1p_0}{e_0+e_1}$, $\delta_{th} = \lambda^{-0.1}$.
2. Sample $r \leftarrow \{0,1\}^\lambda$ according to the following Bernoulli distribution with parameter $1-\eta$: For each $i \in [\lambda]$, sample
$$r_i \leftarrow \begin{cases} 0 & \text{with probability } \eta \\ 1 & \text{with probability } 1-\eta \end{cases}$$
3. Construct the function $f : \{0,1\}^\lambda \to \{m, \perp\}$ as follows

> $f_\lambda(x)$:
> **Input**: $x \in \{0,1\}^\lambda$.
> **Hardwired constants**: r, b, e_0, e_1, p_0, p_1.
>
> 1. If the Hamming distance $\Delta_H(x, r) < (t + \delta_{th}) \cdot \lambda$, then output b. Else, output $\perp$.

[1] We briefly explain why a simple repetition code suffices. The repetition code allows recovery of the bit b as long as the output distribution $\mathsf{BAC}_{p_0,p_1}(0)$ "sufficiently" differs from the output distribution $\mathsf{BAC}_{p_0,p_1}(1)$, that is, $p_0 \neq 1 - p_1$ (otherwise the channel completely randomizes the input). In this setting, to recover the original bit b, the decoding algorithm counts the number of 0's in the output string and for sufficiently large $\lambda \in \mathbb{N}$, the number of 0's in $\mathsf{BAC}_{p_0,p_1}(0^\lambda)$ and $\mathsf{BAC}_{p_0,p_1}(1^\lambda)$ and a standard Chernoff argument guarantees that these two distributions are concentrated far from each other.

4. Output $(\mathcal{C}_B.\mathrm{Enc}(i\mathcal{O}(f_\lambda)), r)$ where f_λ (viewed as a circuit) is padded to be the maximum circuit size of the circuits in $\{f_\lambda\} \cup \left\{f_\lambda^{(i)}\right\}_{i\in[4]}$ where the $f_\lambda^{(i)}$ are defined later in the security proof.

$\mathrm{Dec}(1^\lambda, \hat{f}, z)$:

1. Let $f \leftarrow \mathcal{C}_B.\mathrm{Dec}(\hat{f})$.
2. Output $f(z)$.

Intuition for the Coding Scheme Since ChB is not a degradation of ChE, by Lemma 10 the channel parameters satisfy the relation $e_0e_1 > p_0e_1 + p_1e_0$. Then, observe that the expected relative Hamming distance of Bob's received string with respect to the initially sent string is

$$\eta p_0 + (1-\eta)p_1 = \frac{p_0e_1 + p_1e_0}{e_0 + e_1}.$$

In the case of Eve's received string, we observe that any erasure position is equally likely, from Eve's perspective, to have been either a 0 or 1. This is because the encoding algorithm samples every bit r_i from a Bernoulli distribution with parameter $1-\eta$ where $\eta = \frac{e_1}{e_0+e_1}$. Then Eve's best guessing strategy to guess the initial string r sampled by the encoding algorithm is to uniform randomly guess each bit r_i for which the ith position of Eve's received string is an erasure. The expected fraction (of the total string length λ) of erasures that Eve receives is

$$\eta e_0 + (1-\eta)e_1 = \frac{2e_0e_1}{e_0 + e_1}.$$

Since Eve's best guess of r is to randomly guess for each of these erasures, the expected relative Hamming distance of Eve's best guess with respect to r is half of the expected fraction of erasures:

$$\frac{e_0e_1}{e_0 + e_1}.$$

Consider the constant d_{diff} where we define $d_{diff} := \frac{e_0e_1 - (p_0e_1 + p_1e_0)}{e_0+e_1}$. Then observe that the channel degradation condition exactly states that $d_{diff} > 0$ is some positive constant. The threshold parameter δ_{th} is set to be $\lambda^{-0.1}$. Under this parameter setting, a standard Chernoff argument ensures that with overwhelming probability Bob's received string $\hat{r}_B \in \{0,1\}^\lambda$ will satisfy the statistical check that $\Delta_H(\hat{r}_B, r) \leq (t + \delta_{th}) \cdot \lambda$. Intuitively, Eve's best guess will fail the statistical check, and we will formally show this by a series of computationally indistinguishable hybrids (experiments) to prove that no polynomial-time non-uniform algorithm can guess a random message bit b with non-negligible advantage using Eve's channel outputs (her view). We give the formal details below.

Lemma 11 (Correctness of the Computational Wiretap Encoding Scheme)**.** *There exists a negligible function $\mu : \mathbb{N} \to [0,1]$ such that for every message bit $b \in \{0,1\}$,*

$$\Pr[\mathrm{Dec}(1^\lambda, \mathsf{ChB}(\mathrm{Enc}(1^\lambda, b))) = b] \geq 1 - \epsilon(\lambda)$$

Proof. We will use (f, r) to denote random variables representing the output of $\mathrm{Enc}(1^\lambda, b)$ and we will use $(\tilde{f}, \tilde{r})$ to denote random variables representing the output of the channel $\mathsf{ChB}(f, r)$. Using the notation $\Delta_H(\cdot,\cdot)$ to denote Hamming distance (non-relative), we note that the expected value of the Hamming distance of $\tilde{r}$ from r is given as $\mathbb{E}_{\mathsf{ChB},\mathrm{Enc}}\left[\Delta_H(\tilde{r}, r)\right] = \frac{e_0 p_1 + e_1 p_0}{e_0 + e_1} \cdot \lambda = t \cdot \lambda$ where the t stated here is exactly as was defined in Coding Scheme 2. Then the probability over the channel randomness and the coins used by the encoding algorithm $\mathrm{Enc}(\cdot,\cdot)$ that the received string $\tilde{r}$ fails the statistical check is given by an additive Chernoff (Lemma 2) bound:

$$\Pr_{\mathsf{ChB},\mathrm{Enc}}\left[\Delta_H(\tilde{r}, r) \geq (t + \delta_{th}) \cdot \lambda\right] \leq \exp\left(-2 \cdot \delta_{th}^2 \cdot \lambda\right) = \exp\left(-2 \cdot \lambda^{0.8}\right)$$

which is negligible in λ.

