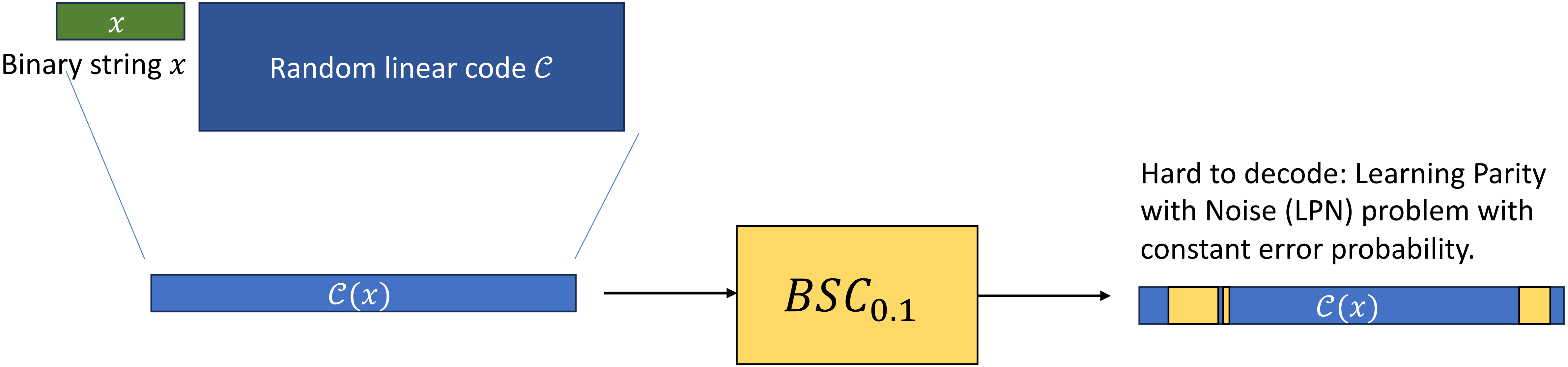
Computational Wiretap Coding from Indistinguishability Obfuscation

Yuval Ishai (Technion), Aayush Jain (CMU), Paul Lou (UCLA),
Amit Sahai (UCLA), Mark Zhandry (NTT Research)

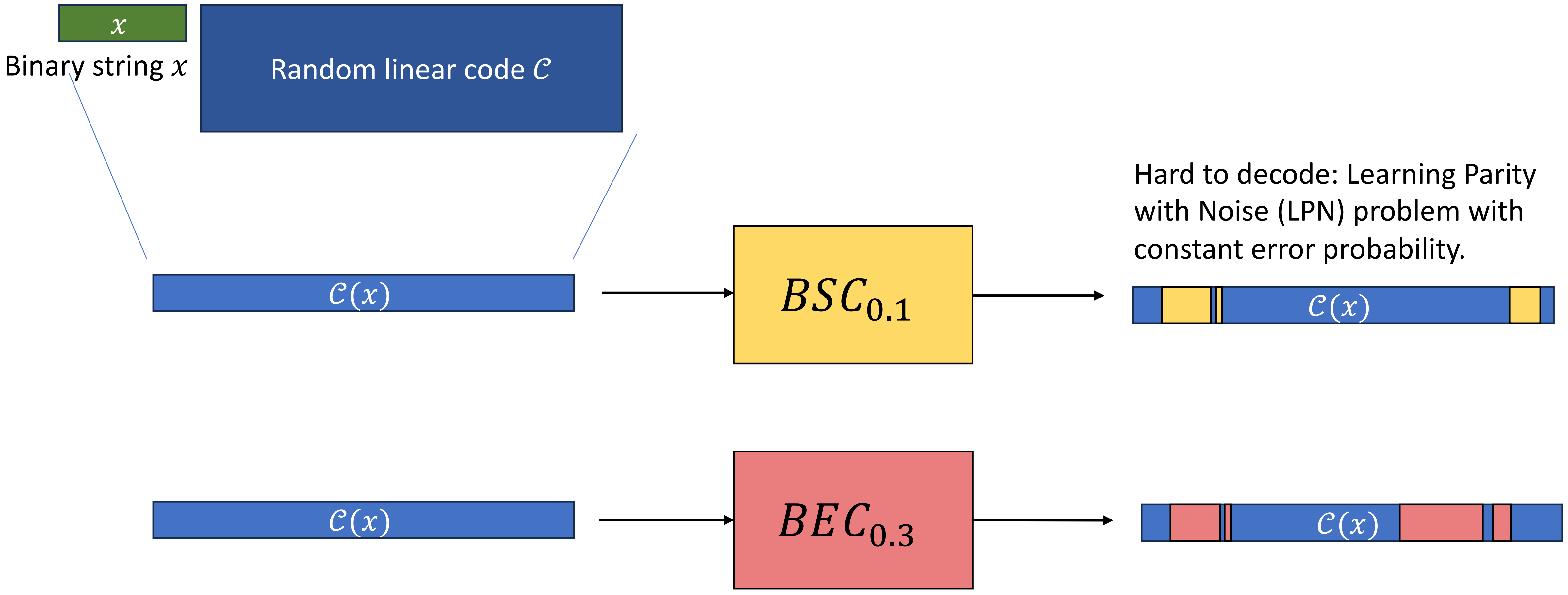
Teaser: Interesting special case of the general wiretap problem

Teaser: Curious Coding Theory Question



Hard to decode: Learning Parity with Noise (LPN) problem with constant error probability.

Teaser: Curious Coding Theory Question



For the right choice of parameters, Gaussian elimination recovers x.

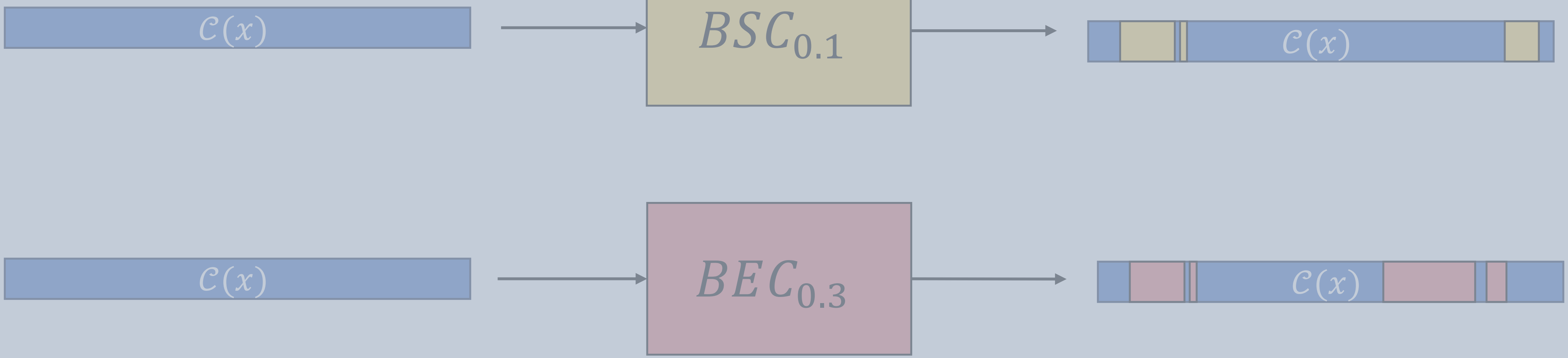
Teaser: Curious Coding Theory Question

Do there exist error-correcting codes that satisfy the following?

1. **Easy to decode from 0.1 bitflip error rate. [LDPC, BCH, etc.]**
2. **Computationally hard to decode from 0.3 erasure rate. [Linear codes fail]**

Binary st

Hard to decode: Learning Parity with Noise (LPN) problem with constant error probability.



For the right choice of parameters, Gaussian elimination recovers x.

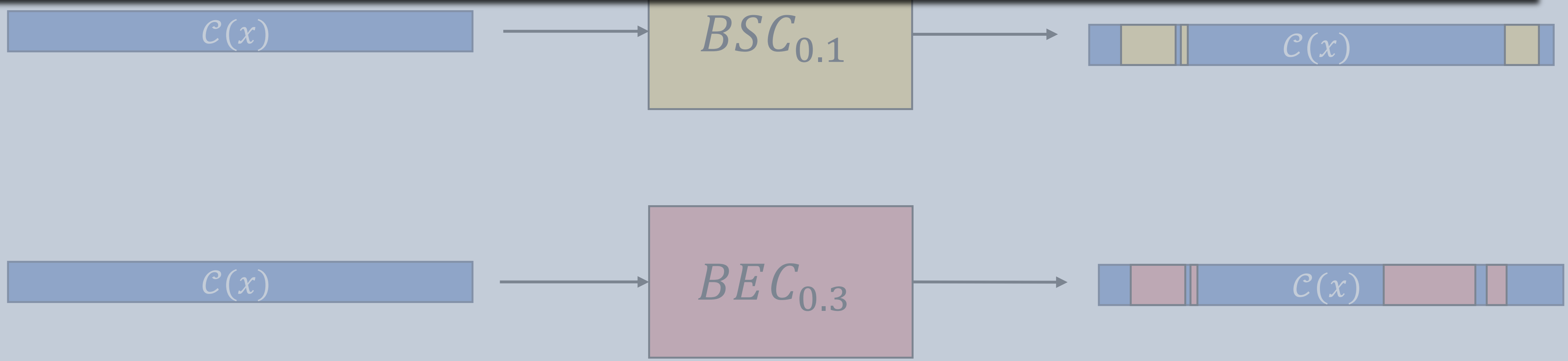
Teaser: Curious Coding Theory Question

Do there exist error-correcting codes that satisfy the following?

1. **Easy to decode from 0.1 bitflip error rate. [LDPC, BCH, etc.]**
2. **Computationally hard to decode from 0.3 erasure rate. [Linear codes fail]**

Hard to decode: Learning Parity

Until last year, no such codes known to satisfy both.



For the right choice of parameters, Gaussian elimination recovers x.

Teaser: Curious Coding Theory Question

Do there exist error-correcting codes that satisfy the following?

1. **Easy to decode from 0.1 bitflip error rate. [LDPC, BCH, etc.]**
2. **Computationally hard to decode from 0.3 erasure rate. [Linear codes fail]**

Until last year, no such codes known to satisfy both.

Ishai, Korb, Lou, Sahai '22: Yes*, in the ideal obfuscation model (or non-standard VBB obfuscation assumptions)!

For the right choice of parameters, Gaussian elimination recovers x.

Teaser: Curious Coding Theory Question

Do there exist error-correcting codes that satisfy the following?

1. **Easy to decode from 0.1 bitflip error rate. [LDPC, BCH, etc.]**
2. **Computationally hard to decode from 0.3 erasure rate. [Linear codes fail]**

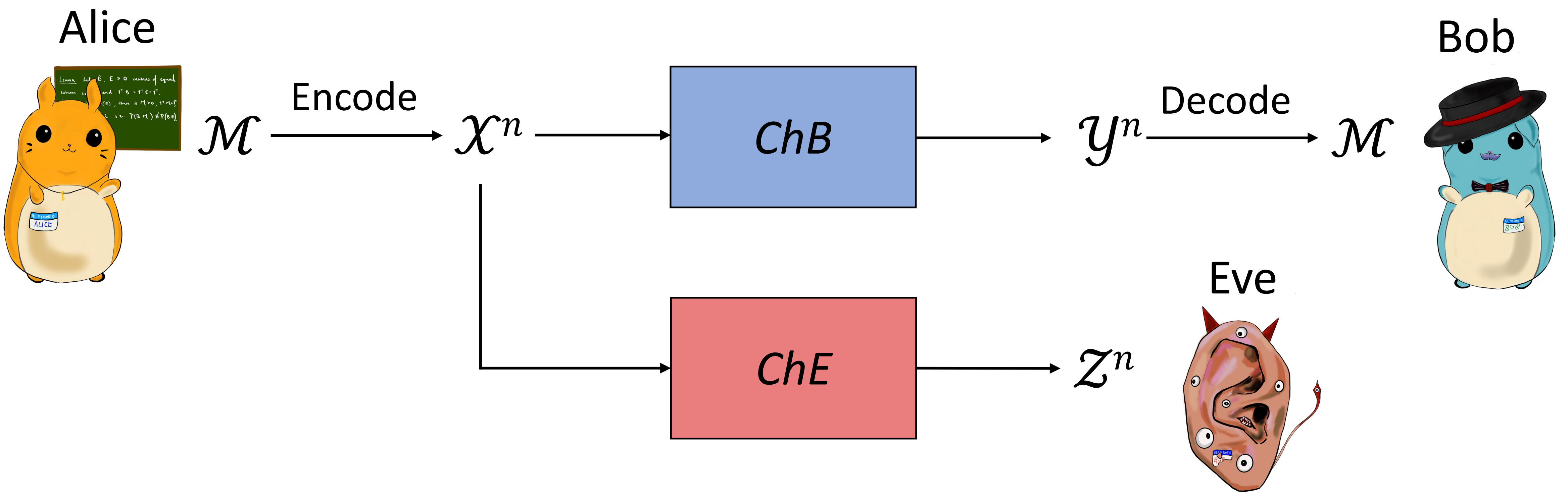
Until last year, no such codes known to satisfy both.

Ishai, Korb, Lou, Sahai '22: Yes*, in the ideal obfuscation model (or non-standard VBB obfuscation assumptions)!

This Work: Yes*, assuming standard hardness assumptions!

