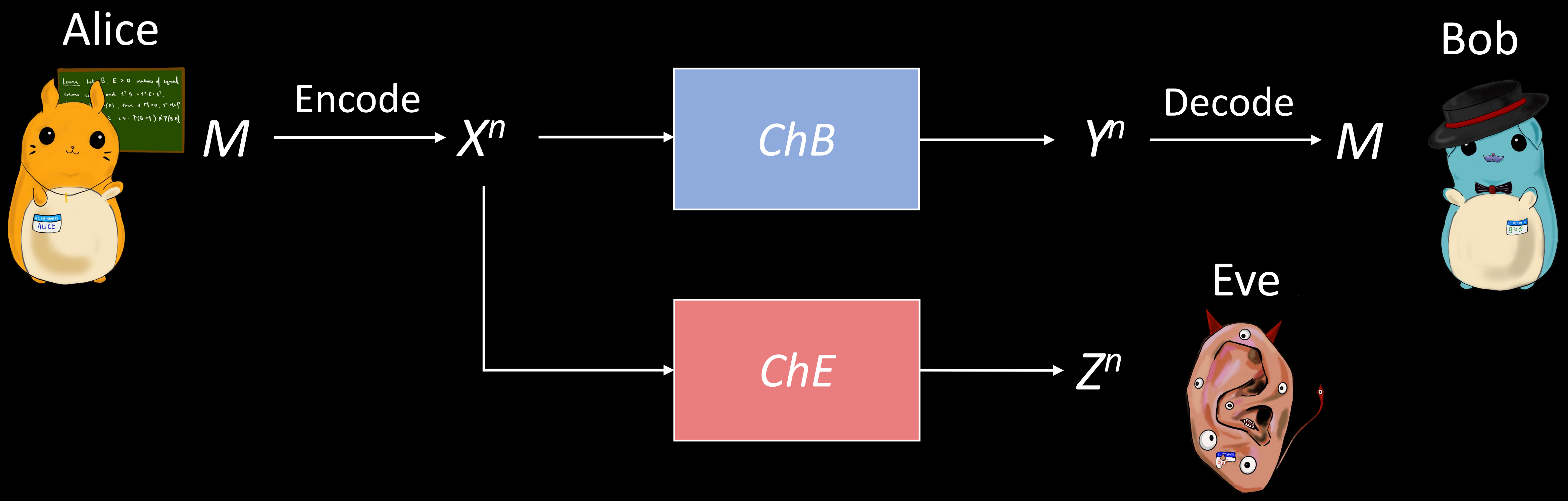
Computational Wiretap Coding via Obfuscation

Paul Lou
UCLA

Based on joint works with Yuval Ishai, Aayush Jain, Alexis Korb, Amit Sahai, & Mark Zhandry
[IKLS22, IJLSZ22]

Wiretap Channel [Wyn75]



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

Wiretap Channel [Wyn75]

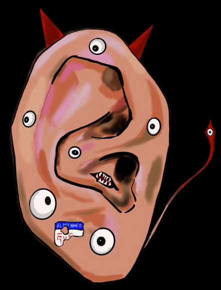
Alice

$M \xrightarrow{\text{Encode}} X^n \rightarrow$ *ChB* $\rightarrow Y^n \xrightarrow{\text{Decode}} M$

Bob

$X^n \rightarrow$ *ChE* $\rightarrow Z^n$

Eve



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

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

Formal Definition (Statistical)

Def: *(Enc, Dec)* is a statistically secure wiretap coding scheme for wiretap channel *(ChB, ChE)* if

- **Correctness:** For all messages $m \in \{0, 1\}$,

  $$\Pr\left[Dec\left(1^\lambda, ChB\left(Enc(1^\lambda, m)\right)\right) = m\right] \geq 1 - negl(\lambda)$$

- **Security:** For all adversaries *A*,

  $$\Pr\left[A\left(1^\lambda, ChE\left(Enc(1^\lambda, b)\right)\right) = b\right] \leq \frac{1}{2} + negl(\lambda)$$

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

Formal Definition (Computational)

Def: *(Enc, Dec)* is a statistically (resp. computationally) secure wiretap coding scheme for wiretap channel *(ChB, ChE)* if

- **Correctness:** For all messages $m \in \{0,1\}$,
  $$\Pr\left[Dec\left(1^\lambda, ChB\left(Enc(1^\lambda, m)\right)\right) = m\right] \geq 1 - negl(\lambda)$$
- **Security:** For all (resp. non-uniform polynomial-time) adversaries *A*,
  $$\Pr\left[A\left(1^\lambda, ChE\left(Enc(1^\lambda, b)\right)\right) = b\right] \leq \frac{1}{2} + negl(\lambda)$$
  where b is uniformly distributed over $\{0,1\}$.
- (Computational Definition Only): *(Enc, Dec)* are PPT algorithms.

Formal Definition (Computational)

Def: *(Enc, Dec)* is a statistically (resp. computationally) <u>secure wiretap coding scheme</u> for wiretap channel *(ChB, ChE)* if

- **Correctness:** For all messages $m \in \{0,1\}$,

  $$\Pr\left[Dec\left(1^\lambda, ChB\left(Enc(1^\lambda, m)\right)\right) = m\right] \geq 1 - negl(\lambda)$$

- **Security:** For all (resp. non-uniform polynomial-time) adversaries *A*,

  $$\Pr\left[A\left(1^\lambda, ChE\left(Enc(1^\lambda, b)\right)\right) = b\right] \leq \frac{1}{2} + negl(\lambda)$$

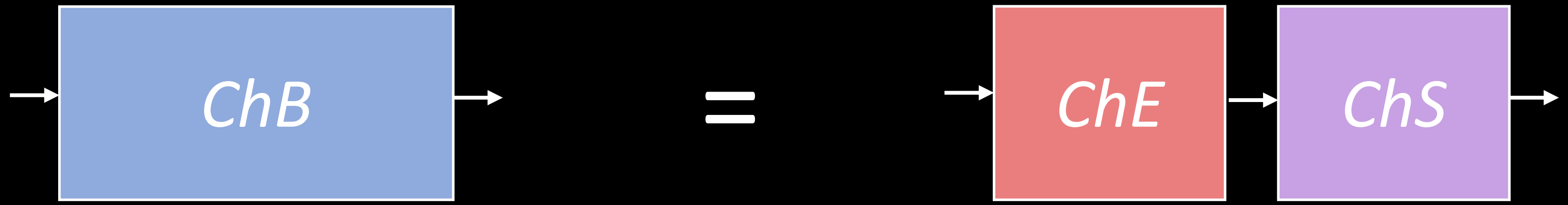
  where b is uniformly distributed over $\{0,1\}$.

- (Computational Definition Only): *(Enc, Dec)* are PPT algorithms.

Our results also generalize to larger message spaces.