□

Lemma 12 (Security of the Computational Wiretap Encoding Scheme)**.** *Let* ChB *be a* BAC_{p_0,p_1} *channel and let* ChE *be a* BAEC_{e_0,e_1} *channel such that* $e_0 e_1 > e_0 p_1 + e_1 p_0$*. For all polynomial-time non-uniform algorithms* $\mathcal{A}$*, there exists a negligible function* $\mu : \mathbb{N} \to [0, 1]$ *such that*

$$\Pr[\mathcal{A}(1^\lambda, \mathsf{ChE}(\mathrm{Enc}(1^\lambda, b))) = b] \leq \frac{1}{2} + \mu(\lambda)$$

where b *is uniformly distributed over* $\{0, 1\}$*.*

Proof. We will proceed through the following series of hybrids (experiments) that models Eve's view. We will show that what Eve receives from this encoding process is computationally indistinguishable from Eve receiving a null circuit, thereby rendering Eve unable to recover the message bit b except with negligible advantage. In each of the following hybrids, each function (viewed as a circuit) is padded to be the maximum circuit size of the circuits in $\{f_\lambda\} \cup \left\{f_\lambda^{(i)}\right\}_{i\in[4]}$ where f_λ is defined in the construction and $f_\lambda^{(i)}$ are defined in each of the below hybrids.

1. $H_0(1^\lambda)$: In the real world, Alice sends $\mathrm{Enc}(1^\lambda, b) = (\mathcal{C}.\mathrm{Enc}(i\mathcal{O}(f)), r)$ through ChE and Eve receives the output of the channel, $\mathsf{ChE}((\mathcal{C}.\mathrm{Enc}(i\mathcal{O}(f))), r)$. We assume that Eve successfully recovers $i\mathcal{O}(f)$, since such an assumption only gives Eve more information. The output of the experiment is $(i\mathcal{O}(f), \mathsf{ChE}(r))$.

2. $H_1(1^\lambda)$: In this hybrid, we consider a slight variation of the above experiment. Let $r \leftarrow \{0,1\}^\lambda$ be chosen as in $\mathrm{Enc}(1^\lambda, b)$ where each bit r_i is independently identically sampled from a Bernoulli distribution with parameter $1 - \eta$. Then let $\hat{r} := \mathsf{ChE}(r) \in \{0, 1, \perp\}^\lambda$. Let $S_\perp \subseteq [\lambda]$ be the set of indices for which $\hat{r}_i = \perp$ and let $\overline{S_\perp} := [\lambda] \setminus S_\perp$. Let $\kappa := |S_\perp|$. Define the finite subsequence (a string) $r_{S_\perp} := (r_{i_j})_{i_j \in S_\perp \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^\kappa$ consisting of the bits from the indices from $S_\perp$ and analogous finite subsequence $r_{\overline{S_\perp}} := (r_{i_j})_{i_j \in \overline{S_\perp} \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^{\lambda-\kappa}$ of the bits from the indices from $\overline{S_\perp}$.

   We now give an alternate encoding method where instead of constructing the function f as in Coding Scheme 2, Alice instead uses the following function $f^{(1)}$:

   $f_\lambda^{(1)}(x)$:
   Input: $x \in \{0,1\}^\lambda$
   Hardwired constants: $r_{S_\perp}, r_{\overline{S_\perp}}, b, e_0, e_1, p_0, p_1, S_\perp$.

1. If the Hamming distances satisfy $\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) \leq (t + \delta_{th}) \cdot \lambda$, then output b. Else, output $\perp$.

The output of the experiment is $(i\mathcal{O}(f_\lambda^{(1)}), \hat{r})$.

3. $H_2(1^\lambda)$: Let $r \leftarrow \{0,1\}^\lambda$ be chosen as in $\mathrm{Enc}(1^\lambda, b)$ where each bit r_i is independently identically sampled from a Bernoulli distribution with parameter $1-\eta$. Then let $\hat{r} \coloneqq \mathsf{ChE}(r) \in \{0,1,\perp\}^\lambda$. Let $S_\perp \subseteq [\lambda]$ be the set of indices for which $\hat{r}_i = \perp$ and let $\overline{S_\perp} \coloneqq [\lambda] \setminus S_\perp$. Let $\kappa = |S_\perp|$. Define the finite subsequence (a string) $r_{S_\perp} \coloneqq (r_{i_j})_{i_j \in S_\perp \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^\kappa$ consisting of the bits from the indices from $S_\perp$ and analogous finite subsequence $r_{\overline{S_\perp}} \coloneqq (r_{i_j})_{i_j \in \overline{S_\perp} \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^{\lambda-\kappa}$ of the bits from the indices from $\overline{S_\perp}$.

We now give an alternate encoding method where instead of constructing the function $f^{(1)}$ as in $H_1(1^\lambda)$, Alice will do the following in order to construct a different function $f^{(2)}$ which we will define shortly:

(a) Let $\varepsilon = \frac{1}{4} - \frac{t(e_0+e_1)}{4e_0e_1}$. Let $C_{LD,\kappa} : \{0,1\}^{\kappa^d} \to \{0,1\}^\kappa$ be a code from a $(1/2 - \varepsilon, q(\kappa, 1/\varepsilon))$ list-decodable ensemble of binary codes for some constant $0 < d < 1$ and some polynomial $q(\kappa, 1/\varepsilon)$. We will use $C_{LD,\kappa}.\mathsf{ListDec}(\cdot)$ to denote an efficient list-decoding function for $C_{LD,\kappa}$.

(b) Sample $\alpha \in \{0,1\}^{\kappa^d}$ uniform randomly and set $c \leftarrow C_{LD,\kappa}(\alpha)$ so $c \in \{0,1\}^\kappa$.

(c) Let $z = c \oplus r_{S_\perp}$.

(d) Let $G : \{0,1\}^{\kappa^d} \to \{0,1\}^{3 \cdot \kappa^d}$ be a length-tripling injective PRG.

$f_\lambda^{(2)}(x)$:
Input: $x \in \{0,1\}^\lambda$
Hardwired constants: $r_{\overline{S_\perp}}, z, G(\alpha), b, e_0, e_1, p_0, p_1, S_\perp$.