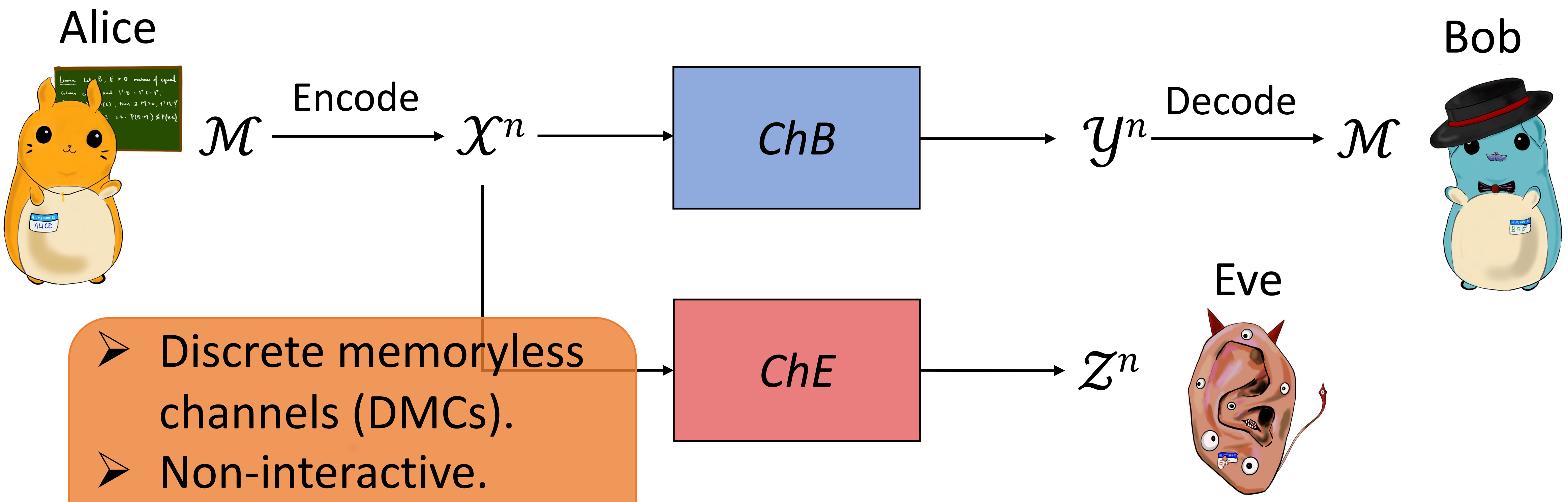
For the right choice of parameters, Gaussian elimination recovers x.

General Setting: Wiretap Channel [Wyn75]



Goal: Alice wants to send a message to Bob without Eve learning it.

More General Setting:
Wiretap Channel [Wyn75]



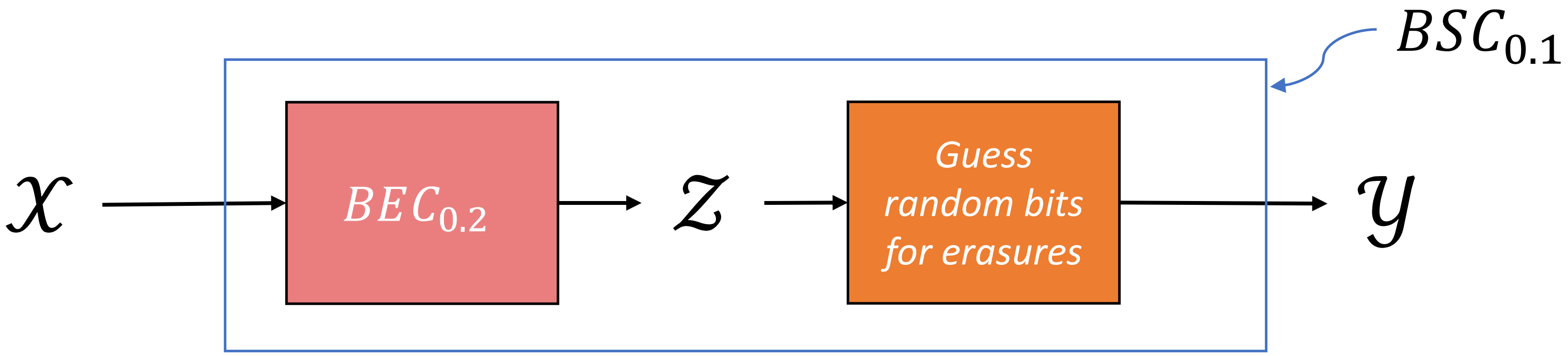
- Discrete memoryless channels (DMCs).
- Non-interactive.
- No shared secrets.

Goal: Alice wants to send a message to Bob without Eve learning it.

For what pairs of channels do wiretap coding schemes exist?

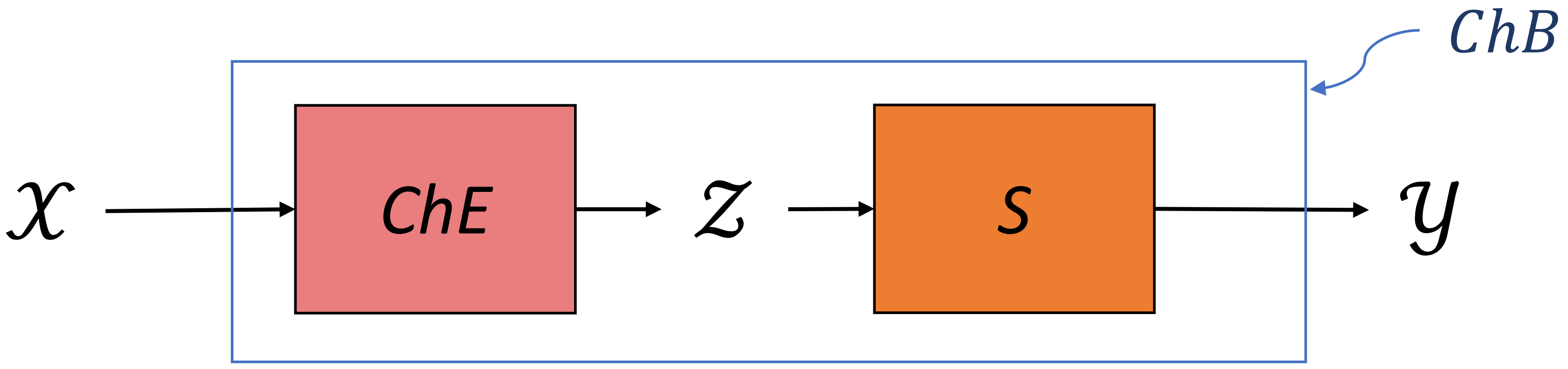
Intuitive Impossibility for Degraded Pairs

Impossible for channel pair $(BSC_{0.1}, BEC_{0.2})$. Eve can perfectly simulate $BSC_{0.1}$'s output distribution using an output of $BEC_{0.2}$.



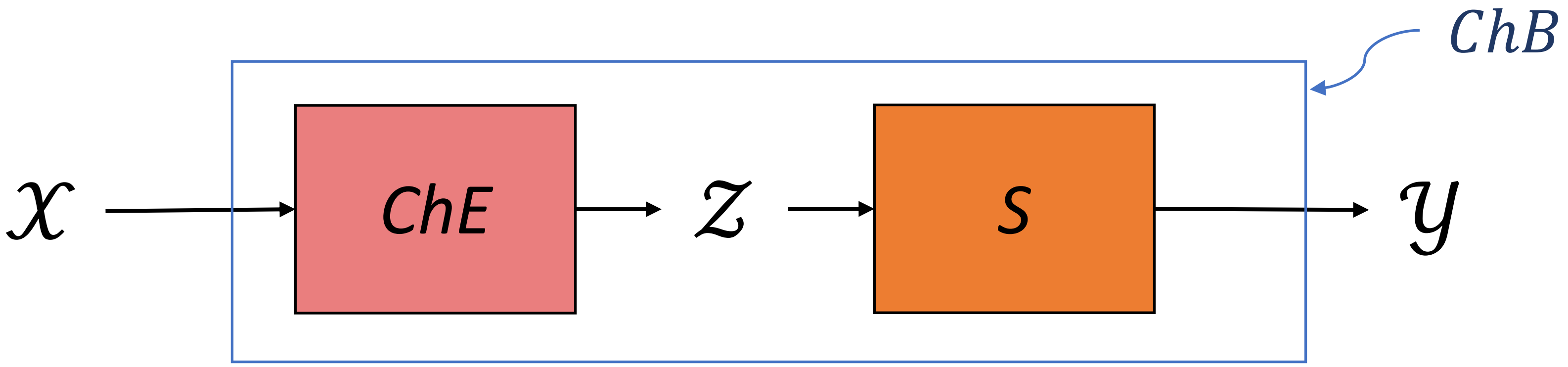
Intuitive Impossibility for Degraded Pairs

Impossible for any channel pair (ChB, ChE) where Eve can perfectly simulate ChB's output distribution using an output of ChE.



Intuitive Impossibility for Degraded Pairs

Impossible for any channel pair (ChB, ChE) where Eve can perfectly simulate ChB's output distribution using an output of ChE.

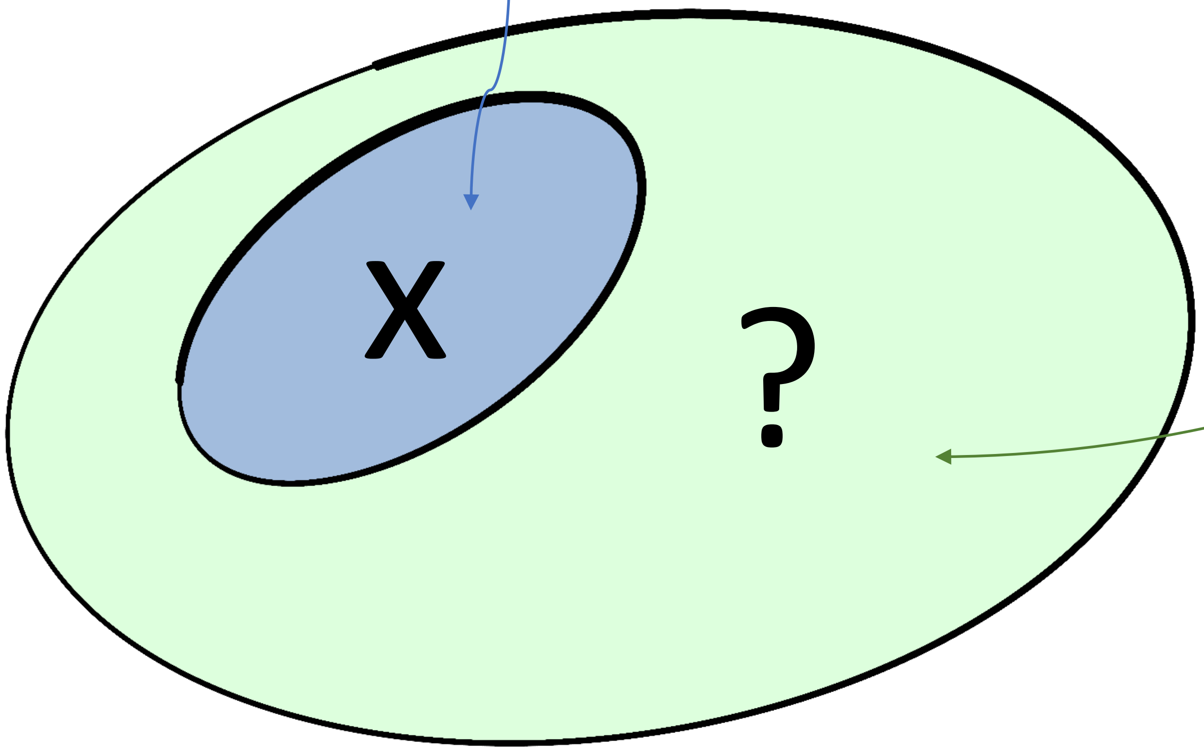


Degradation: ChB is a degradation of ChE if and only if Eve can perfectly simulate ChB using ChE.

Existence of Wiretap Coding Schemes

None for (ChB, ChE) where ChB is a degradation of ChE.

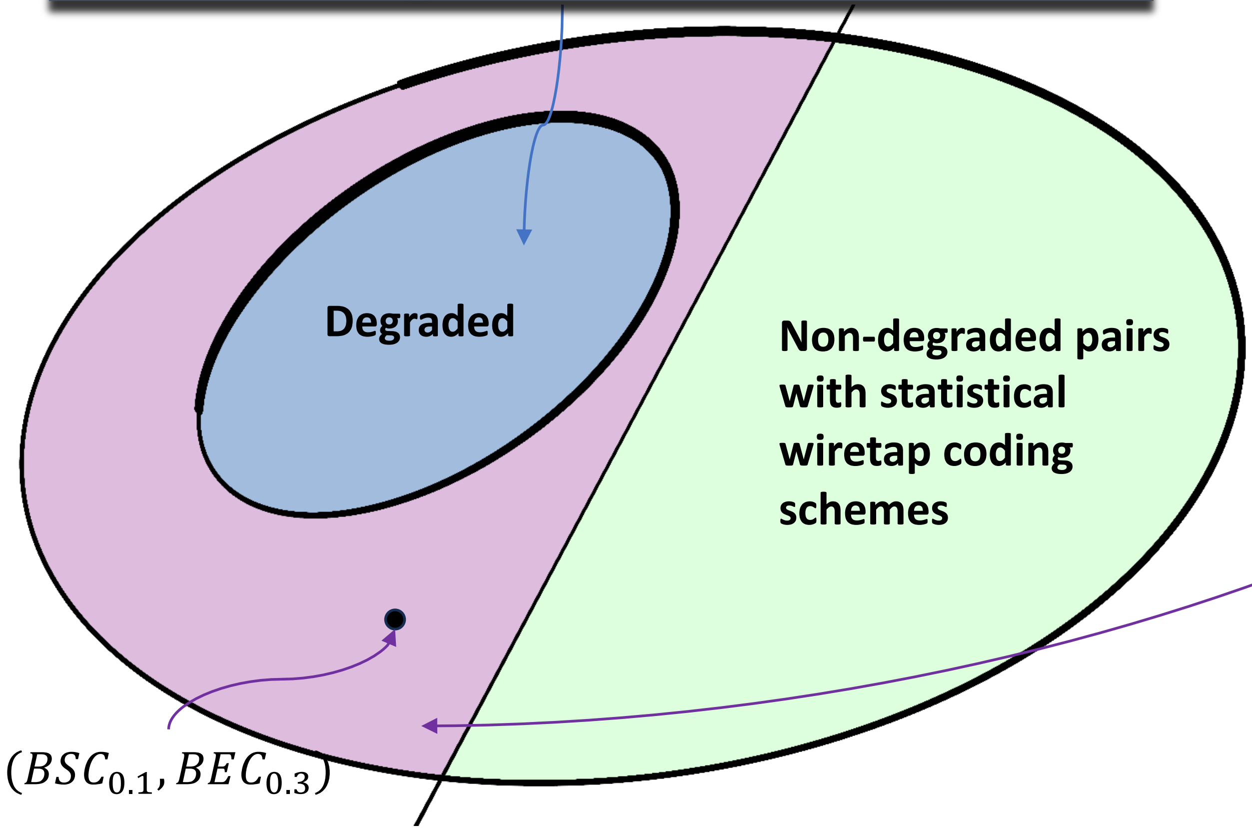
Do there exist wiretap coding schemes for non-degraded channel pairs (ChB, ChE)?