Simple Impossibility

Def: *ChB* is a <u>degradation</u> of *ChE* if there exists a channel *ChS* such that

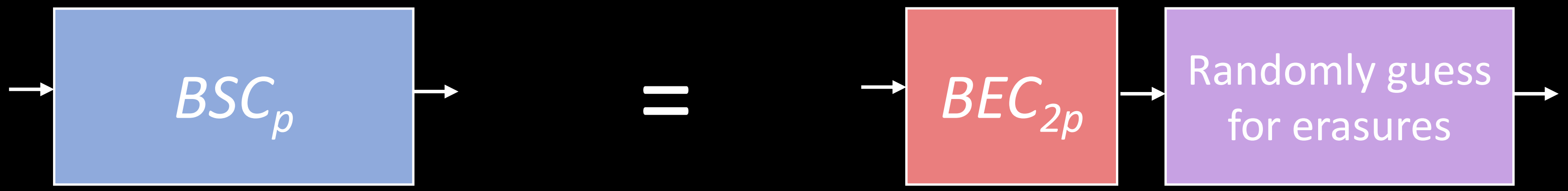


Observation: In this case, Eve can learn the same distribution Bob learns, so wiretap coding is impossible.

Simple Impossibility

Def: *ChB* is a <u>degradation</u> of *ChE* if there exists a channel *ChS* such that

ex)



Observation: In this case, Eve can learn the same distribution Bob learns, so wiretap coding is impossible.

Information Theoretic Setting

Can we create a wiretap coding scheme whenever *ChB* is not a degradation of *ChE*?

Information Theoretic Setting

Can we create a wiretap coding scheme whenever *ChB* is not a degradation of *ChE*?

No!

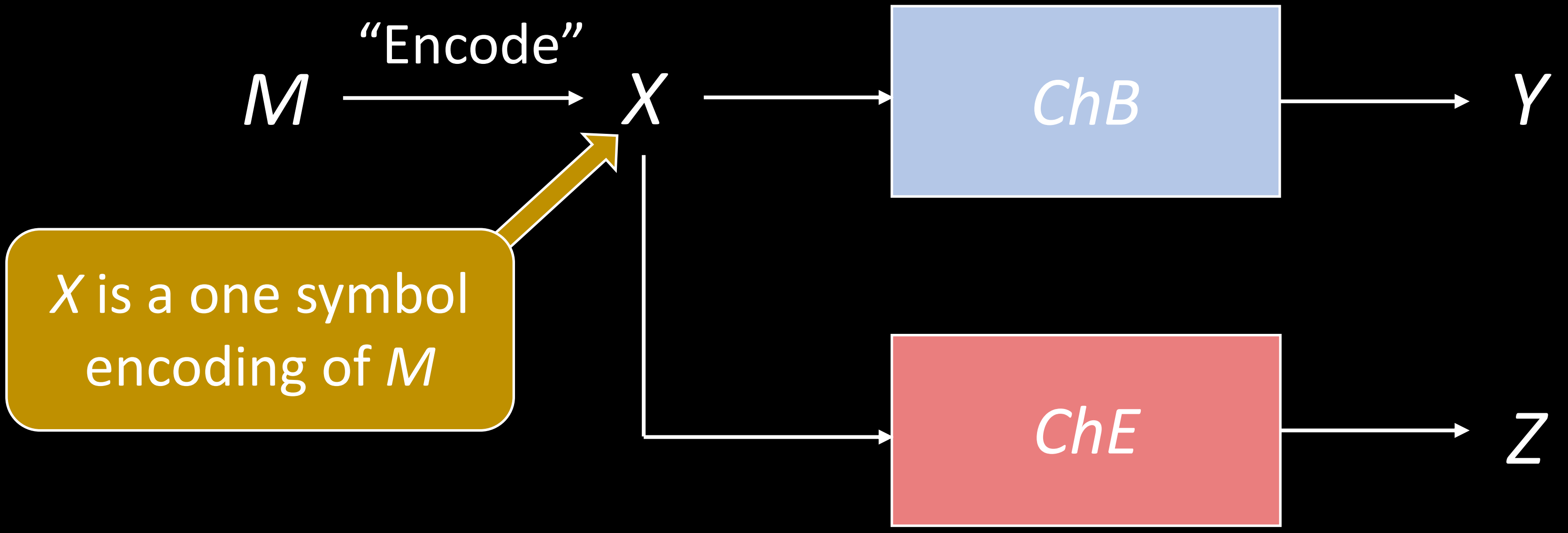
[CK78] Wiretap coding schemes are possible if and only if *ChE* is not less noisy than *ChB*.

(Not) Less Noisy [CK78]

Def: *ChE* is not <u>less noisy</u> than *ChB* if there exists a Markov chain $M \rightarrow X \rightarrow YZ$ where $p_{Y|X}(y|x)$ corresponds to *ChB*, $p_{Z|X}(z|x)$ corresponds to *ChE*, and

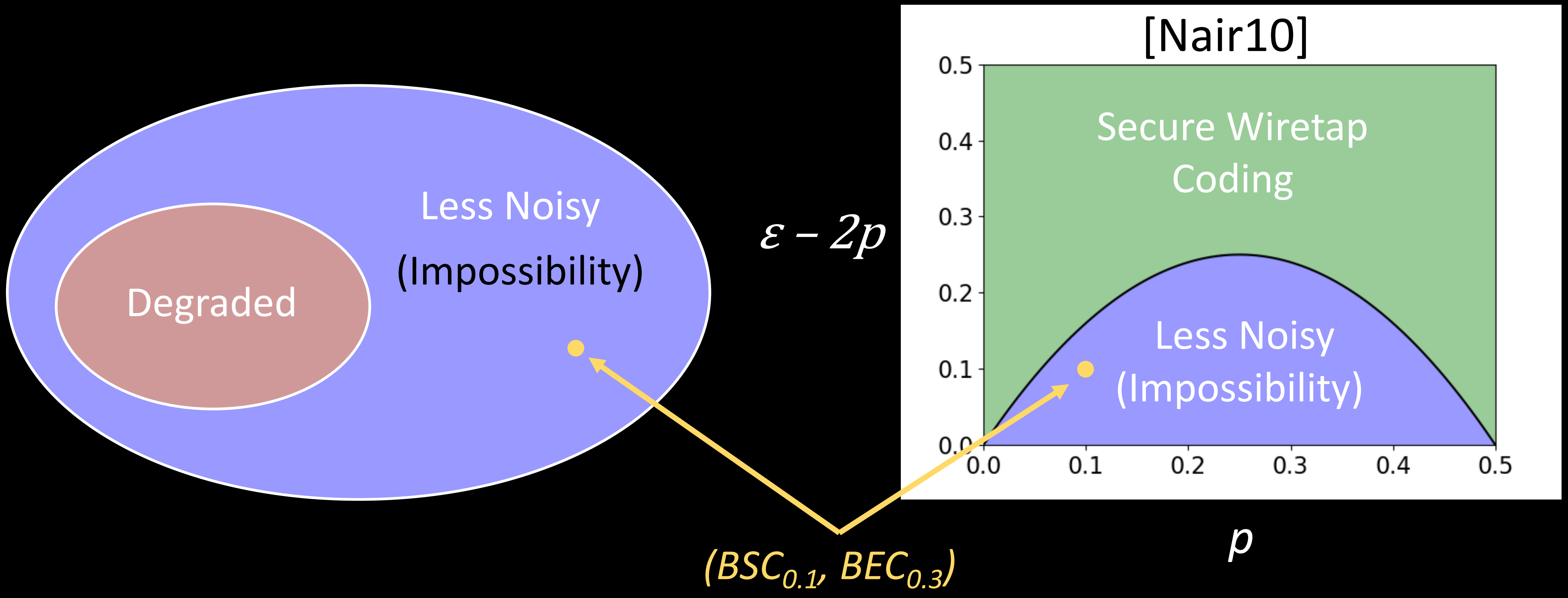
$$H(M \mid Y) < H(M \mid Z)$$

Bob has an advantage over Eve in terms of conditional entropy.