1. Let $D \leftarrow C_{LD,\kappa}.\mathsf{ListDec}(z \oplus x_{S_\perp})$. D is a list of at most $q(\kappa, 1/\varepsilon)$ many elements in $\{0,1\}^{\kappa^d}$.
2. If $G(s) \neq G(\alpha)$ for all strings $s \in D$, output $\perp$. Otherwise, set α' to be the string s such that $G(s) = G(\alpha)$.
3. Set $r_{S_\perp} \leftarrow C_{LD,\kappa}(\alpha') \oplus z$.
4. If the Hamming distances satisfy $\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) \leq (t + \delta_{th}) \cdot \lambda$, then output b. Else, output $\perp$.

The output of the experiment is $(i\mathcal{O}(f_\lambda^{(2)}), \hat{r})$.

4. $H_3(1^\lambda)$: Let $r \leftarrow \{0,1\}^\lambda$ be chosen as in $\mathrm{Enc}(1^\lambda, b)$ where each bit r_i is independently identically sampled from a Bernoulli distribution with parameter $1-\eta$. Then let $\hat{r} \coloneqq \mathsf{ChE}(r) \in \{0,1,\perp\}^\lambda$. Let $S_\perp \subseteq [\lambda]$ be the set of indices for which $\hat{r}_i = \perp$ and let $\overline{S_\perp} \coloneqq [\lambda] \setminus S_\perp$. Let $\kappa \coloneqq |S_\perp|$. Define the finite subsequence (a string) $r_{S_\perp} \coloneqq (r_{i_j})_{i_j \in S_\perp \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^\kappa$ consisting of the bits from the indices from $S_\perp$ and analogous finite subsequence $r_{\overline{S_\perp}} \coloneqq (r_{i_j})_{i_j \in \overline{S_\perp} \text{ s.t. } i_j < i_{j+1}} \in \{0,1\}^{\lambda-\kappa}$ of the bits from the indices from $\overline{S_\perp}$.

We now give an alternate encoding method where instead of constructing the function $f^{(1)}$ as in $H_1(1^\lambda)$, Alice will do the following in order to construct a different function $f^{(2)}$ which we will define shortly:

(a) Let $\varepsilon = \frac{1}{4} - \frac{t(e_0+e_1)}{4e_0e_1}$. Let $C_{LD,\kappa} : \{0,1\}^{\kappa^d} \to \{0,1\}^\kappa$ be a code from a $(1/2-\varepsilon, q(\kappa, 1/\varepsilon))$ list-decodable ensemble of binary codes for some constant $0 < d < 1$ and polynomial $q(\kappa, 1/\varepsilon)$. We will use $C_{LD,\kappa}.\mathsf{ListDec}(\cdot)$ to denote an efficient list-decoding function for $C_{LD,\kappa}$.

(b) Sample $\alpha \in \{0,1\}^{\kappa^d}$ uniform randomly and set $c \leftarrow C_{LD,\kappa}(\alpha)$ so $c \in \{0,1\}^\kappa$.

(c) Let $z \leftarrow c \oplus r_{S_\perp}$.

(d) Let $G : \{0,1\}^{\kappa^d} \to \{0,1\}^{3\cdot\kappa^d}$ be a length-tripling injective PRG.

(e) Let R be a string sampled uniform randomly from $\{0,1\}^{3\cdot\kappa^d}$.

$f^{(3)}_\lambda(x)$:
Input: $x \in \{0,1\}^\lambda$
Hardwired constants: $r_{\overline{S_\perp}}, z, R, m, e_0, e_1, p_0, p_1, S_\perp$.

1. Let $D \leftarrow C_{LD,\kappa}.\mathsf{ListDec}(z \oplus x_{S_\perp})$. D is a list of at most $q(\kappa, 1/\varepsilon)$ many elements in $\{0,1\}^{\kappa^d}$.
2. If $G(s) \neq R$ for all strings $s \in D$, output $\perp$. Otherwise, set α' to be the string s such that $G(s) = G(\alpha)$.
3. Set $r_{S_\perp} \leftarrow C_{LD,\kappa}(\alpha') \oplus z$.
4. If the Hamming distances satisfy $\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) \leq (t + \delta_{th}) \cdot \lambda$, then output m. Else, output $\perp$.

The output of the experiment is $(i\mathcal{O}(f^{(3)}_\lambda), \hat{r})$.

5. $H_4(1^\lambda)$: We now consider when Eve simply gets the $i\mathcal{O}$ of a null circuit. Let $r \leftarrow \{0,1\}^\lambda$ be chosen as in $\mathrm{Enc}(1^\lambda, m, p_0, p_1, e_0, e_1)$ where each bit r_i is independently identically sampled from a Bernoulli distribution with parameter $1-\eta$. Then let $\hat{r} \coloneqq \mathsf{ChE}(r) \in \{0,1,\perp\}^\lambda$.

$f^{(4)}_\lambda(x)$:
Input: $x \in \{0,1\}^\lambda$

1. Output $\perp$.

The output of the experiment is $(i\mathcal{O}(f^{(4)}_\lambda), \hat{r})$.

We now make the following claims:

1. $H_0(1^\lambda) \approx_c H_1(1^\lambda)$: First, $\hat{r}$ is sampled identically as $\mathsf{ChE}(r)$. Then, observe that for any subset $S_\perp \subseteq [\lambda]$, the function $f^{(1)}_\lambda(\cdot)$ is functionally equivalent to f_λ because for any string $x \in \{0,1\}^\lambda$,
$$\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) = \Delta_H(x, r).$$
Therefore, the claim follows by the indistinguishability of the $i\mathcal{O}$ scheme.

2. $H_1(1^\lambda) \approx_c H_2(1^\lambda)$: We claim that $f_\lambda^{(2)}$ is functionally equivalent to $f_\lambda^{(1)}$ with overwhelming probability over the coins used in generation of $\hat{r} \in \{0,1,\perp\}^\lambda$.