Existence of Wiretap Coding Schemes

None for (ChB, ChE) where ChB is a degradation of ChE.

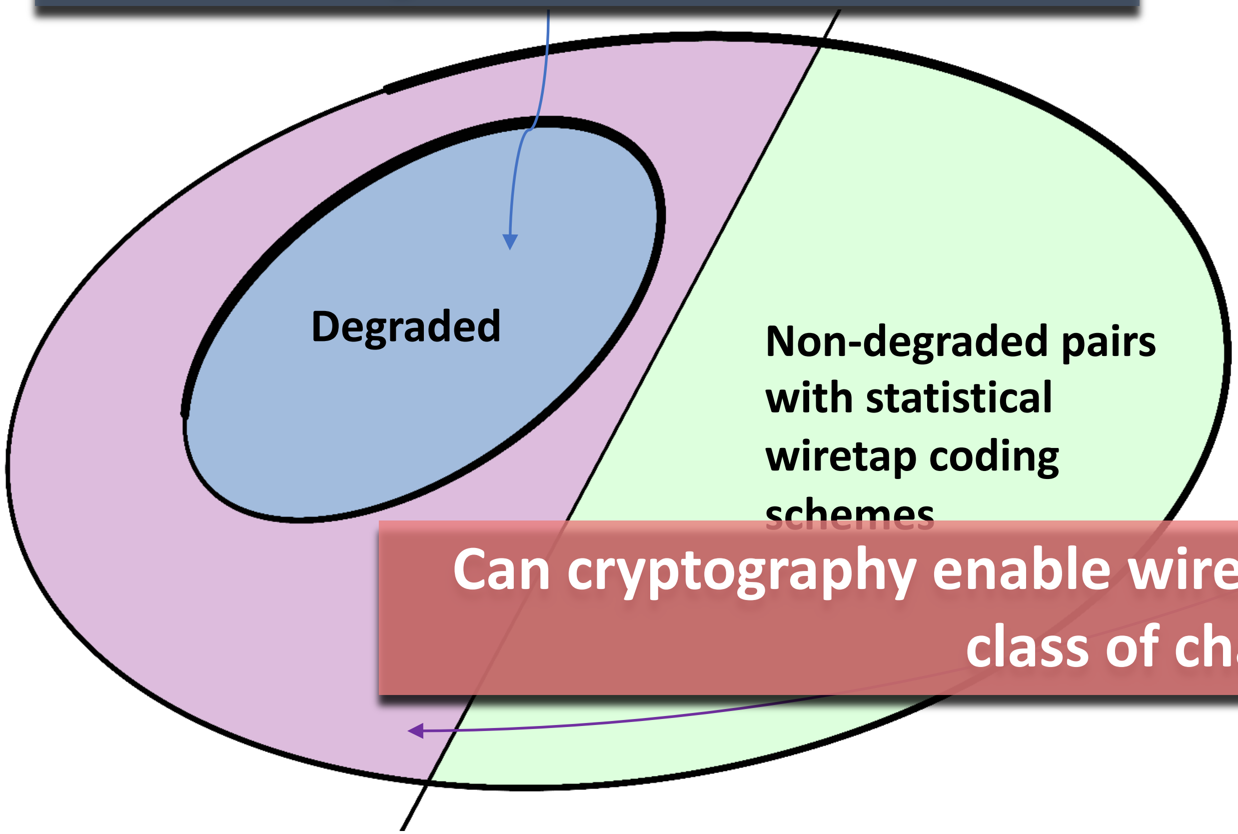
Csiszár, Korner '78: There are non-degraded channel pairs that do not have statistical wiretap coding schemes.



Existence of Wiretap Coding Schemes

None for (ChB, ChE) where ChB is a degradation of ChE.

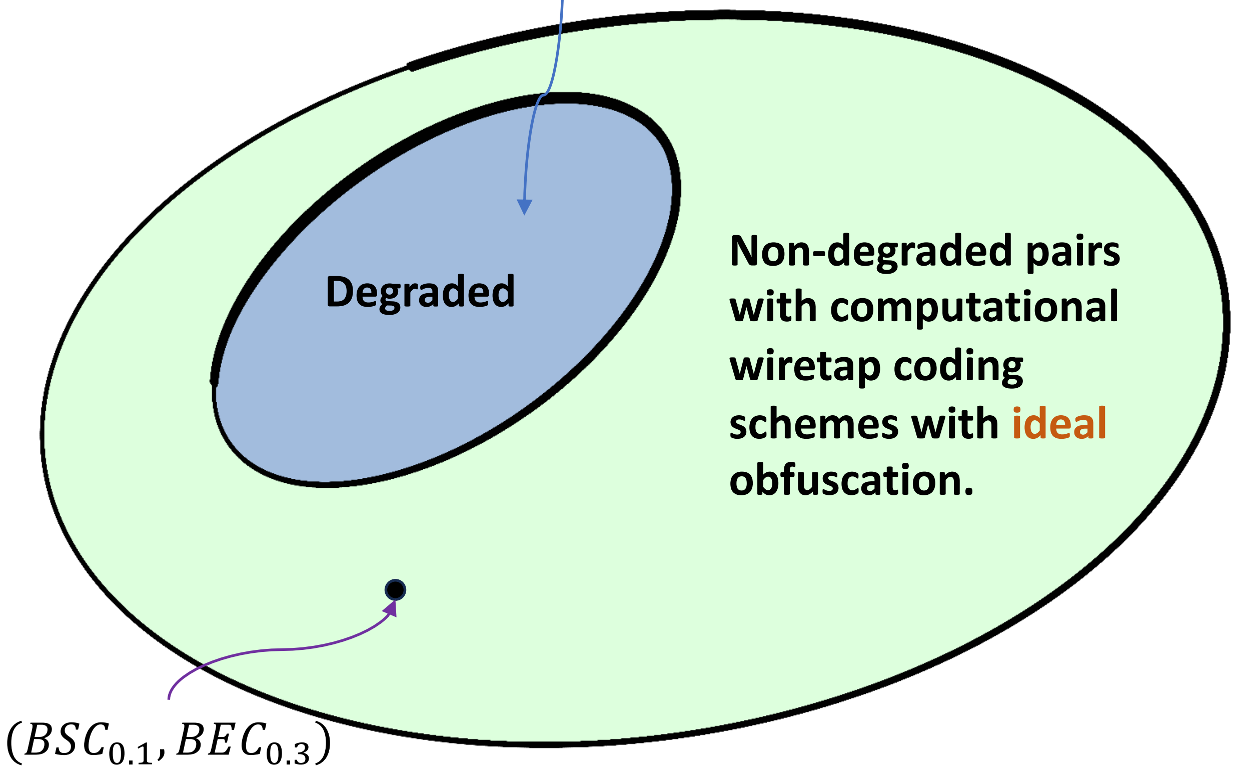
Csiszár, Korner '78: There are non-degraded channel pairs that do not have statistical wiretap coding schemes.



Can cryptography enable wiretap coding schemes for a larger class of channel pairs?

Existence of Wiretap Coding Schemes

None for (ChB, ChE) where ChB is a degradation of ChE.



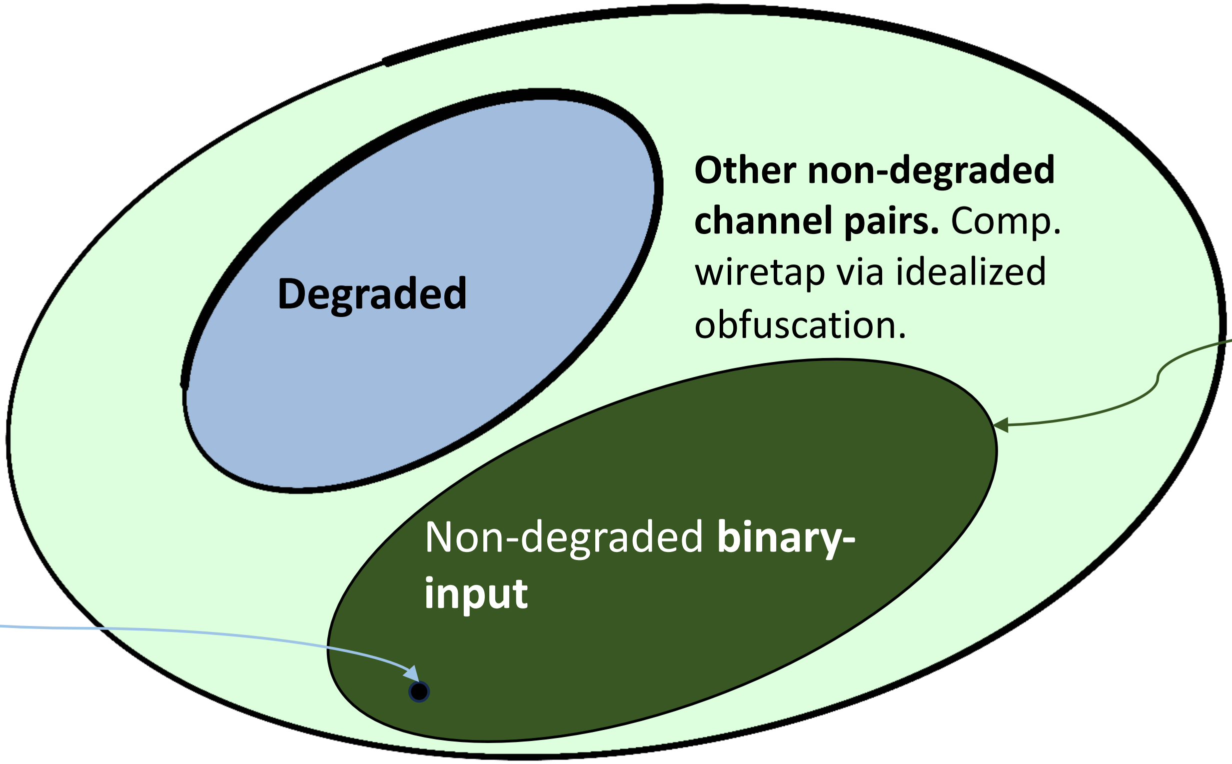
Csiszár, Korner '78: There are non-degraded channel pairs that do not have statistical wiretap coding schemes.

Ishai, Korb, Lou, Sahai '22: There exists a computational wiretap coding scheme for all non-degraded channel pairs in **the Ideal Obfuscation Model (or non-std. VBB obfuscation)**.

Can we obtain computational wiretap coding schemes from standard assumptions?

Our Main Result: YES

Theorem: Assuming the existence of indistinguishability obfuscation (iO) and injective PRGs, there exists a computational wiretap coding scheme for any pair of non-degraded **binary-input** channels (ChB, ChE).



Non-degraded binary-input channel pairs with computational wiretap coding schemes from **standard assumptions**

Solves* the teaser:

$(BSC_{0.1}, BEC_{0.3})$

Our Techniques

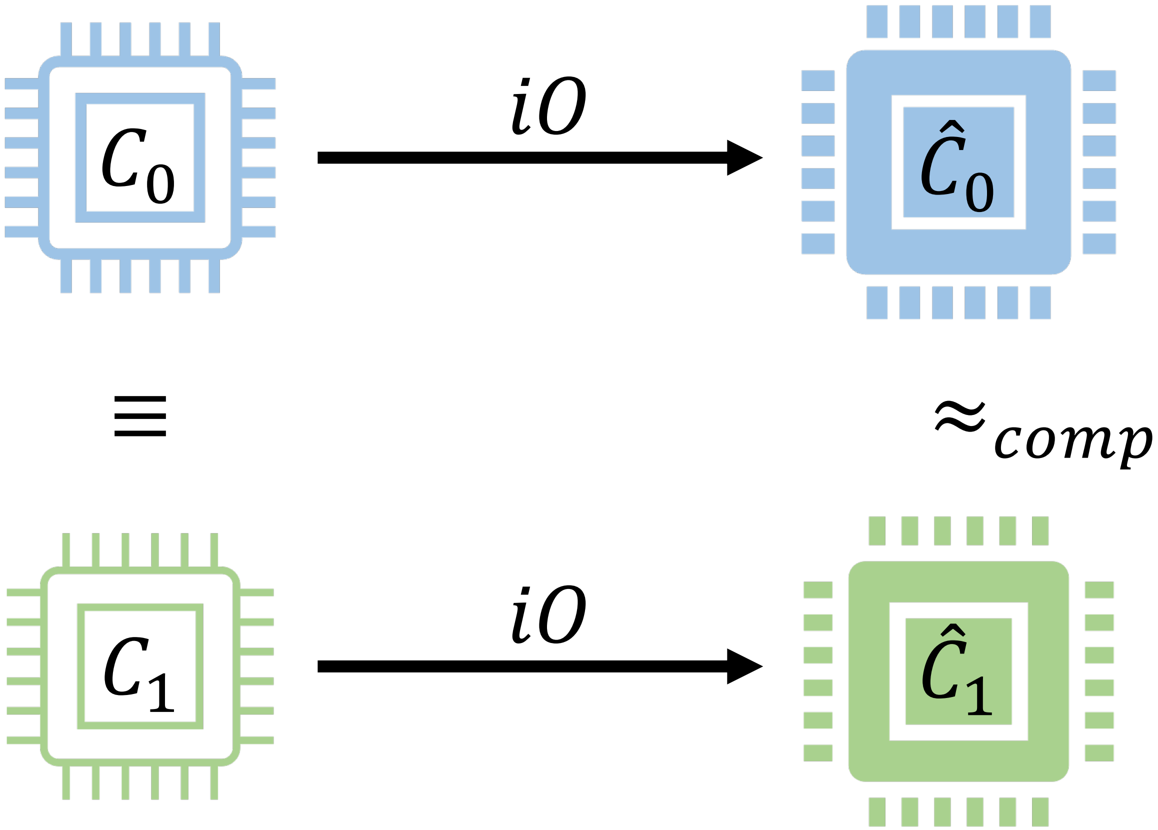
1. Using iO and injective PRGs, we construct a Hamming ball obfuscator.
   - Construction uses a new gadget: PRG with Self-Correction.
   - Using this, we build computational wiretap coding schemes for binary asymmetric channels (BAC) and binary asymmetric erasure channels (BAEC).
2. We introduce a polytope characterization of degradation.
   - Using this polytope characterization, we reduce the problem of constructing a computational wiretap coding scheme for any non-degraded binary-input channel pair to constructing one for (BAC, BAEC).