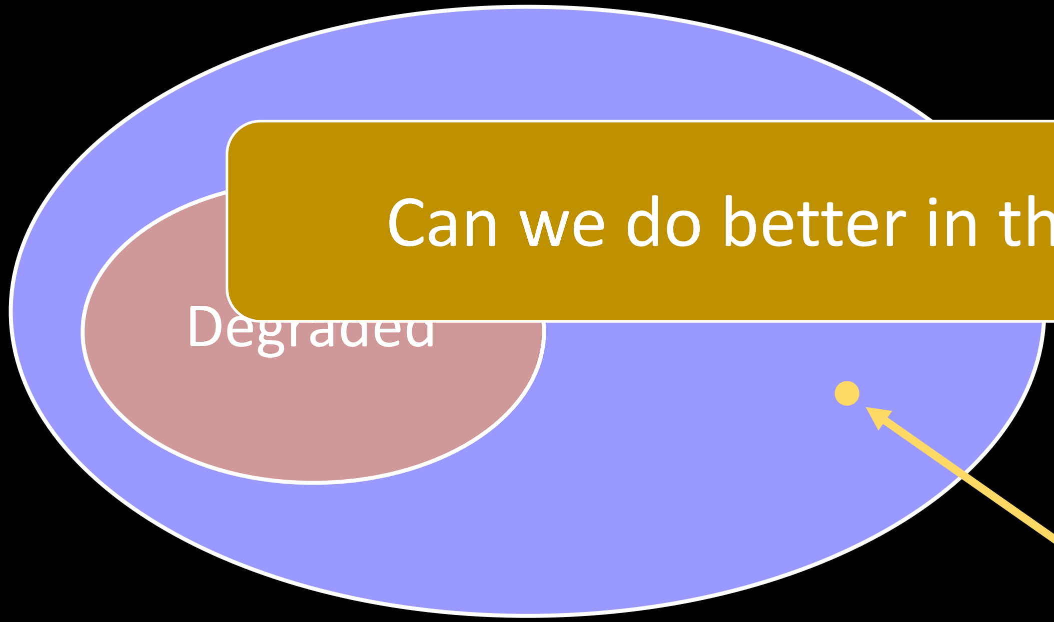
Information Theoretic Impossibility

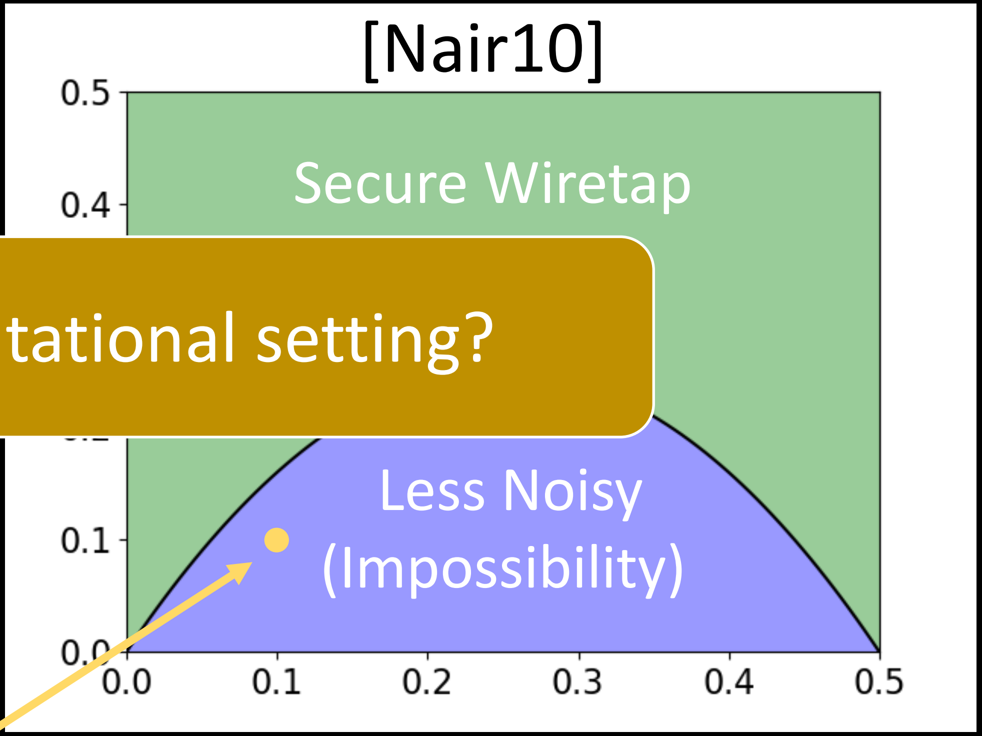
ex) $ChB = BSC_p$ $ChE = BEC_\varepsilon$



Information Theoretic Impossibility

ex) $ChB = BSC_p$ $ChE = BEC_\varepsilon$





Can we do better in the computational setting?

p

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

Computational Assumptions and Feasibility Results

	Information Theoretic	Computational
Secure Encryption	key length $\geq$ message length [Shannon1949]	Fixed key length, unlimited messages (1970s)
Secure Multi-Party Computation	Honest majority of parties needed [BGW88,CCD88]	Only need one honest party [GMW87]
Secure Wiretap Coding Schemes	Introduced [Wyner75], "Less Noisy" characterization [CK78]	**OPEN** Until our paper [IKLS22] in 2022, no improvement

Computational Setting

Can we create a wiretap coding scheme whenever *ChB* is not a <u>degradation</u> of *ChE*?

Recall: Impossible (even computationally) if *ChB* is a degradation of *ChE*.

Computational Setting

Can we create a wiretap coding scheme whenever
ChB is not a degradation of *ChE*?

Yes!

Our Work [IKLS22]: Assuming secure evasive function obfuscation for the class of generalized fuzzy point functions,

wiretap coding schemes are possible if and only if
ChB is not a degradation of *ChE*.

Computational Setting

Can we create a wiretap coding scheme whenever

Follow-up [IJLSZ22]: Assuming indistinguishability obfuscation and injective PRGs, for binary input channel pairs *(ChB, ChE):*

Computational wiretap coding schemes are possible if and only if *ChB* is not a degradation of *ChE.*

Our W n for

the cl

wiretap coding schemes are possible if and only if

ChB is not a degradation of ChE.