   For the functional equivalence to hold we require that on inputs $x \in \{0,1\}^\lambda$, that if $\Delta_H(x_{S_\perp}, r_{S_\perp}) + \Delta_H(x_{\overline{S_\perp}}, r_{\overline{S_\perp}}) \leq (t + \delta_{th}) \cdot \lambda$, then the list decoding algorithm is able to recover α. The list decoding algorithm recovers α when $\Delta_H(x_{S_\perp}, r_{S_\perp}) \leq \left(\frac{1}{2} - \varepsilon\right) \cdot \kappa$. A sufficient condition for this implication to occur is, therefore, that $(t + \delta_{th}) \cdot \lambda \leq \left(\frac{1}{2} - \varepsilon\right) \cdot \kappa$. A standard Chernoff argument shows that κ satisfies this inequality with overwhelming probability for our choice of parameters.

   In detail, let κ_i be a 0/1 indicator random variable for the event that $\hat{r}_i = \perp$, and let $\kappa \coloneqq \sum_{i \in [\lambda]} \kappa_i$. Note that $\mathbb{E}[\kappa] = \left(\frac{2e_0e_1}{e_0+e_1}\right) \cdot \lambda$. By a standard additive Chernoff (Lemma 2), we have

$$\Pr\left[\kappa < \left(\frac{2e_0e_1}{e_0+e_1} - \lambda^{-0.1}\right) \cdot \lambda\right] \leq \exp\left(-2 \cdot \lambda^{0.8}\right)$$

   Recall that our objective is to show that $\left(\frac{1}{2} - \varepsilon\right) \cdot \kappa \geq (t + \delta_{th}) \cdot \lambda$ with high probability so that on inputs of small Hamming distance less that $(t + \delta_{th}) \cdot \lambda$, our new function $f_\lambda^{(2)}$ successfully recovers α via a list decoding procedure. For there to exist a setting of ε such that the following probability is overwhelming:

$$\Pr\left[\left(\frac{1}{2} - \varepsilon\right) \cdot \kappa \geq (t + \delta_{th}) \cdot \lambda\right]$$

   it suffices to choose a constant ε such that

$$\left(\frac{1}{2} - \varepsilon\right)^{-1} \cdot (t + \delta_{th}) \cdot \lambda \leq \left(\frac{2e_0e_1}{e_0+e_1} - \lambda^{-0.1}\right) \cdot \lambda. \qquad (\star)$$

   To see why, observe that if this inequality holds, then the same Chernoff above implies that

$$\Pr\left[\kappa < \left(\frac{1}{2} - \varepsilon\right)^{-1} \cdot (t + \delta_{th}) \cdot \lambda\right] \leq \exp(-2 \cdot \lambda^{0.8})$$

   Rearranging the inequality $(\star)$, we obtain an equivalent inequality:

$$\varepsilon \leq \frac{1}{2} - \frac{t \cdot \lambda^{0.1} + 1}{\gamma \cdot \lambda^{0.1} - 1}$$

   where $\gamma = \frac{2e_0e_1}{e_0+e_1} = \frac{\mathbb{E}[\kappa]}{\lambda}$. Then observe that the degradation condition guarantees that $t < \frac{\gamma}{2}$, so by choosing any constant $\varepsilon \in \left[0, \frac{1}{2} - \frac{t}{\gamma}\right]$, the above inequality holds for sufficiently large $\lambda \in \mathbb{N}$. Therefore, we conclude that by choosing any constant $\varepsilon \in \left[0, \frac{1}{2} - \frac{t}{\gamma}\right]$, for sufficiently large λ,

$$\Pr\left[\left(\frac{1}{2} - \varepsilon\right) \cdot Y > (t + \delta_{th}) \cdot \lambda\right] \geq 1 - \exp(-2 \cdot \lambda^{0.8})$$

   Conditioning on the event that $\kappa > \left(\frac{1}{2} - \varepsilon\right)^{-1} \cdot (t + \delta_{th}) \cdot \lambda = \Omega(\lambda)$, we can analyze the behavior of $f_\lambda^{(2)}$:

(a) If the input x satisfies that $\Delta_H(x_{S_\perp}, r_{S_\perp}) \leq (t + \delta_{th}) \cdot \lambda$, then the Hamming weight of $x_{S_\perp} \oplus r_{S_\perp}$ satisfies

$$\mathsf{wt}_H(x_{S_\perp} \oplus r_{S_\perp}) \leq (t + \delta_{th}) \cdot \lambda \leq \left(\frac{1}{2} - \varepsilon\right) \cdot \kappa.$$

The $(1/2 - \varepsilon, q(\kappa, 1/\varepsilon))$-list decodable property, implies that the preimage α of the randomly chosen codeword c will be recovered in the list D. By the injectivity of the PRG, there is a unique such preimage α and therefore, the function $f_\lambda^{(2)}$ can correctly compute $C_{LD,\kappa}(\alpha)$ and recover $r_{S_\perp}$. Finally, Step 3 is exactly computing $f_\lambda^{(1)}(r')$.

(b) If the input x instead satisfies the complement relation $\Delta_H(x_{S_\perp}, r_{S_\perp}) > t + \delta_{th}$, then observe that either $f_\lambda^{(2)}$ will output $\perp$ either due to Step 2, or due to Step 3. This is exactly the output behavior of $f_\lambda^{(1)}$.

Therefore, conditioning on the event that $\kappa > \left(\frac{1}{2} - \varepsilon\right)^{-1} \cdot (t + \delta_{th}) \cdot \lambda = \Omega(\lambda)$, we have that $f_\lambda^{(2)}$ has the same input-output behavior as $f_\lambda^{(1)}$ on all inputs and we appeal to the indistinguishability of the $i\mathcal{O}$ scheme to show that $(i\mathcal{O}(f_\lambda^{(1)}), \hat{r}) \approx_c (i\mathcal{O}(f_\lambda^{(2)}), \hat{r})$. This event occurs with all but negligible probability, so the two hybrids are computationally indistinguishable: $H_1(1^\lambda) \approx_c H_2(1^\lambda)$.