Focus of this talk:
A computational wiretap coding scheme from $i\mathcal{O}$ for ($ChB = BSC_{0.1}$, $ChE = BEC_{0.3}$)

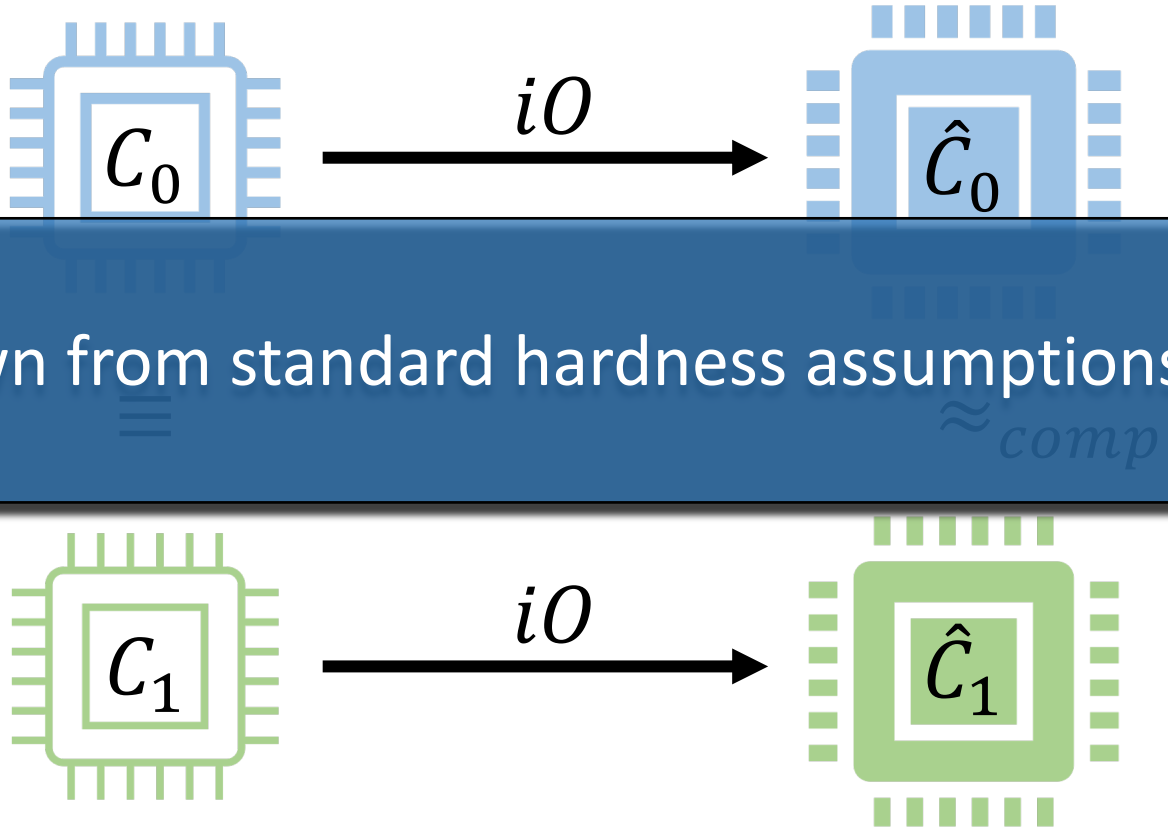
*Construction idea easily extends to the non-degraded (BAC, BAEC) setting.

**See paper or slide appendix for extension to all non-degraded binary-input.

Indistinguishability Obfuscation ($i\mathcal{O}$) [BGIRSVY01]



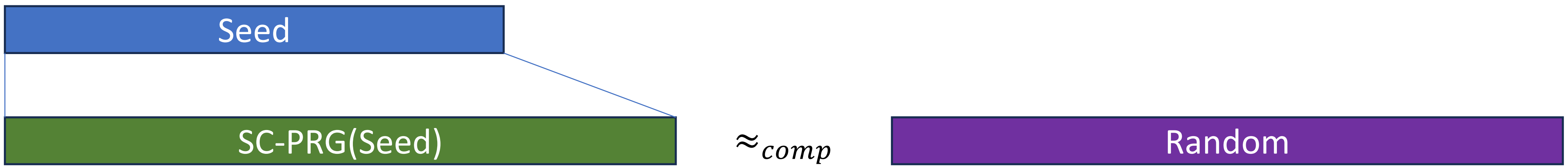
Indistinguishability Obfuscation ($i\mathcal{O}$) [BGIRSVY01]



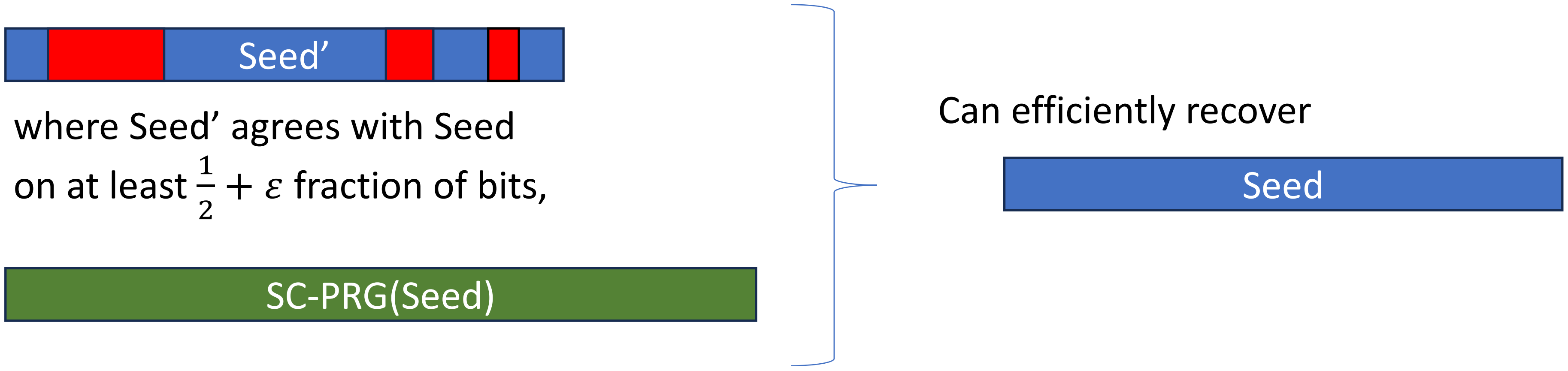
Now known from standard hardness assumptions !! [JLS21]

New Gadget:
PRG with Self-Correction (SCPRG)

1. Polynomial Stretch & Pseudorandomness



2. ε-Self-Correction



New Gadget:
PRG with Self-Correction (SCPRG)

1. Polynomial Stretch & Pseudorandomness

For this talk, $\varepsilon = \frac{1}{12}$. In general, some constant.

Seed

SC-PRG(Seed) $\approx_{comp}$ Random

2. ε-Self-Correction (recovery works w.h.p. over choices of seeds)

Seed'

where Seed' agrees with Seed on at least $\frac{1}{2} + \varepsilon$ fraction of bits,

SC-PRG(Seed)

Can efficiently recover

Seed

New Gadget:
PRG with Self-Correction (SCPRG)

1. Polynomial Stretch & Pseudorandomness

Seed

SC-PRG(Seed) $\approx_{comp}$ Random

Bogdanov, Qiao ‘12: Goldreich PRG, even with linear stretch, is a SCPRG.

2. ε-Self-Correction

Seed’

where Seed’ agrees with Seed on at least $\frac{1}{2} + \varepsilon$ fraction of bits,

SC-PRG(Seed)

Can efficiently recover

Seed

New Gadget:
PRG with Self-Correction (SCPRG)

1. Polynomial Stretch & Pseudorandomness

Seed

SC-PRG(Seed) $\approx_{comp}$ Random

Our Work: Injective SC-PRG from any injective PRG.

2. ε-Self-Correction

Seed'

where Seed' agrees with Seed on at least $\frac{1}{2}+\varepsilon$ fraction of bits,

SC-PRG(Seed)

Can efficiently recover

Seed

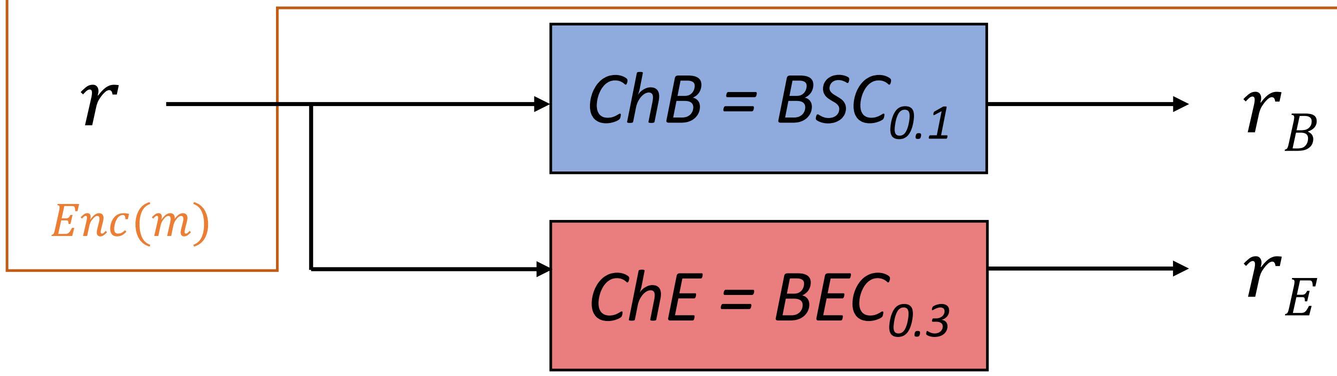
$ChB = BSC_{0.1}$, $ChE = BEC_{0.3}$

Using ideal obfuscation [IKLS22]: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send an obfuscation of f_r, encoded to Bob's channel.

$f_r(r')$:

- If $\Delta(r', r) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Correctness:
$f_r(r_B) = m$ with high probability



$ChB = BSC_{0.1}$, $ChE = BEC_{0.3}$

Using ideal obfuscation [IKLS22]: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send an obfuscation of f_r, encoded to Bob's channel.

$f_r(r')$:

- If $\Delta(r', r) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

iO

Correctness:
$f_r(r_B) = m$ with high probability

r

$Enc(m)$

Eve's best guess for r' has ≈ 0.15 error rate.

If we were using an ideal obfuscation, then r and m are hidden.

$ChB = BSC_{0.1}$, $ChE = BEC_{0.3}$

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send an iO of f_r, encoded to Bob's channel.

$f_r(r')$:

- If $\Delta(r', r) < 0.1n + n^{0.9}$ output m
- Output $\bot$ otherwise.

Correctness:
$f_r(r_B) = m$ with high probability

r

$Enc(m)$

Security: Why does $iO(f_r)$ hide m or r?

Security: What Does Eve See?

Eve sees:

r_E = ⊥010⊥1011⊥

Eve does not know:

r = 1010010110

$f_r(r')$:

- If $\Delta(r', r) < 0.1n + n^{0.9}$ output m
- Output ⊥ otherwise.

Security: What Does Eve See?

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

Eve does not know:

$r = 1010010110$

Goal: Use a hybrid argument to show that this circuit is indistinguishable from the null circuit.

Problem: There are **exponentially** many points in the Hamming ball!

$f_r(r')$:

- If $\Delta(r', r) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Security: What Does Eve See?

Eve sees:

$r_E = \bot 010 \bot 1011 \bot$

Eve does not know:

$r = 1010010110$

$f_r(r')$:

- If $\Delta(r', r) < 0.1n + n^{0.9}$ output m
- Output $\bot$ otherwise.

Critical observation: In intermediate hybrids, this circuit can depend on the actual received string r_E.

Security: What Does Eve See?

Eve sees:

$\boxed{r_E = \perp 010 \perp 1011 \perp}$

$S_\perp = \{1, 5, 10\}$ $S_{0,1} = [10] \setminus S_\perp$

$f_r(r')$:

- If $\Delta(r', r) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Eve does not know:

$r = 1010010110$

Critical observation: In intermediate hybrids, this circuit can depend on the actual received string r_E.

Security: An Indistinguishable Viewpoint

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$ $\quad$ $S_{0,1} = [10] \setminus S_\perp$

Eve does not know:

$r = 1010010110$

Split the hardcoded r into two substrings depending on $S_\perp$

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}, r_{S_\perp}, S_\perp$.

- If $\Delta(r', r) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Security: An Indistinguishable Viewpoint

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$ $\quad S_{0,1} = [10] \setminus S_\perp$

Eve does not know:

$r = 1010010110$

r' is Eve's guess.

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}, r_{S_\perp}, S_\perp$.

- If $\Delta(r', r) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Security: An Indistinguishable Viewpoint

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$ $\quad S_{0,1} = [10] \setminus S_\perp$

Eve does not know:

$r = 1010010110$

Rewrite the Hamming distance condition

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}, r_{S_\perp}, S_\perp$.

- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta(r'_{S_{0,1}}, r_{S_{0,1}}) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Security: An Indistinguishable Viewpoint

Eve sees:

r_E = $\perp 010 \perp 1011 \perp$

$S_\perp$ = $\{1, 5, 10\}$ $\quad S_{0,1}$ = $[10] \setminus S_\perp$

Eve does not know:

r = 1010010110

$r'_{S_\perp}, r'_{S_{0,1}}$ are substrings of Eve's guess.

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}, r_{S_\perp}, S_\perp$

- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta(r'_{S_{0,1}}, r_{S_{0,1}}) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Security: An Indistinguishable Viewpoint

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$ $\quad S_{0,1} = [10] \setminus S_\perp$

Eve does not know:

$r = 1010010110$

$r_{S_\perp}, r_{S_{0,1}}$ are substrings of the sent random string.

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}, r_{S_\perp}, S_\perp$.

- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta(r'_{S_{0,1}}, r_{S_{0,1}}) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Security: An Indistinguishable Viewpoint

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$ $\quad S_{0,1} = [10] \setminus S_\perp$

Eve does not know:

$r = 1010010110$

Functionally Equivalent to $f_r(\cdot)$!!

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}, r_{S_\perp}, S_\perp$.

- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta(r'_{S_{0,1}}, r_{S_{0,1}}) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Security: An Indistinguishable Viewpoint

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$ $S_{0,1} = [10] \setminus S_\perp$

Eve does not know:

$r = 1010010110$

Eve knows the non-erased coordinates.

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}, r_{S_\perp}, S_\perp$.

- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta(r'_{S_{0,1}}, r_{S_{0,1}}) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Security: An Indistinguishable Viewpoint

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$

Eve does not know:

$r = 1010010110$

Eve's best strategy is to uniformly guess for $r'_{S_\perp}$.

There are exponentially many guesses that cause the function to output m.

We will compress them into a single branch that can be removed by a hybrid argument.

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}, r_{S_\perp}, S_\perp$

- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta(r'_{S_{0,1}}, r_{S_{0,1}}) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Using injective length-tripling SCPRGs

Eve sees:

r_E = $\perp 010 \perp 1011 \perp$

$S_\perp$ = $\{1, 5, 10\}$ $S_{0,1}$ = $[10] \setminus S_\perp$

Eve does not know:

r = 1010010110

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}$, $\cancel{r_{S_\perp}}$, $S_\perp$.

- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta\left(r'_{S_{0,1}}, r_{S_{0,1}}\right) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Using injective length-tripling SCPRGs

Eve sees:

r_E = ⊥010⊥1011⊥

$S_\perp = \{1, 5, 10\}$ $\quad S_{0,1} = [10] \setminus S_\perp$

Eve does not know:

r = 1010010110

Replace with $SCPRG_\varepsilon(r_{S_\perp})$ for some choice of ε dependent on degradation condition. Here, $\varepsilon = \frac{1}{12}$.

$f^{(1)}(r')$:

Constants: $r_{S_{0,1}}, \cancel{r_{S_\perp}}, S_\perp$.

- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta(r'_{S_{0,1}}, r_{S_{0,1}}) < 0.1n + n^{0.9}$ output m
- Output ⊥ otherwise.

Using injective length-tripling SCPRGs

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$

Eve does not know:

$r = 1010010110$

Parameter ε , dependent on degradation condition, is set so that Eve is unable to recover. Here, $\varepsilon = \frac{1}{12}$.

$f^{(2)}(r')$:

Constants: $r_{S_{0,1}}, SCPRG_\varepsilon(r_{S_\perp}), S_\perp$.

- Let $\alpha := SCPRG_\varepsilon.Recover\left(SCPRG_\varepsilon(r_{S_\perp}), r'_{S_\perp}\right)$.
- If $SCPRG_\varepsilon(\alpha) \neq SCPRG_\varepsilon(r_{S_\perp})$, then output $\perp$.
- Otherwise, set $r_{S_\perp} \leftarrow \alpha$.
- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta\left(r'_{S_{0,1}}, r_{S_{0,1}}\right) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Using injective length-tripling SCPRGs

Eve sees:

r_E = $\perp 010 \perp 1011 \perp$

$S_\perp$ = $\{1, 5, 10\}$ $\qquad$ $S_{0,1}$ = $[10] \setminus S_\perp$

Eve does not know:

r = 1010010110

From Eve's point of view, $r_{S_\perp}$ is an unknown uniform random string.

$f^{(2)}(r')$:

Constants: $r_{S_{0,1}}, SC-PRG_\varepsilon(r_{S_\perp}), S_\perp$.

- Let $\alpha \coloneqq SCPRG_\varepsilon.Recover\left(SCPRG_\varepsilon(r_{S_\perp}), r'_{S_\perp}\right)$.
- If $SCPRG_\varepsilon(\alpha) \neq SCPRG_\varepsilon(r_{S_\perp})$, then output $\perp$.
- Otherwise, set $r_{S_\perp} \leftarrow \alpha$.
- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta\left(r'_{S_{0,1}}, r_{S_{0,1}}\right) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Using injective length-tripling SCPRGs

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$ $S_{0,1} = [10] \setminus S_\perp$

Eve does not know:

$r = 1010010110$

Can therefore apply pseudorandomness property.

$f^{(3)}(r')$:

Constants: $r_{S_{0,1}}, R, S_\perp$.

- Let $\alpha \coloneqq SCPRG_\varepsilon.Recover(R, r'_{S_\perp})$.
- If $SCPRG_\varepsilon(\alpha) \neq R$, then output $\perp$.
- Otherwise, set $r_{S_\perp} \leftarrow \alpha$.
- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta(r'_{S_{0,1}}, r_{S_{0,1}}) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

Using injective length-tripling SCPRGs

Eve sees:

$r_E = \perp 010 \perp 1011 \perp$

$S_\perp = \{1, 5, 10\}$

Eve does not know:

$r = 1010010110$

With overwhelming probability R is not in the range of the $SCPRG$, so will be functionally equivalent to null circuit.

$f^{(3)}(r')$:

Constants: $r_{S_{0,1}}$, R, $S_\perp$.

- Let $\alpha \coloneqq SCPRG_\varepsilon.Recover(R, r'_{S_\perp})$.
- If $SCPRG_\varepsilon(\alpha) \neq R$, then output $\perp$.
- Otherwise, set $r_{S_\perp} \leftarrow \alpha$.
- If $\Delta(r'_{S_\perp}, r_{S_\perp}) + \Delta(r'_{S_{0,1}}, r_{S_{0,1}}) < 0.1n + n^{0.9}$ output m
- Output $\perp$ otherwise.

End of the Security Proof: Null Circuit

$f^{(4)}(r')$:

- Output $\perp$.

“Code Offset” construction of SCPRG

Injective PRG G.

List-decodable error correcting code $\mathcal{C}$ for up to $\frac{1}{2} - \varepsilon$ error rate for any constant $\varepsilon > 0$.

Concatenated code of binary Reed-Solomon codes with Hadamard code [Sudan, Trevisan, Vadhan ’99, Sudan ‘00]

$SCPRG_{\varepsilon}(s_1, s_2)$:

- Output $(s_1 + \mathcal{C}(s_2), G(s_2))$.

“Code Offset” construction of SCPRG

Injective PRG G.

List-decodable error correcting code $\mathcal{C}$ for up to $\frac{1}{2} - \varepsilon$ error rate for any constant $\varepsilon > 0$.

e.g. concatenated code of binary Reed-Solomon codes with Hadamard code
[Sudan, Trevisan, Vadhan ’99, Sudan ‘00]

$SCPRG_{\varepsilon}(s_1, s_2)$:

- Output $(s_1 + \mathcal{C}(s_2), G(s_2))$.

Pseudorandomness: s_1 is uniform random, so $s_1 + \mathcal{C}(s_2)$ is uniform random. Then, apply pseudorandomness of $G(s_2)$.

“Code Offset” construction of SCPRG

Injective PRG G.

List-decodable error correcting code $\mathcal{C}$ for up to $\frac{1}{2} - \varepsilon$ error rate for any constant $\varepsilon > 0$.

e.g. concatenated code of binary Reed-Solomon codes with Hadamard code
[Sudan, Trevisan, Vadhan ’99, Sudan ‘00]

$SCPRG_\varepsilon(s_1, s_2)$:

- Output $(s_1 + \mathcal{C}(s_2), G(s_2))$.

Self-correction: Can show, if $s_1', s'_2 \approx s_1, s_2$ and for appropriate lengths of s_1 and s_2, then $s_1' \approx s_1$.

Therefore, if $s_1', s'_2 \approx s_1, s_2$ then can recover a polynomial size list containing s_2 from $s_1 + \mathcal{C}(s_2)$.

Use $G(s_2)$ iterate over list to find s_2, then recover s_1.

Recap

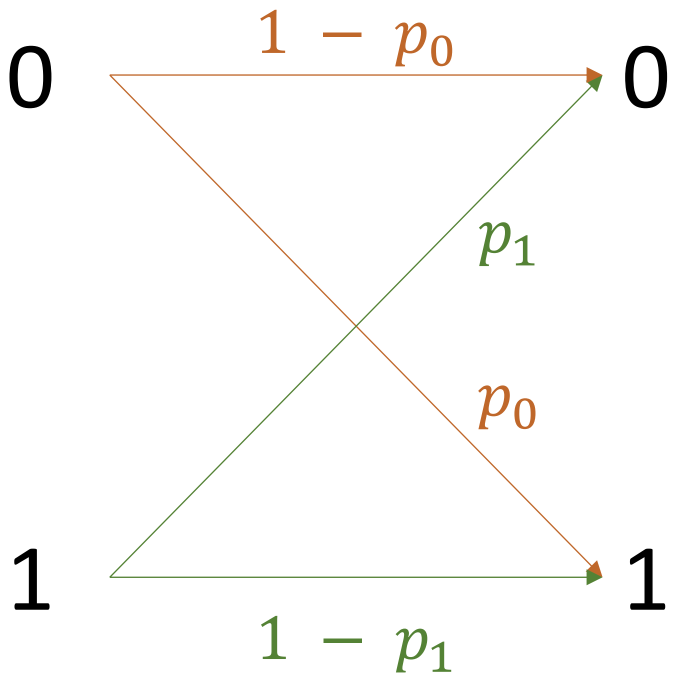
We sketched the construction and security proof for a computational wiretap coding scheme for the non-degraded (BSC, BEC) case via iO & injective PRG.

Theorem: Assuming the existence of indistinguishability obfuscation (iO) and injective PRGs, there exists a computational wiretap coding scheme for any pair of non-degraded **binary-input** channels (ChB, ChE).

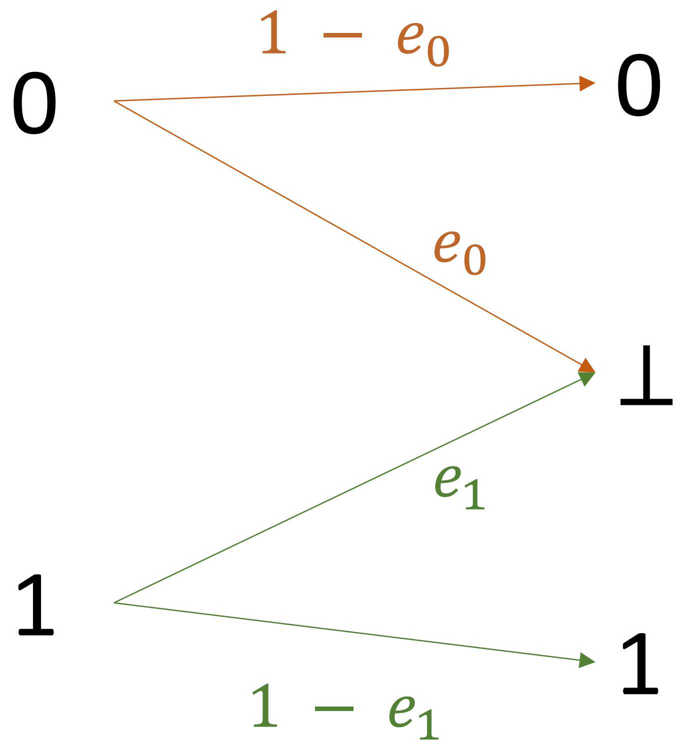
1. The given construction idea easily extends to the non-degraded $(BAC, BAEC)$ setting.

Theorem: Assuming the existence of indistinguishability obfuscation (iO) and injective PRGs, there exists a computational wiretap coding scheme for any pair of non-degraded **binary-input** channels (ChB, ChE).

1. The given construction idea easily extends to the non-degraded ($BAC, BAEC$) setting.



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

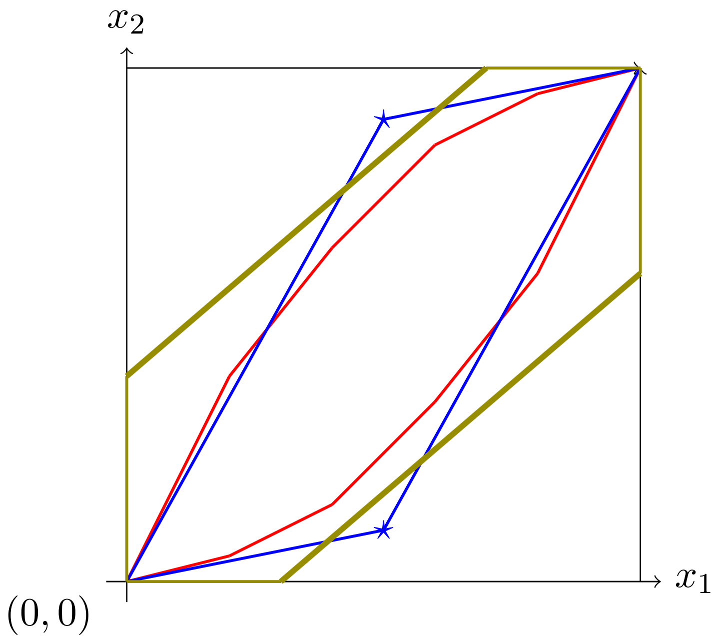


$$\begin{bmatrix} 1 - e_0 & 0 & e_0 \\ 0 & 1 - e_1 & e_1 \end{bmatrix}$$

Theorem: Assuming the existence of indistinguishability obfuscation (iO) and injective PRGs, there exists a computational wiretap coding scheme for any pair of non-degraded **binary-input** channels (ChB, ChE).