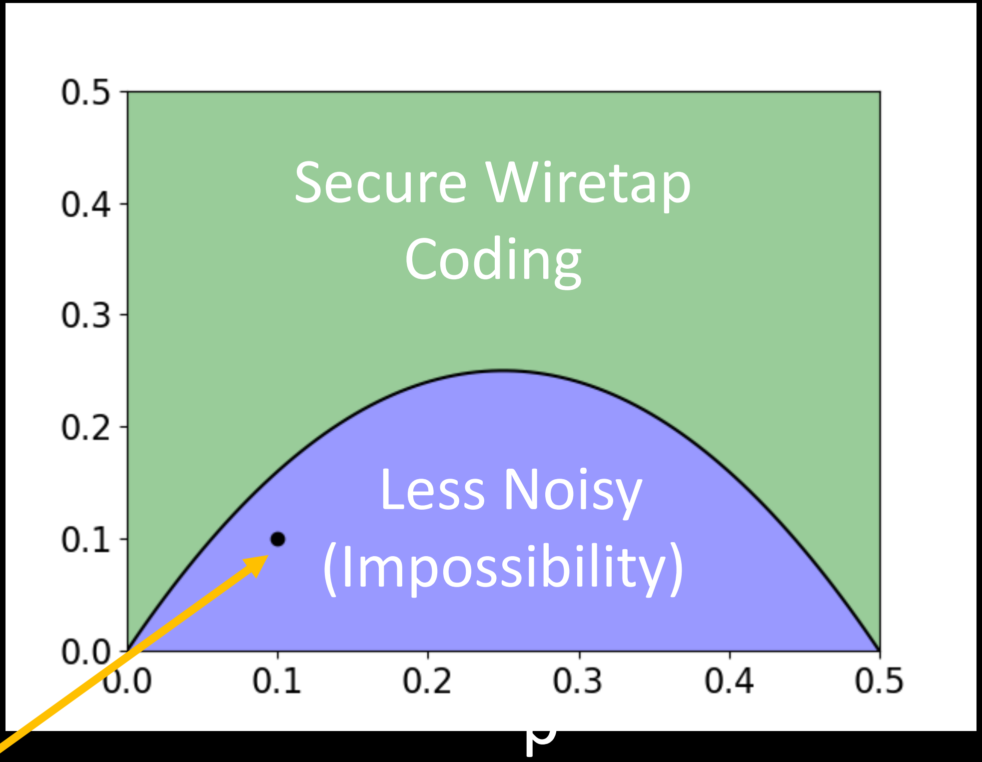
Construction of Wiretap Coding Schemes via Program Obfuscation

Based on joint work with Yuval Ishai, Alexis Korb, Amit Sahai [IKLS22]

Starting Point: Example

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

$\varepsilon - 2p$

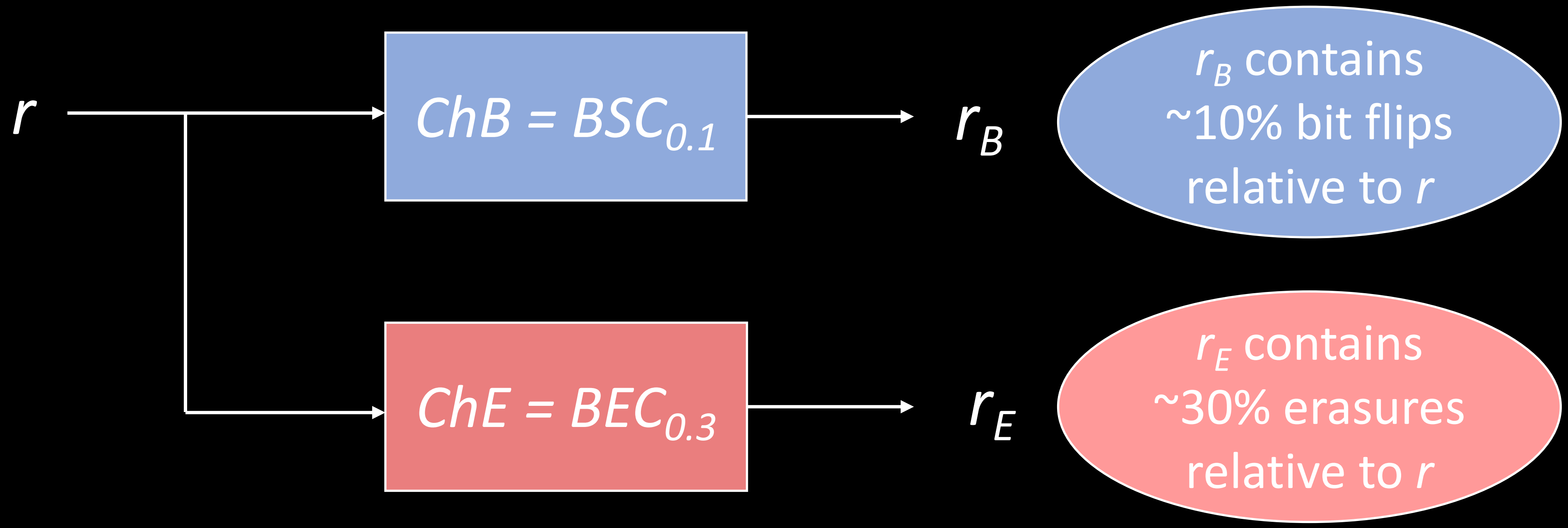


($BSC_{0.1}$, $BEC_{0.3}$)

ChB is not a degradation of *ChE*.