3. $H_2(1^\lambda) \approx_c H_3(1^\lambda)$: We will use the computationally indistinguishability of the PRG G to show this statement. Again, we condition on the event that the number of observed erasures, κ, satisfies $\kappa > \left(\frac{1}{2} - \varepsilon\right)^{-1} \cdot (t + \delta_{th}) \cdot \lambda = \Omega(\lambda)$. This event occurs with all but negligible probability in λ where the probability is over the the coins used in the generation of $\hat{r}$.

If there is a polynomial-time non-uniform adversary $\mathcal{A}$ who can distinguish between $(i\mathcal{O}(f_\lambda^{(2)}), \hat{r})$ and $(i\mathcal{O}(f_\lambda^{(3)}), \hat{r})$, then we can construct a polynomial-time non-uniform adversary $\mathcal{B}$ who can distinguish between the output of G on a random string of length κ^d and a uniform random string of length $3 \cdot \kappa^d$. Namely, $\mathcal{B}$ on input $R_{Challenge}$ follows the construction template of the experiment $H_2(1^\lambda)$ and uses $R_{Challenge}$ in-place of $G(\alpha)$ to obtain some output $(\hat{f}, \hat{r})$.

Crucially, observe that z is independently sampled from $G(\alpha)$. This is because $r_{S_\perp}$, from the viewpoint of the adversary, is uniform randomly sampled, since for all $i \in S_\perp$,

$$\Pr\left[r_i = 0 \mid \hat{r}_i = \perp\right] = \frac{1}{2}.$$

Therefore, z, from the viewpoint of the adversary is distributed uniform randomly over $\{0,1\}^\kappa$ because of the above observation and the fact that $r_{S_\perp}$ is not hardwired anywhere in the function.

If $R_{Challenge}$ is sampled from the distribution $G(U_{p(\lambda)})$, where $U_{p(\lambda)}$ is the uniform distribution on $p(\lambda)$ bits, then $\mathcal{B}$ has exactly sampled $(\hat{f}, \hat{r})$ from the distribution of $H_3(1^\lambda)$'s output. Otherwise, $R_{Challenge}$ is sampled from the distribution $U_{3 \cdot p(\lambda)}$, where $U_{3 \cdot p(\lambda)}$ is the uniform distribution on $3 \cdot p(\lambda)$ bits, then $\mathcal{B}$ has exactly sampled $(\hat{f}, \hat{r})$ from the distribution of $H_4(1^\lambda)$'s output. Then $\mathcal{B}$ passes $(\hat{f}, \hat{r})$ as input, as well as the appropriate advice string, to $\mathcal{A}$ who distinguishes between the two with non-negligible probability in λ. Therefore, $\mathcal{B}$ breaks the security of the PRG with non-negligible probability in λ.

4. $H_3(1^\lambda) \approx_c H_4(1^\lambda)$: Again, we condition on the event that $\kappa > \left(\frac{1}{2} - \varepsilon\right)^{-1} \cdot (t + \delta_{th}) \cdot \lambda = \Omega(\lambda)$. Observe that in $H_3(1^\lambda)$, with all but negligible probability in λ, the uniform randomly chosen string R is not in the image of G. Then with all but negligible probability in λ, $f_\lambda^{(3)}$ always outputs $\perp$ so $f_\lambda^{(3)}$ is identical to the null circuit $f_\lambda^{(4)}$ that always outputs $\perp$. Then, by the indistinguishability property of the $i\mathcal{O}$ scheme, we have that $H_3(1^\lambda) \approx_c H_4(1^\lambda)$.

This series of hybrids show that Eve's view is computationally indistinguishable from receiving a null circuit. Therefore, there cannot exist any polynomial-time non-uniform adversary that is able to recover b efficiently from the real output of the coding scheme with non-negligible advantage. □

6.1 Reducing the General Binary Input Case to the BAC/BAEC Case

Lemma 13. *Let* $\mathsf{ChB} : \{0,1\} \to \mathcal{Y}$ *be a channel with a binary input alphabet. For any channel* $\mathsf{ChE} : \{0,1\} \to \mathcal{Z}$ *with a binary input alphabet, if* ChB *is not a degradation of* ChE*, then there exists a channel* $\mathsf{ChPost} : \mathcal{Y} \to \{0,1\}$ *such that* $\mathsf{ChPost} \circ \mathsf{ChB}$ *is not a degradation of* ChE*.*

Proof. For any channel $\mathsf{ChE} : \{0,1\} \to \mathcal{Z}$ for an arbitrary constant-sized output alphabet $\mathcal{Z}$, if ChB is not a degradation of ChE then $\mathcal{P}(\mathbf{B}) \not\subseteq \mathcal{P}(\mathbf{E})$ by Theorem 6. Therefore, there exists some vector $\mathbf{u} \in [0,1]^{|\mathcal{Y}| \times 1}$ such that $\mathbf{B} \cdot \mathbf{u} \in \mathcal{P}(\mathbf{B}) \setminus \mathcal{P}(\mathbf{E})$. Observe that we can in fact assume that $\mathbf{u}$ is a 0/1 vector in $\{0,1\}^{|\mathcal{Y}| \times 1}$ by taking an extreme point of $\mathcal{P}(\mathbf{B})$ that is not contained in $\mathcal{P}(\mathbf{E})$ (such a point exists by the convexity of $\mathcal{P}(\mathbf{B})$). Then observe that $\mathbf{B} \cdot (\mathbf{1} - \mathbf{u}) \in \mathcal{P}(\mathbf{B}) \setminus \mathcal{P}(\mathbf{E})$ by the aforementioned symmetry of the polytope formulation. The row-stochastic matrix $\mathbf{P} \coloneqq \left[\mathbf{u} \;\middle|\; \mathbf{1} - \mathbf{u}\right]$ defines a channel ChPost, and $\mathbf{P}$ is such that

$$\mathcal{P}(\mathbf{B} \cdot \mathbf{P}) \not\subseteq \mathcal{P}(\mathbf{E})$$

which implies (by Theorem 6) that $\mathsf{ChPost} \circ \mathsf{ChB}$ is not a degradation of ChE. □