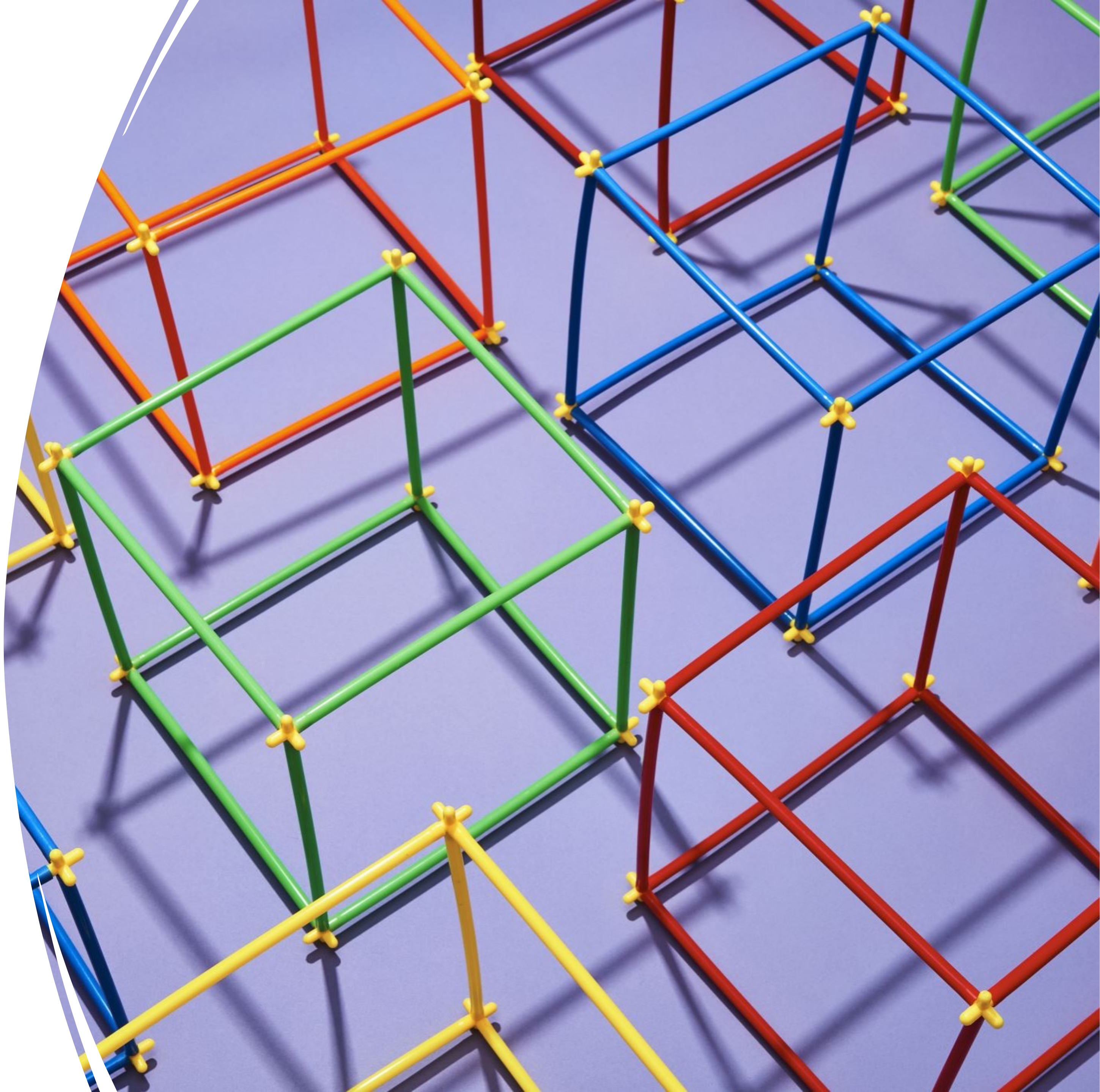
1. The given construction idea easily extends to the non-degraded $(BAC, BAEC)$ setting.

2. The case of every non-degraded binary-input channel pair (ChB, ChE) reduces to (1).



Some Open Directions

- Expanding construction beyond binary-input channels.
  - Characterize degradation for dimension three and beyond.
- Realizing computational wiretap coding from simpler cryptographic primitives or directly from hardness assumptions like LWE.
- Addressing the asterisk* in the initial riddle: Can we derandomize the encoding?

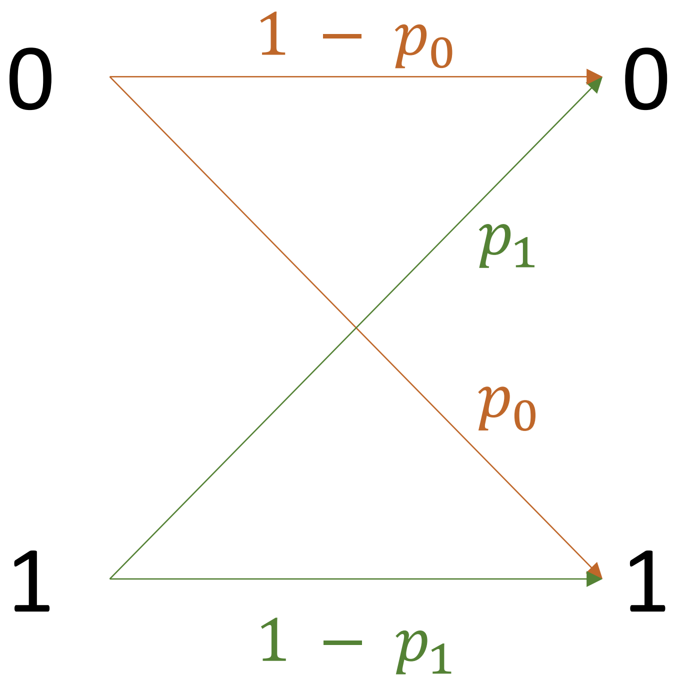


Thank you !

Appendix: The BAC/BAEC Case and General Binary-Input Case

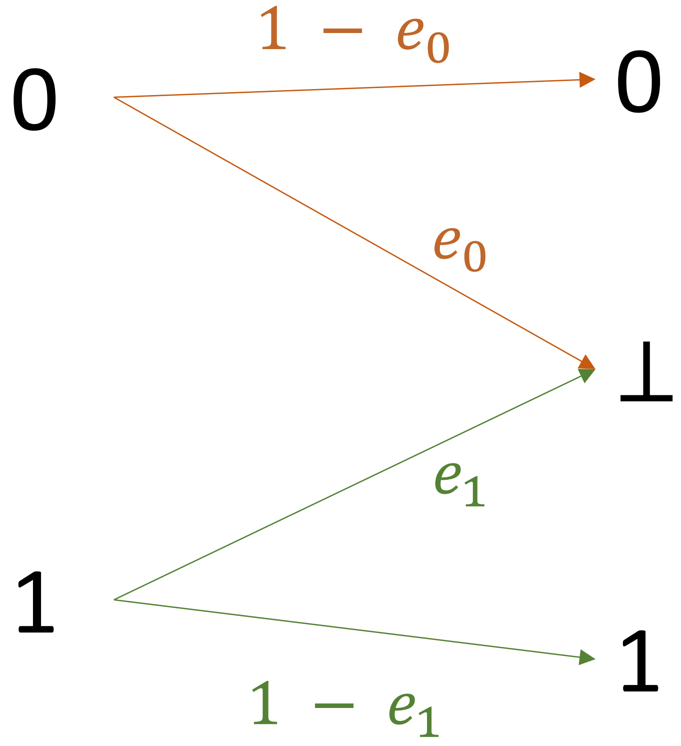
Asymmetric Binary Channels

Binary Asymmetric Channel (BAC)



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

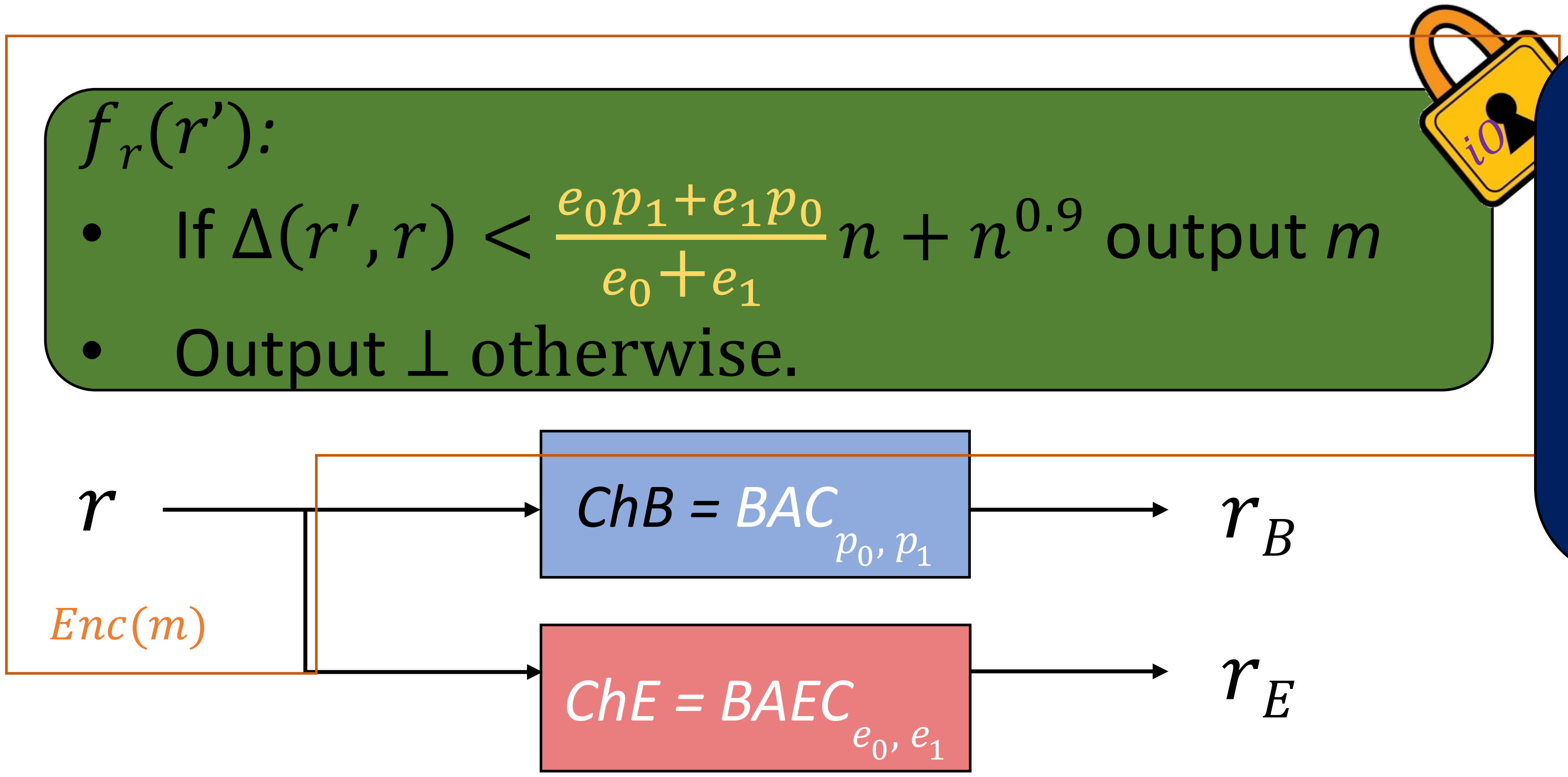
Binary Asymmetric Erasure Channel (BAEC)



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

$$ChB = BAC_{p_0,\, p_1},\quad ChE = BAEC_{e_0,\, e_1}$$

Construction: Same as before, except initial distribution is such that from Eve's view, each erasure equally likely to have been 0 or 1.



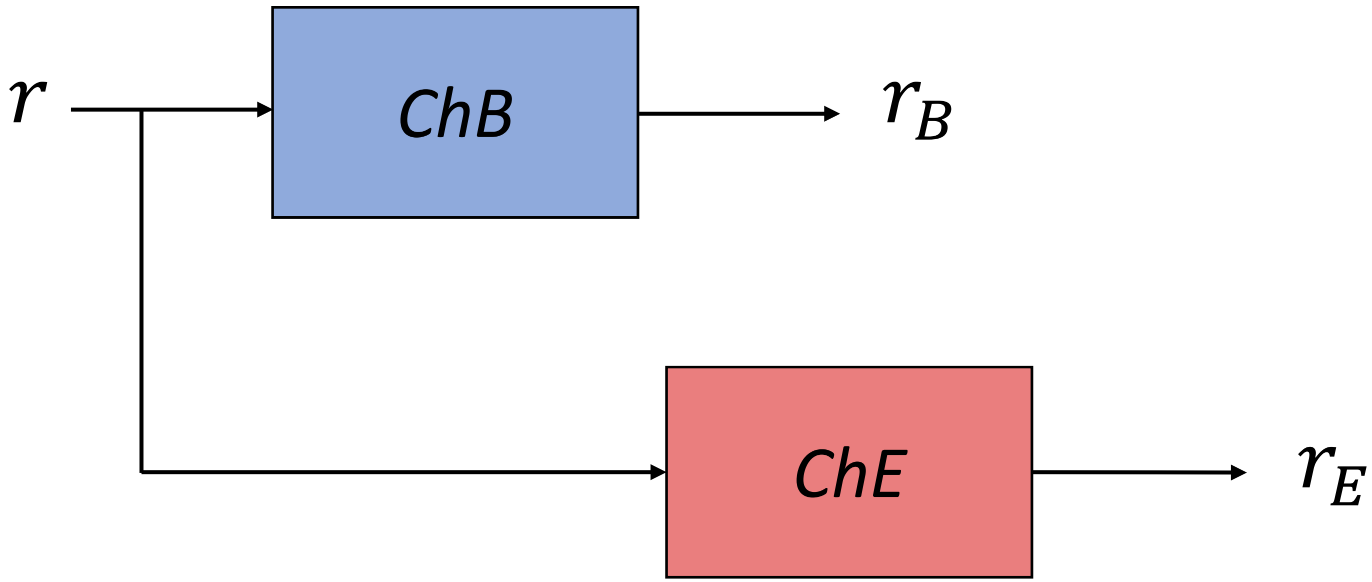
Then proceed by similar hybrid arguments as before.

Turns out, non-degradation condition for this channel pair
$e_0 p_1 + e_1 p_0 < e_0 e_1$
precisely implies Bob has enough advantage to recover m.