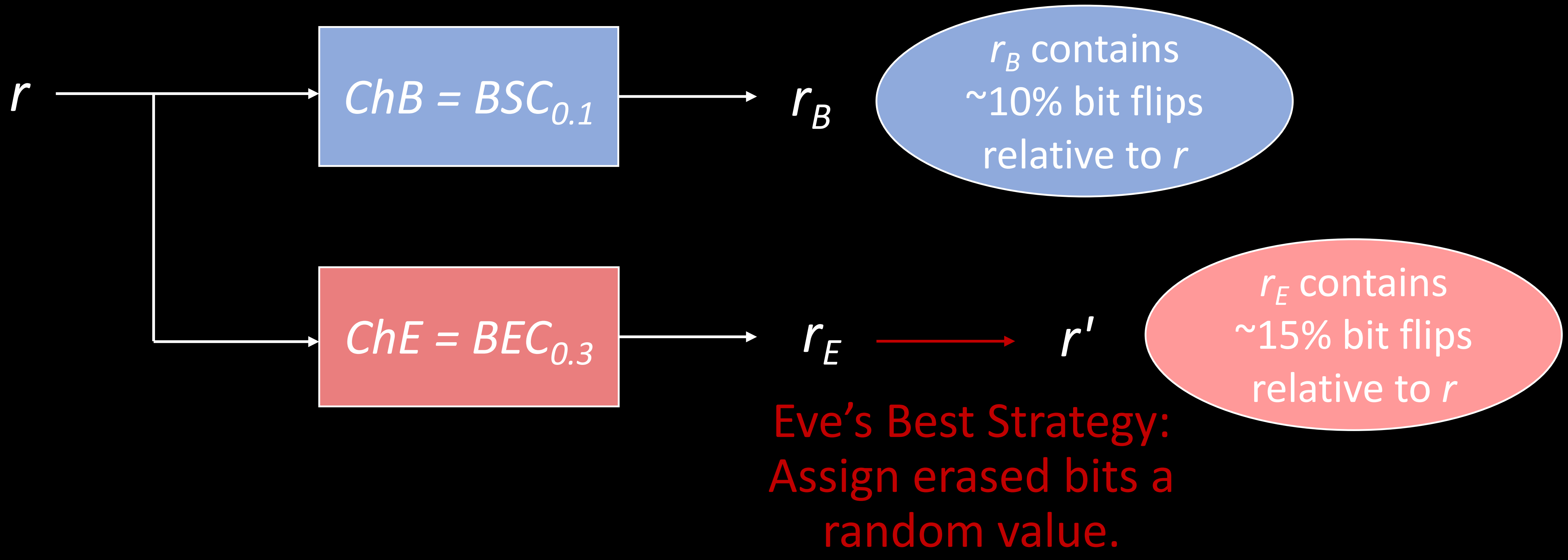
ChE is less noisy than *ChB* [Nair10] (and hence wiretap coding impossible information theoretically [CK78])

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



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

Observation: If $r \in \{0,1\}^n$ is uniformly random, then w.h.p. Eve cannot find a string that contains ~10% bit flips relative to r.

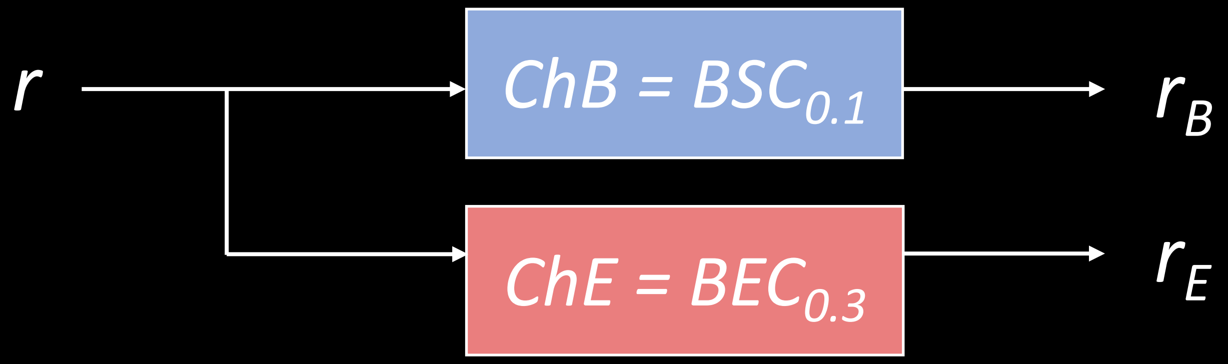


Example: $ChB = BSC_{0.1}$, $ChE = BEC_{0.3}$

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

- Output m if r' contains ~10% bit flips relative to r.
- Output $\perp$ otherwise.



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

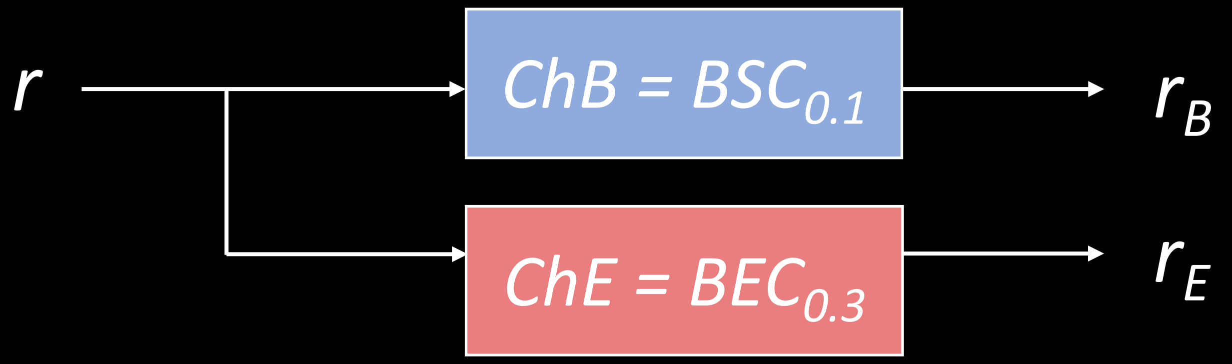
Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

- Output m if r' contains ~10% bit flips relative to r.
- Output $\perp$ otherwise.

Correctness:

$f_r(r_B) = m$ with high probability

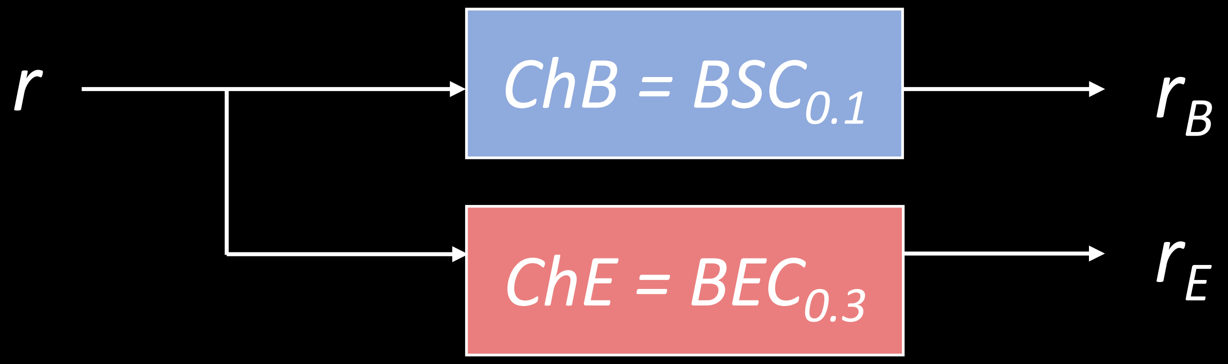


Example: $ChB = BSC_{0.1}$, $ChE = BEC_{0.3}$

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

- Output m if r' contains ~10% bit flips relative to r.
- Output $\perp$ otherwise.