Remark 2. Note that computing the row-stochastic matrix representation of ChPost, given ChB and ChE described as row-stochastic matrices $\mathbf{B} \in [0,1]^{2 \times |\mathcal{Y}|}$ and $\mathbf{E} \in [0,1]^{2 \times |\mathcal{Z}|}$, can be efficiently done. The enclosing path formulation of $\mathcal{P}(\mathbf{B})$ tells us that the extreme points of $\mathcal{P}(\mathbf{B})$ are of the form $\mathbf{B} \cdot (1, 1, \ldots, 1, 0, \ldots, 0)^\top$ or $\mathbf{B} \cdot (0, \ldots, 0, 1, \ldots, 1)^\top$ since the points in the enclosing path formulation are of the form $\sum_{i=1}^{t} \mathbf{b}_i$ and $\sum_{i=t}^{|\mathcal{Y}|} \mathbf{b}_i$ for $t \in [|\mathcal{Y}|]$ and where $\mathbf{b}_i$ denotes the ith column of $\mathbf{B}$. Then, computing a single extreme point of $\mathcal{P}(\mathbf{B})$ that is not in $\mathcal{P}(\mathbf{E})$ can be done by iterating through all (a constant number at most $2 \cdot |\mathcal{Y}|$) of extreme points of $\mathcal{P}(\mathbf{B})$ and using standard linear programming to check if each point is in $\mathcal{P}(\mathbf{E})$. Upon finding a single extreme point $\mathbf{B} \cdot \mathbf{v}$ not in $\mathcal{P}(\mathbf{E})$ for a vector $\mathbf{v} \coloneqq (1, 1, \ldots, 1, 0, \ldots, 0)^\top$, the vector $\mathbf{v}$ defines one column of the row-stochastic matrix formulation of ChPost and the second column is obtained by taking $\mathbf{1} - \mathbf{v}$ (due to two-fold symmetry).

Lemma 14. *Let* $\mathsf{ChB} : \{0,1\} \to \{0,1\}$ *be a channel with binary input and output alphabet. For any channel* $\mathsf{ChE} : \{0,1\} \to \mathcal{Z}$ *with a binary input alphabet, if* ChB *is not a degradation of* ChE*, then there exists a channel* ChE' *that is a binary asymmetric erasure channel with some parameters* e_0, e_1 *such that* ChE *is a degradation of* ChE' *and* ChB *is not a degradation of* ChE*.*

Remark 3. Computing the row-stochastic matrix of ChE' can be done efficiently. As in the case of Remark 2, we can iterate through all extreme points of $\mathcal{P}(\mathbf{B})$, and upon finding an extreme point of $\mathcal{P}(\mathbf{B})$, one can efficiently compute a strict separating hyperplane by linear programming. The other facet is obtained by applying two-fold symmetry and the enclosing path for ChE' consists of just 6 points readily obtained as described in the proof.

Proof of Lemma 14. The proof is essentially a proof by picture. If ChB is not a degradation of ChE, then $\mathcal{P}(\mathbf{B}) \not\subseteq \mathcal{P}(\mathbf{E})$. Therefore, there exists an extreme point of $\mathcal{P}(\mathbf{B})$, of the form $\mathbf{B}\cdot\mathbf{u} \in \mathcal{P}(\mathbf{B})\setminus\mathcal{P}(\mathbf{E})$ where $\mathbf{u}$ is a 0/1 vector in $\{0,1\}^{2\times 1}$, that is not contained in $\mathcal{P}(\mathbf{E})$. By the strict separating hyperplane theorem (Theorem 4), there exists $\psi \in \mathbb{R}^2$ such that $\langle \mathbf{u}, \psi\rangle > \sup\langle \mathcal{P}(\mathbf{E}), \psi\rangle$. Geometrically, ψ is a line L that strictly separates $\mathcal{P}(\mathbf{E})$ and $\mathbf{u}$ into two different half-planes. This line L forms a new facet of a polygon, which we will define to be $\mathcal{P}(\mathbf{E}')$, that contains $\mathcal{P}(\mathbf{E})$.

We will use a x-y plane terminology for $\mathbb{R}^2$ for indeterminates x and y. Assume without loss of generality that $\mathbf{B}\cdot\mathbf{u}$ is below the line $y = x$ (we can assume this because otherwise we can instead consider $\mathbf{B}\cdot(\mathbf{1}-\mathbf{u}) \in \mathcal{P}(\mathbf{B}) \setminus \mathcal{P}(\mathbf{E})$). The line L must intersect the line defined by $x = 1$ at some point $(1, b)$ for some scalar $0 < b < 1$ by the *strict* separation property since $\mathbf{1} \coloneqq (1, 1) \in \mathcal{P}(\mathbf{E})$. Similarly the line L must intersect the line defined by $y = 0$ at some point $(1 - a, 0)$ for some scalar $0 < a < 1$ such that $1 - a > 0$ by the *strict* separation property since $\mathbf{0} \coloneqq (0, 0) \in \mathcal{P}(\mathbf{E})$. Therefore, the line L is given by the equation

$$y = \frac{b}{a}\cdot x - \frac{b(1-a)}{a}.$$

Then observe that the aforementioned two-fold symmetry gives us another separating hyperplane (a line) L' that separates another extreme point $\mathbf{B}\cdot(\mathbf{1}-\mathbf{u}) \in \mathcal{P}(\mathbf{B}) \setminus \mathcal{P}(\mathbf{E})$ from $\mathcal{P}(\mathbf{E})$. The equation for this new separating line L' is given by the two-fold symmetry, for which we simply replace y and x with $1 - y$ and $1 - x$ respectively in the above equation for L to obtain the equation $y = \frac{b}{a}x + (1 - b)$ for line L'. These lines L and L' now give us an enclosing path P for a convex solid that contains $\mathcal{P}(\mathbf{E})$ where $P = (\mathbf{0}, (1 - a, 0), (1, b), \mathbf{1}, (a, 1), (0, 1 - b), \mathbf{0})$ which is exactly the polygon of the following row-stochastic matrix:

$$\mathbf{E}' \coloneqq \begin{bmatrix} 1-a & 0 & a \\ 0 & 1-b & b \end{bmatrix}.$$