Pairs of Binary-input Channels Reduce to the BAC/BAEC Case

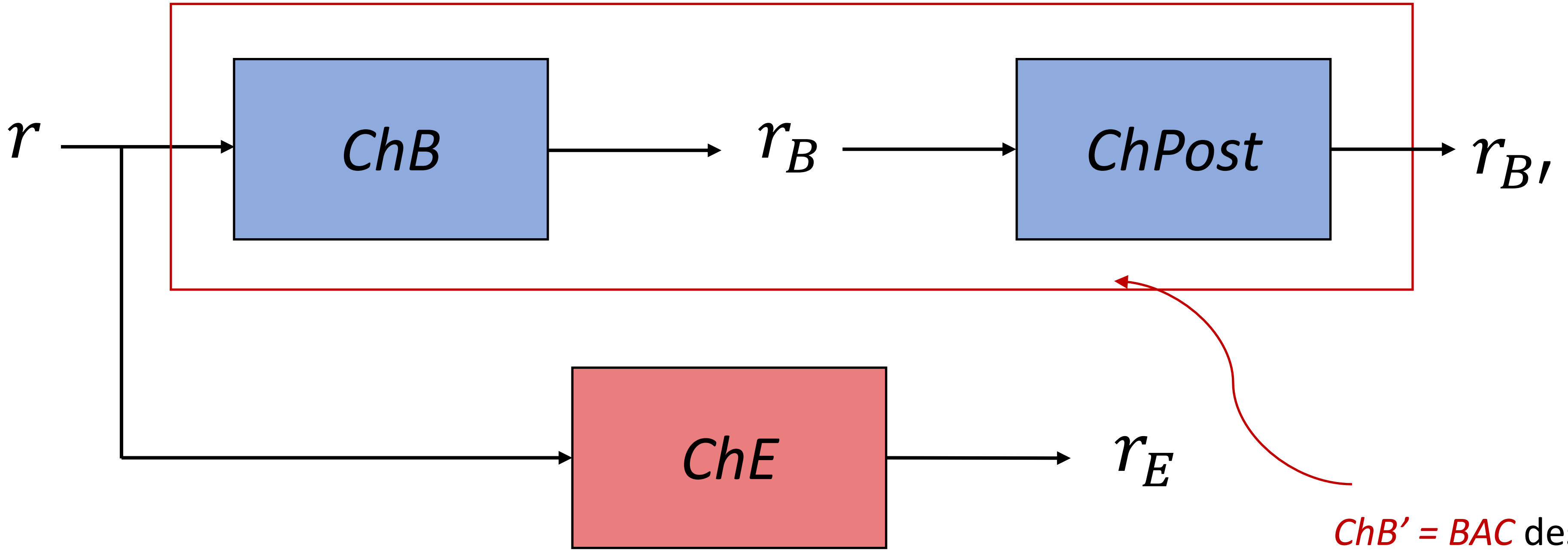
Pair of Arbitrary Binary Input Channels

Consider $\left(B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}, E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}\right)$ s.t. B not a degradation of E.



Reducing Pair of Arbitrary Binary Input Channels to BAC/BAEC Case: Bob's Output Alphabet

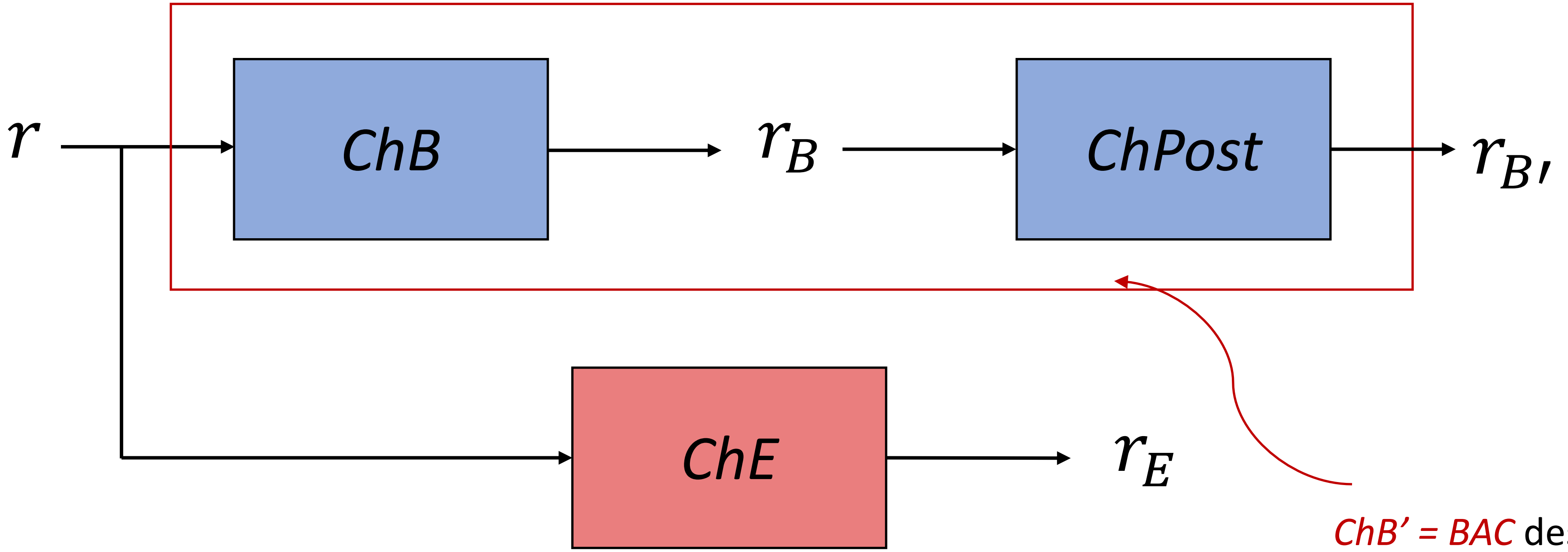
Consider $\left(B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}, E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}\right)$ s.t. B not a degradation of E.



ChB' = BAC described by some matrix B'

Reducing Pair of Arbitrary Binary Input Channels to BAC/BAEC Case: Bob's Output Alphabet

Consider ($B' = \begin{bmatrix} u'_{11} & u'_{12} \\ u'_{21} & u'_{22} \end{bmatrix}$, $E = \begin{bmatrix} v_{11} & \cdots & v_{1n_E} \\ v_{21} & \cdots & v_{2n_E} \end{bmatrix}$) s.t. B not a degradation of E.

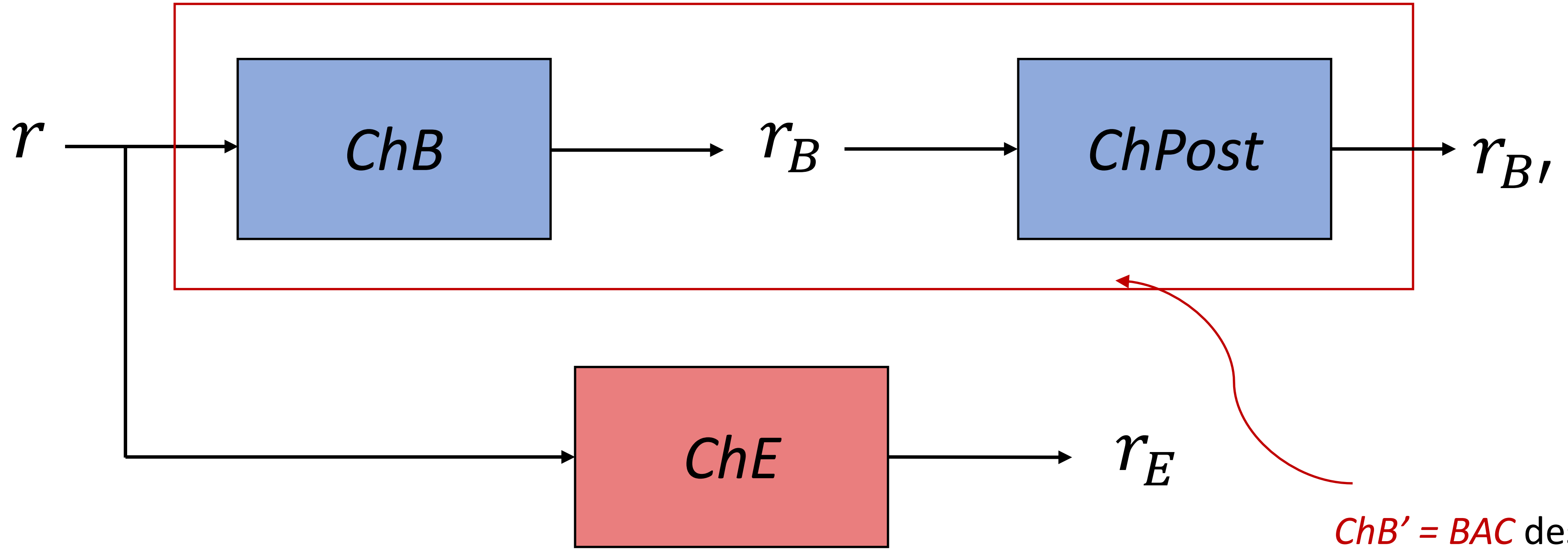


ChB' = BAC described by some matrix B'

Find B' s.t. (1) B' not a degradation of E.
(2) B' degradation of B.

Reducing Pair of Arbitrary Binary Input Channels to BAC/BAEC Case: Bob's Output Alphabet

Consider $\left(B' = \begin{bmatrix} u'_{11} & u'_{12} \\ u'_{21} & u'_{22} \end{bmatrix}, E = \begin{bmatrix} v_{11} & \cdots & v_{1n_E} \\ v_{21} & \cdots & v_{2n_E} \end{bmatrix}\right)$ s.t. B not a degradation of E.



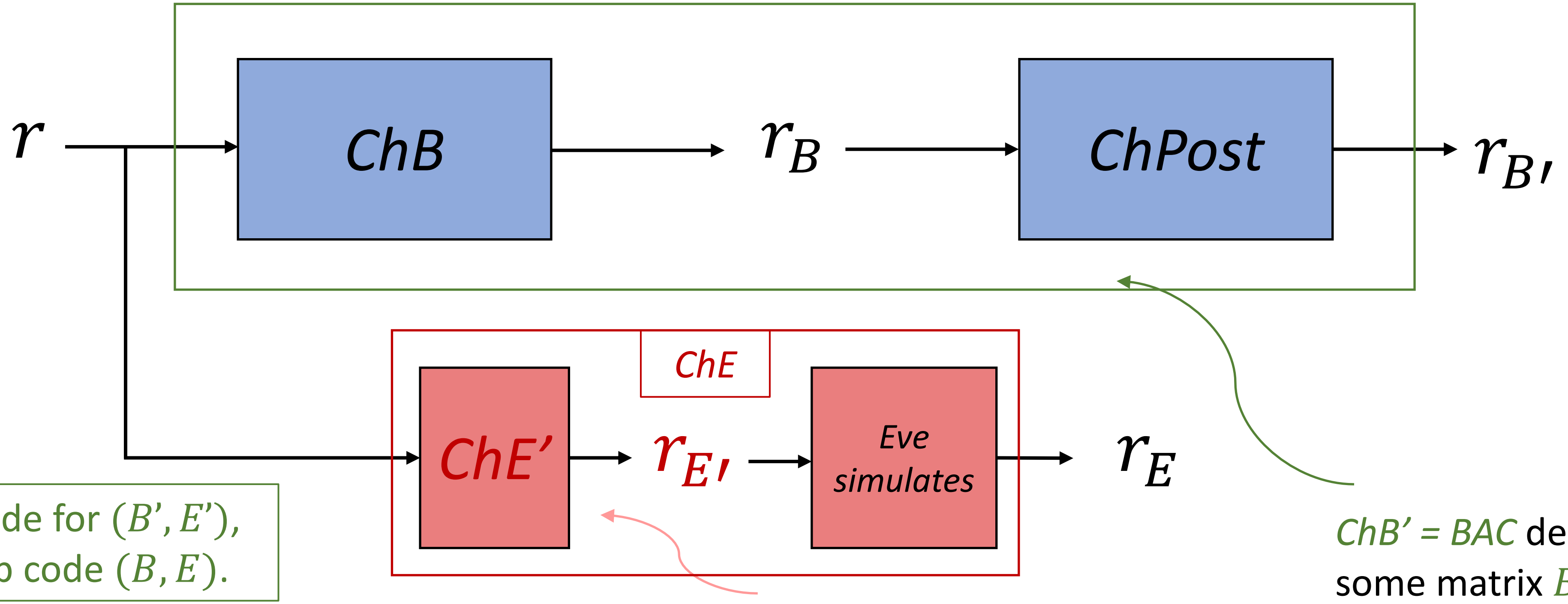
ChB' = BAC described by some matrix B'

Find B' s.t. (1) B' not a degradation of E.
(2) B' degradation of B.

Any wiretap code for (B', E), gives a wiretap code (B, E).

Reducing Pair of Arbitrary Binary Input Channels to BAC/BAEC Case: Simulating ChE with a BAEC

Consider $\left(B' = \begin{bmatrix} u'_{11} & u'_{12} \\ u'_{21} & u'_{22} \end{bmatrix}, E = \begin{bmatrix} v_{11} & \cdots & v_{1n_E} \\ v_{21} & \cdots & v_{2n_E} \end{bmatrix}\right)$ such that $\mathcal{P}(B') \nsubseteq \mathcal{P}(E)$, $\mathcal{P}(B') \subseteq \mathcal{P}(B)$.