Correctness:

$f_r(r_B) = m$ with high probability

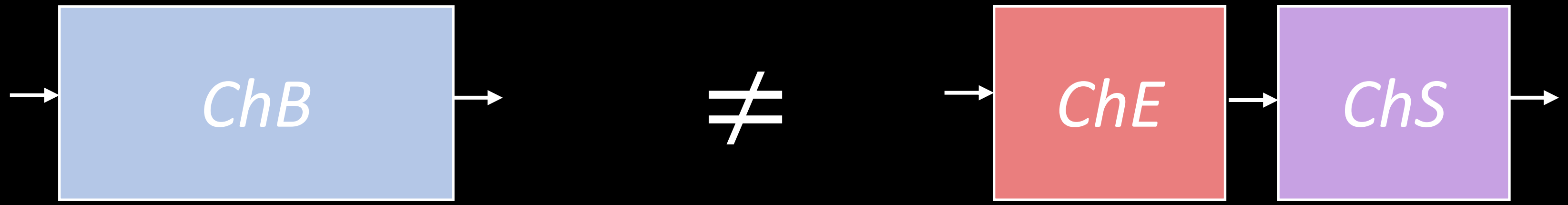
Security:

- W.h.p. Eve cannot find an r' such that $f_r(r') = m$.
- Obfuscation hides value of m in this case.

General Case

General Case: Not Degraded

Def: *ChB* is <u>not a degradation</u> of *ChE* if for all channels *ChS* we have:



For every *ChS*, there exists *(x*, y*)* such that

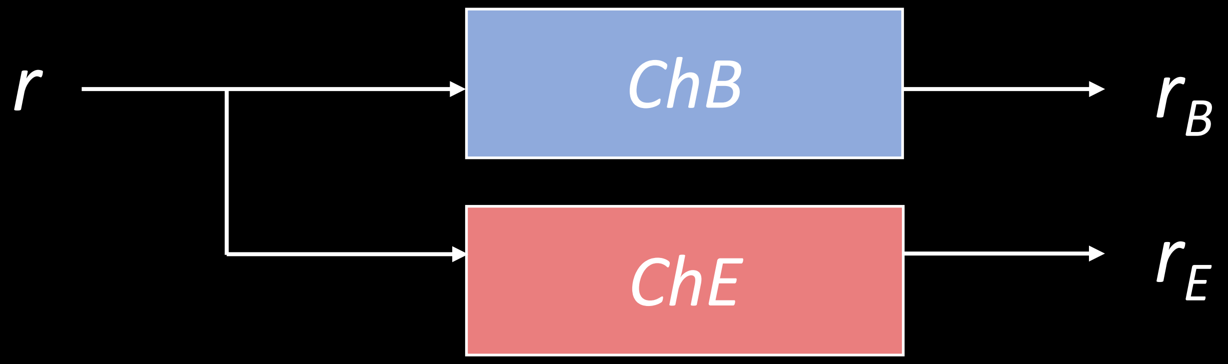
$$|Pr[ChB(x^*) = y^*] - Pr[ChS(ChE(x^*)) = y^*]| > 0$$

In fact, we can show the difference is at least some constant dependent on *ChB* and *ChE*.

General Case: Not Degraded

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:
- Output m if for all (x,y), $|\{i \in [n]: r_i = x \; and \; r'_i = y\}|$ ~ as expected for an $r' = ChB(r)$.
- Output $\perp$ otherwise.

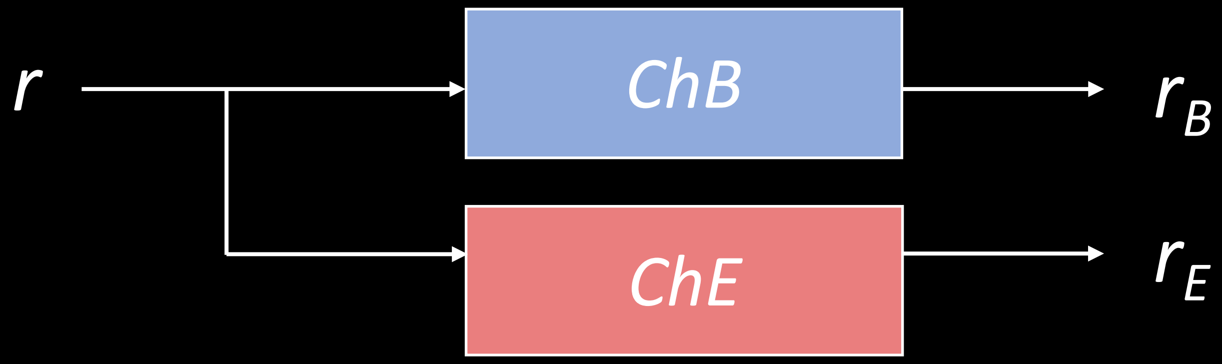


General Case: Not Degraded

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:
- Output m if for all (x,y), $|\{i \in [n]: r_i = x \text{ and } r'_i = y\}|$ ~ as expected for an $r' = ChB(r)$.
- Output $\perp$ otherwise.

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

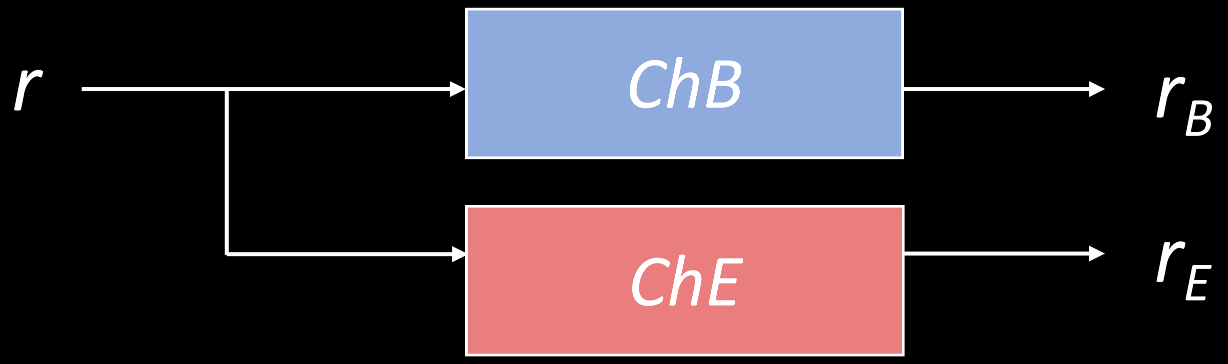


General Case: Not Degraded

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

- Output m if for all (x,y), $|\{i \in [n]: r_i = x \text{ and } r'_i = y\}|$ ~ as expected for an $r' = ChB(r)$.
- Output $\perp$ otherwise.



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

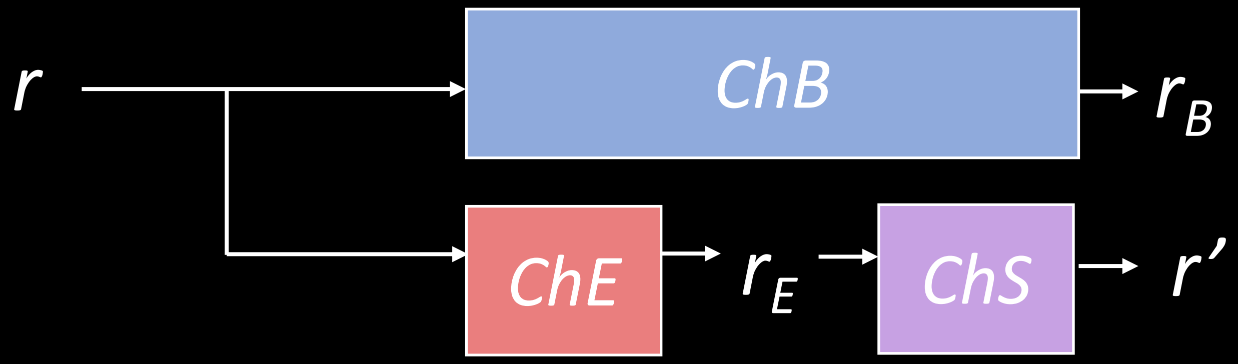
Security:
???