This row-stochastic matrix $\mathbf{E}'$ exactly describes a $\mathsf{BAEC}_{a,b}$ channel. Therefore, we have constructed a channel $\mathsf{ChE}' = \mathsf{BAEC}_{a,b}$ for which $\mathcal{P}(\mathbf{E}) \subseteq \mathcal{P}(\mathbf{E}')$ and for which $\mathcal{P}(\mathbf{B}) \not\subseteq \mathcal{P}(\mathbf{E}')$ (since $\mathbf{B}\cdot\mathbf{u}$ remains separated by construction). By Theorem 6, we have that these two polygon conditions imply that ChE is a degradation of ChE', whose matrix representation is $\mathbf{E}'$, and ChB is not a degradation of ChE. □

Theorem 8. *Assuming the existence of $i\mathcal{O}$ and injective PRGs, there exists a computational wiretap coding scheme for any pair of binary input channels* $(\mathsf{ChB}, \mathsf{ChE})$ *such that* ChB *is not a degradation of* ChE*,*

Using Lemma 13 and Lemma 14, from any pair of binary input channels $(\mathsf{ChB}, \mathsf{ChE})$ such that ChB is not a degradation of ChE, we can efficiently compute a pair of channels $(\mathsf{ChB}', \mathsf{ChE}')$ such that $\mathsf{ChB}' = \mathsf{BAC}_{p_0,p_1}$ and $\mathsf{ChE}' = \mathsf{BAEC}_{e_0,e_1}$ for some constant parameters p_0, p_1, e_0, e_1 such that the following hold:

1. $e_0 e_1 > e_0 p_1 + e_1 p_0$, (that is, ChB' is not a degradation of ChE').
2. ChB simulates ChB'. That is, $\mathcal{P}(\mathbf{B}') \subseteq \mathcal{P}(\mathbf{B})$ where $\mathbf{B}$ and $\mathbf{B}'$ are the matrix representations of the channels $\mathsf{ChE}, \mathsf{ChE}'$ respectively. In other words, Bob, who has the physical channel ChB, can apply a post-processing step (apply ChPost from Lemma 13) on his received word to exactly simulate receiving an output from ChB'.
3. ChE' simulates ChE. That is, $\mathcal{P}(\mathbf{E}) \subseteq \mathcal{P}(\mathbf{E}')$ where $\mathbf{E}$ and $\mathbf{E}'$ are the matrix representations of the channels $\mathsf{ChE}, \mathsf{ChE}'$ respectively. In other words, for security we consider a channel ChE' which can simulate the received string Eve obtains through her physical channel ChE.

Therefore, to obtain a computational wiretap coding scheme $\mathcal{C}$ for $(\mathsf{ChB}, \mathsf{ChE})$, we can use a computational wiretap coding scheme $\mathcal{C}'$ for the wiretap channel $(\mathsf{ChB}', \mathsf{ChE}')$ and require that Bob post-processes his received string from his channel ChB before applying the decoding algorithm $\mathcal{C}'.\mathrm{Dec}(\cdot, \cdot)$.

- The correctness of this computational wiretap coding scheme follows by observing that for any input Alice sends through ChB, Bob can perfectly simulate having received an output from ChB' corresponding to that input, so the correctness of wiretap coding scheme $\mathcal{C}'$ guarantees that Bob is able to recover the message with all but negligible success.
- The security of this computational wiretap coding scheme follows by observing that the coding scheme $\mathcal{C}'$ gives security against the class of all polynomial-time non-uniform algorithms with a channel ChE' between themselves and Alice. This class of efficient adversaries, by the degradation condition, include those who simulate having a channel ChE between Alice and themselves. So all polynomial-time non-uniform algorithms with a channel ChE between themselves and Alice fall into this class of efficient algorithms.

Following the above outline, we now construct our computational wiretap coding scheme for the general binary input channel case.

Coding Scheme 3. (**Computational Wiretap Coding Scheme for General Binary Input Channels** $(\mathsf{ChB}, \mathsf{ChE})$)

We now describe our wiretap encoder-decoder pair $(\mathrm{Enc}, \mathrm{Dec})$ that depends on both ChB and ChE, for which $\mathbf{B}$ and $\mathbf{E}$ are their respective row-stochastic matrix representations. The encoder takes a bit $b \in \{0, 1\}$ and we only note that the construction readily extends to taking strings as input.

To construct our encoder-decoder pair, we first compute $\mathbf{B}' = \mathbf{B} \cdot \mathbf{P}$ (for a matrix $\mathbf{P} \in [0, 1]^{|Y| \times 2}$ as obtained in Remark 2) and a matrix $\mathbf{E}'$ (see Remark 3) such that $\mathsf{ChB}' = \mathsf{BAC}_{p_0, p_1}$ and $\mathsf{ChE}' = \mathsf{BAEC}_{e_0, e_1}$ such that

- $e_0 e_1 > e_0 p_1 + e_1 p_0$.
- $\mathcal{P}(\mathbf{B}') \subseteq \mathcal{P}(\mathbf{B})$.
- $\mathcal{P}(\mathbf{E}) \subseteq \mathcal{P}(\mathbf{E}')$.

Then, let $\mathcal{C}' = (\mathrm{Enc}', \mathrm{Dec}')$ denote a computational wiretap coding scheme for the wiretap

channel $(\mathsf{ChB}', \mathsf{ChE}')$ (see Theorem 7).

$\mathrm{Enc}(1^\lambda, b)$:

1. Output $f, r \leftarrow \mathrm{Enc}'(1^\lambda, b)$.

$\mathrm{Dec}(1^\lambda, \hat{f}, \hat{r})$:

1. Perfectly simulate running the inputs through the channel ChPost (defined by row-stochastic matrix $\mathbf{P}$) to obtain $\hat{\hat{f}}, \hat{\hat{r}} \leftarrow \mathsf{ChPost}(\hat{f}, \hat{r})$.
2. Output $\mathrm{Dec}'(1^\lambda, \hat{\hat{f}}, \hat{\hat{r}})$.