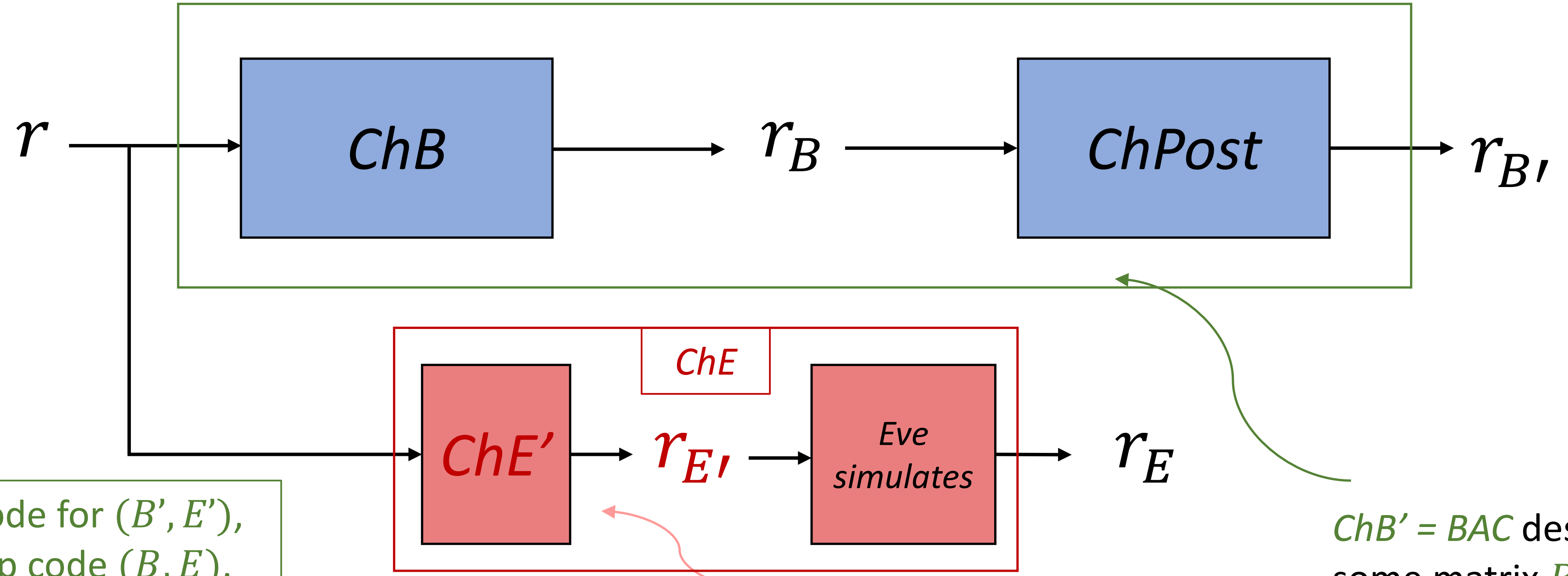
Any wiretap code for (B', E'), gives a wiretap code (B, E).

ChB' = BAC described by some matrix B'

Imagine that Eve instead receives an output through *ChE' = BAEC* described by some matrix E', effectively giving Eve even more information, but hopefully not enough to simulate B'!

Reducing Pair of Arbitrary Binary Input Channels to BAC/BAEC Case: Simulating ChE with a BAEC

Consider $(B' = \begin{bmatrix} u'_{11} & u'_{12} \\ u'_{21} & u'_{22} \end{bmatrix}, E = \begin{bmatrix} v_{11} & \cdots & v_{1n_E} \\ v_{21} & \cdots & v_{2n_E} \end{bmatrix})$ such that $\mathcal{P}(B') \not\subseteq \mathcal{P}(E)$, $\mathcal{P}(B') \subseteq \mathcal{P}(B)$.



Any wiretap code for (B', E'), gives a wiretap code (B, E).

ChB' = BAC described by some matrix B'

Imagine that Eve instead receives an output through *ChE' = BAEC* described by some matrix E', effectively giving Eve even more information, but hopefully not enough to simulate B'!

Finding BAEC E' via Polytope Formulation

A New Polytope formulation

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

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

A New Polytope formulation

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

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

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

A New Polytope formulation

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

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

Theorem: Let $B \in \mathbb{R}^{2\times n_B}$ and E $\in \mathbb{R}^{2\times n_E}$ be arbitrary row-stochastic matrices. Then, $\mathrm{Ch}B$ is not a degradation of $\mathrm{Ch}E$ if and only if $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.

A New Polytope formulation

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

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

In the interest of time, we will not sketch the proof.

If row count > 2, then this is false. Explicit counterexample for case of 3.

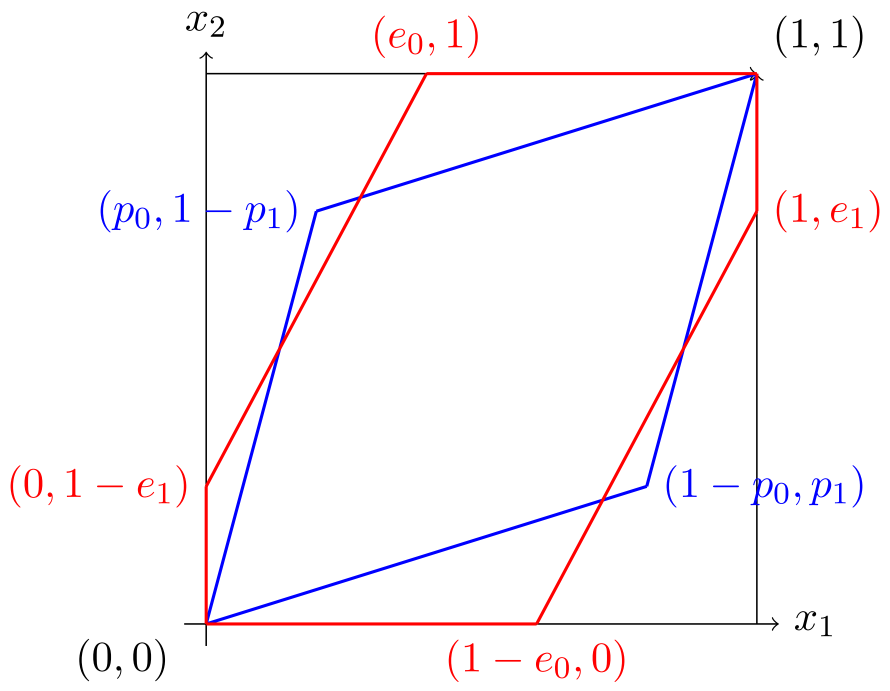
Theorem: Let $B \in \mathbb{R}^{2 \times n_B}$ and $E \in \mathbb{R}^{2 \times n_E}$ be arbitrary row-stochastic matrices. Then, $\mathrm{Ch}B$ is not a degradation of $\mathrm{Ch}E$ if and only if $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.

Polytope Example

Binary Asymmetric Erasure Channel (BAEC)

$$\begin{bmatrix} 1-p_0 & p_0 \\ p_1 & 1-p_1 \end{bmatrix} \qquad \begin{bmatrix} 1-e_0 & 0 & e_0 \\ 0 & 1-e_1 & e_1 \end{bmatrix}$$

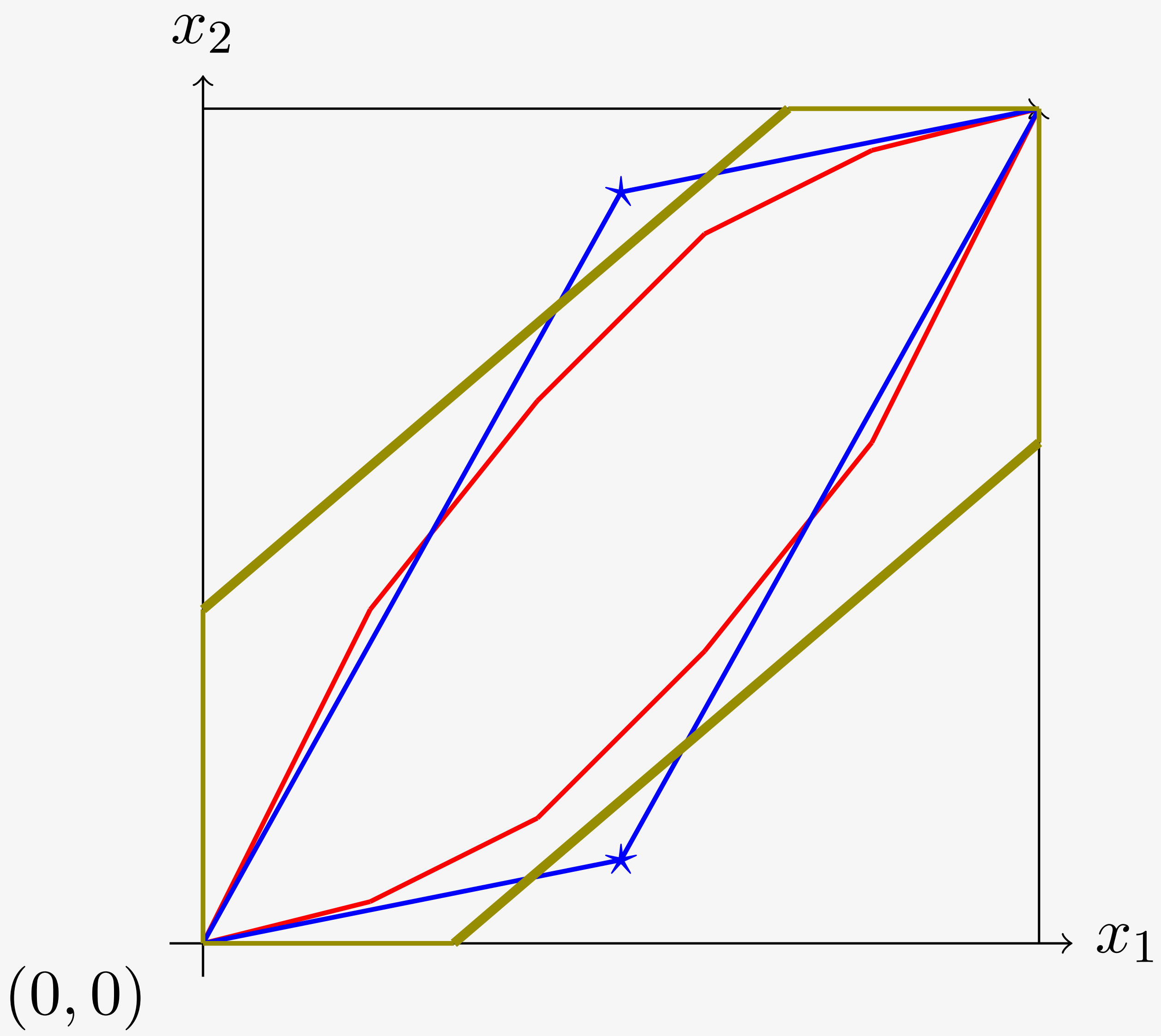
Binary Asymmetric Channel (BAC)



The blue polytope corresponds to the BAC.

The red polytope corresponds to the BAEC.

Since the blue polytope is **not** contained in the red polytope, the BAC channel is **not** a degradation of the BAEC channel.

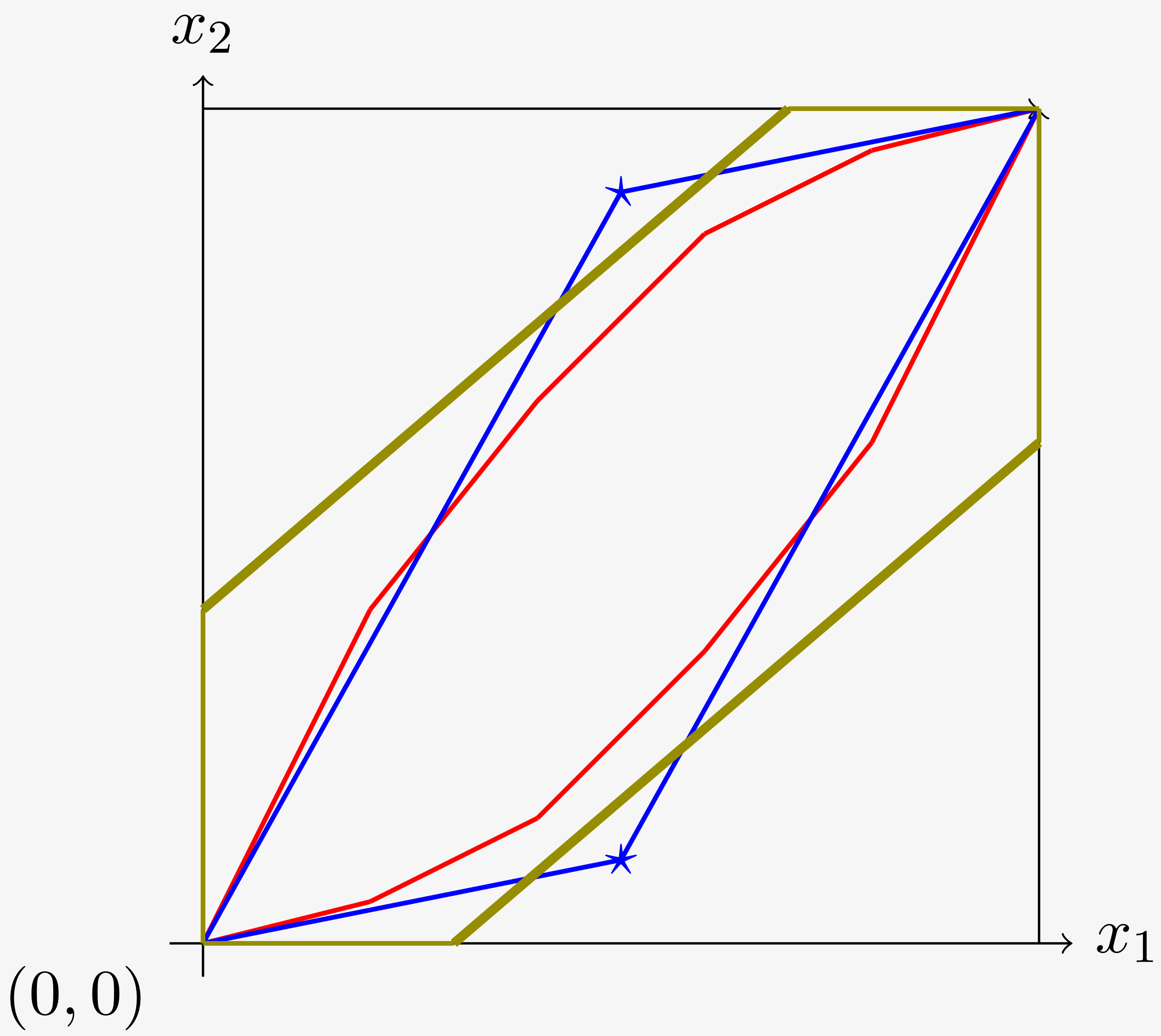


Reducing Eve's Channel to a BAEC

The blue polytope corresponds to the BAC.

The red polytope corresponds to some channel ChE.

Since the blue polytope is **not** contained in the red polytope, the BAC channel is **not** a degradation of ChE.



Reducing Eve's Channel to a BAEC

Apply the strict separating hyperplane theorem!

Take an extreme point of the BAC **not** inside the ChE polytope and separate it from the ChE polytope.

Olive polytope is a BAEC channel s.t. (1) ChE is a degradation and (2) ChB is not a degradation.

Can find this polytope efficiently.