General Case: Not Degraded

Observation: If Eve's strategy for finding inputs *r'* for f_r is to apply a DMC *ChS* to r_E, then we can prove security.

$f_r(r')$:

- Output *m* if for all *(x,y)*,
  $|\{i \in [n]: r_i = x \text{ and } r'_i = y\}|$
  ~ as expected for an *r' = ChB(r)*.
- Output ⊥ otherwise.

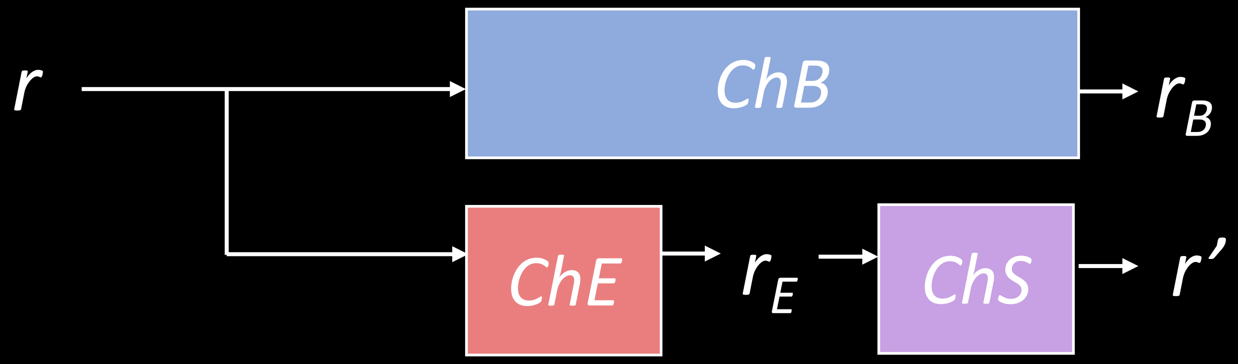


General Case: Not Degraded

Observation: If Eve's strategy for finding inputs *r'* for f_r is to apply a DMC *ChS* to r_E, then we can prove security.

$f_r(r')$:

- Output *m* if for all *(x,y)*, $|\{i \in [n]: r_i = x \; and \; r'_i = y\}|$ ~ as expected for an $r' = ChB(r)$.
- Output $\perp$ otherwise.



1) Since ChB is not a degradation of ChE, there exists (x^*, y^*) such that $\Pr[ChS(ChE(x^*)) = y^*]$ differs from $\Pr[ChB(x^*) = y^*]$.

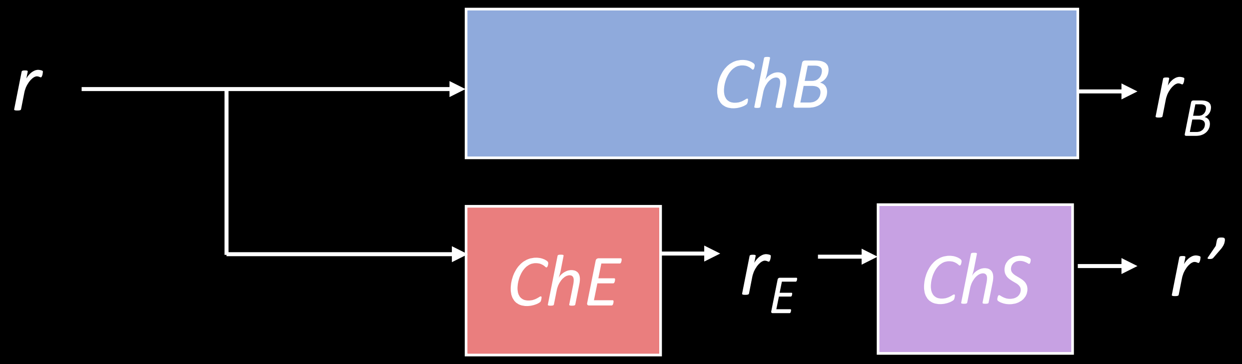
2) Thus, w.h.p., $f_r(r') = \perp$ as the check fails on (x^*, y^*).

l Case: Not Degraded

Observation: If Eve's strategy for finding inputs r' for f_r is to apply a DMC ChS to r_E, then we can prove security.

$f_r(r')$:

- Output m if for all (x,y), $|\{i \in [n]: r_i = x \ and \ r'_i = y\}|$ ~ as expected for an $r' = ChB(r)$.
- Output $\perp$ otherwise.



1) Since ChB is not a degradation of ChE, there exists (x^*, y^*) such that $\Pr[ChS(ChE(x^*)) = y^*]$ differs from $\Pr[ChB(x^*) = y^*]$.

2) Thus, w.h.p., $f_r(r') = \perp$ as the check fails on (x^*, y^*).

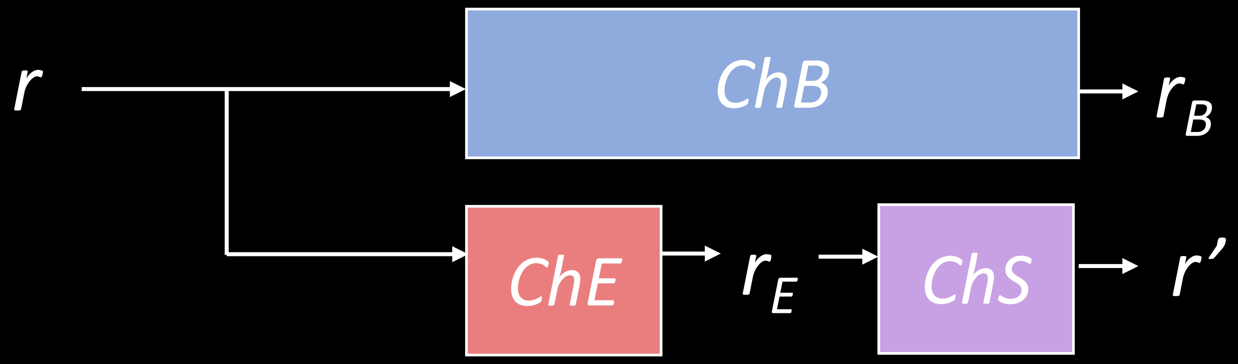
l Case: Not Degraded

Observation: If Eve's strategy for finding inputs r' for f_r is to apply a DMC ChS to r_E, then we can prove security.