Proof. The correctness of the computational wiretap coding scheme $(\mathrm{Enc}, \mathrm{Dec})$ follows immediately from that of $(\mathrm{Enc}', \mathrm{Dec}')$. In particular, if $f, r \leftarrow \mathrm{Enc}'(1^\lambda, b)$, then Bob first observes the output $\mathsf{ChB}(f, r)$ and then efficiently and perfectly simulates running ChPost (the matrix entries are assumed to be rational probabilities and the matrix description is some finite constant size) to obtain $(\mathsf{ChPost} \circ \mathsf{ChB})(f, r) = \mathsf{ChB}'(f, r)$. Then the correctness of $(\mathrm{Enc}', \mathrm{Dec}')$ implies that there exists a negligible function $\mu : \mathbb{N} \to [0, 1]$ such that for every message bit $b \in \{0, 1\}$,

$$\Pr[\mathrm{Dec}'(1^\lambda, \mathsf{ChB}'(\mathrm{Enc}'(1^\lambda, b))) = b] \geq 1 - \epsilon(\lambda).$$

This concludes the proof of correctness for $(\mathrm{Enc}, \mathrm{Dec})$.

Security of $(\mathrm{Enc}, \mathrm{Dec})$ follows from the security of $(\mathrm{Enc}', \mathrm{Dec}')$. If there is a polynomial-time non-uniform algorithm $\mathcal{A}$ such that

$$\Pr[\mathcal{A}(1^\lambda, \mathsf{ChE}(\mathrm{Enc}(1^\lambda, b))) = b] \geq \frac{1}{2} + \mu(\lambda)$$

for some non-negligible function μ, where b is uniformly distributed over $\{0, 1\}$, then we can construct a polynomial-time non-uniform algorithm $\mathcal{B}$ such that

$$\Pr[\mathcal{B}(1^\lambda, \mathsf{ChE}'(\mathrm{Enc}'(1^\lambda, b))) = b] \geq \frac{1}{2} + \mu(\lambda)$$

for some non-negligible function μ. $\mathcal{B}$, given input $\mathsf{ChE}'(\mathrm{Enc}'(1^\lambda, b))$ for a uniform random bit b, can produce $\mathsf{ChE}(\mathrm{Enc}(1^\lambda, b))$ since ChE is perfectly simulatable using ChE' (this is efficiently done since all channel descriptions are constant sized and this is feasible due to the degradation condition that ChE is a degradation of ChE'). Then $\mathcal{B}$ runs $\mathcal{A}$ on $\mathsf{ChE}(\mathrm{Enc}'(1^\lambda, b))$ and therefore has non-negligible distinguishing advantage μ, resulting in a contradiction to the security of $(\mathrm{Enc}', \mathrm{Dec}')$. This concludes the proof of security for $(\mathrm{Enc}, \mathrm{Dec})$.

□

Acknowledgments. Y. Ishai was supported in part by ERC Project NTSC (742754), BSF grant 2018393, ISF grant 2774/20, and a Google Faculty Research Award. A. Jain is supported in part by the Google Research Scholar Award and through various gifts from CYLAB, CMU. A. Sahai was supported in part from a Simons Investigator Award, DARPA SIEVE award, NTT Research, BSF grant 2018393, a Xerox Faculty Research Award, a Google Faculty Research Award, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through Award HR00112020024.

References

[1] Shweta Agrawal, Yuval Ishai, Eyal Kushilevitz, Varun Narayanan, Manoj Prabhakaran, Vinod M. Prabhakaran, and Alon Rosen. Secure computation from one-way noisy communication, or: Anti-correlation via anti-concentration. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 124–154. Springer, 2021.

[2] Michael Alekhnovich. More on average case vs approximation complexity. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 298–307. IEEE Computer Society, 2003.

[3] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Obfuscation for evasive functions. In Yehuda Lindell, editor, *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, volume 8349 of *Lecture Notes in Computer Science*, pages 26–51. Springer, 2014.

[4] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

[5] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 278–291. Springer, Heidelberg, August 1994.

[6] Andrej Bogdanov and Youming Qiao. On the security of goldreich's one-way function. *Comput. Complex.*, 21(1):83–127, 2012.

[7] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[8] Imre Csiszár and Janos Korner. Broadcast channels with confidential messages. *IEEE transactions on information theory*, 24(3):339–348, 1978.

[9] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016.

[10] Venkatesan Guruswami. List decoding of binary codes-a brief survey of some recent results. In Yeow Meng Chee, Chao Li, San Ling, Huaxiong Wang, and Chaoping Xing, editors, *Coding and Cryptology, Second International Workshop, IWCC 2009, Zhangjiajie, China, June 1-5, 2009. Proceedings*, volume 5557 of *Lecture Notes in Computer Science*, pages 97–106. Springer, 2009.

[11] Venkatesan Guruswami and Madhu Sudan. List decoding algorithms for certain concatenated codes. In *32nd ACM STOC*, pages 181–190. ACM Press, May 2000.

[12] Yuval Ishai, Alexis Korb, Paul Lou, and Amit Sahai. Beyond the Csiszár-Körner bound: Best-possible wiretap coding via obfuscation. In *Crypto 2022*, 2022.

[13] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 60–73, 2021.

[14] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb {F}_p$, dlin, and prgs in nc^0. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part I*, volume 13275 of *Lecture Notes in Computer Science*, pages 670–699. Springer, 2022.

[15] Chandra Nair. Capacity regions of two new classes of two-receiver broadcast channels. *IEEE Transactions on Information Theory*, 56(9):4207–4214, 2010.

[16] H. Vincent Poor and Rafael F. Schaefer. Wireless physical layer security. *Proceedings of the National Academy of Sciences*, 114(1):19–26, 2017.

[17] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *SIAM J. Comput.*, 50(3):857–908, 2021.

[18] Madhu Sudan. List decoding: Algorithms and applications. In Jan van Leeuwen, Osamu Watanabe, Masami Hagiya, Peter D. Mosses, and Takayasu Ito, editors, *Theoretical Computer Science, Exploring New Frontiers of Theoretical Informatics, International Conference IFIP TCS 2000, Sendai, Japan, August 17-19, 2000, Proceedings*, volume 1872 of *Lecture Notes in Computer Science*, pages 25–41. Springer, 2000.

[19] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma (extended abstract). In *31st ACM STOC*, pages 537–546. ACM Press, May 1999.

[20] Aaron D Wyner. The wire-tap channel. *Bell system technical journal*, 54(8):1355–1387, 1975.