$f_r(r')$:

- Output m if for all (x,y), $|\{i \in [n]: r_i = x \text{ and } r'_i = y\}|$ ~ as expected for an $r' = ChB(r)$.
- Output $\perp$ otherwise.

3) Obfuscation hides m in this case.



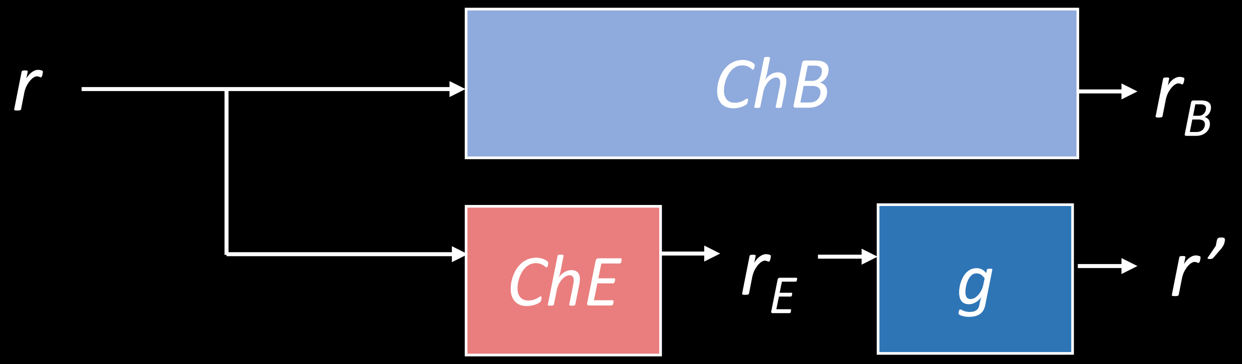
1) Since ChB is not a degradation of ChE, there exists (x^*, y^*) such that $\Pr[ChS(ChE(x^*)) = y^*]$ differs from $\Pr[ChB(x^*) = y^*]$.

General Case: Not Degraded

Observation: If Eve's strategy for finding inputs r' for f_r is to apply a DMC ChS to r_E, then we can prove security.

f_r(r'):

- Output *m* if for all *(x,y)*, $|\{i \in [n]: r_i = x \text{ and } r'_i = y\}|$ ~ as expected for an *r' = ChB(r)*.
- Output ⊥ otherwise.



Issue: Eve can use any arbitrary strategy *g* (not necessarily a DMC) to find *r'*!

Proving Security

Goal: Show that for any strategy g, there exists a DMC ChS and a polynomial p such that

$$\Pr\left[f_r\left(g(r_E)\right) = m\right] \leq \mathrm{p}(\mathrm{n}) \cdot \Pr\left[f_r\left(ChS(r_E)\right) = m\right] + negl(n)$$

Eve cannot do much better by using g than by using ChS!
This gives us security!

We show this via a hybrid argument.

$i\mathcal{O}$-based Construction of Computational Wiretap Coding Schemes for Binary Input Channels

Based on joint work with Yuval Ishai, Aayush Jain, Amit Sahai, Mark Zhandry [IJLSZ22]

Construction Road Map

1. The setting of the binary **asymmetric** channels (**BAC**) and binary **asymmetric** erasure channels (**BAEC**): an iO + injective PRG based construction.

2. Polytope formulation of degradation

3. Reducing constructing a computational wiretap coding scheme for any pair of binary input channels to the asymmetric case.

Indistinguishability Obfuscation (iO) [BGIRSVY01]

A secure indistinguishability obfuscation (iO) scheme satisfies (informally)

- **Completeness:**

$$C \equiv iO(C)$$

- **Indistinguishability:** Circuits C_0 and C_1 of *<u>same size</u>*, *<u>same input length</u>*, *<u>same output length</u>*, *and <u>functionally equivalent</u>* satisfy:

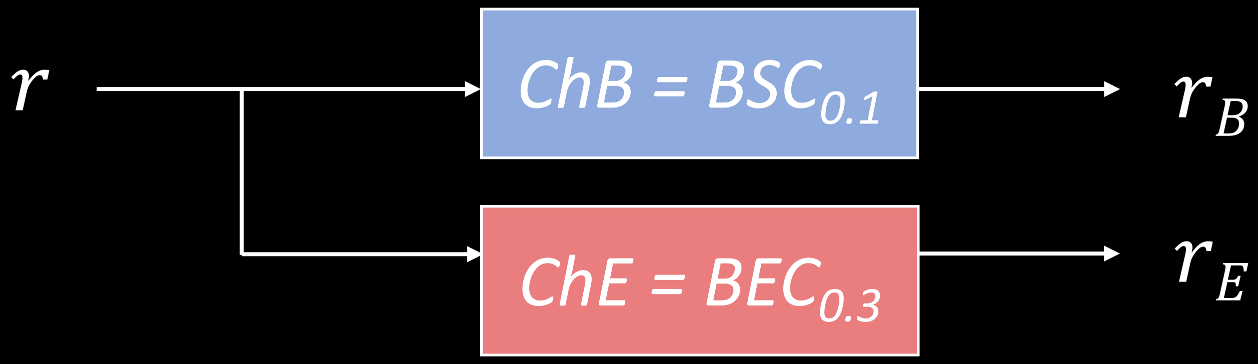
$$iO(C_0) \approx_c iO(C_1)$$

Warm-up: $ChB = BSC_{0.1}$, $ChE = BEC_{0.3}$

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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

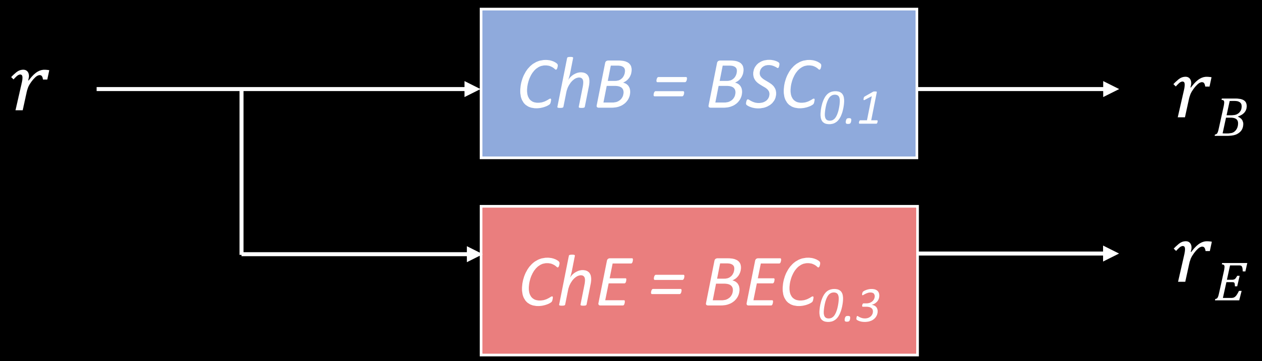


Warm-up: $ChB = BSC_{0.1}$, $ChE = BEC_{0.3}$

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

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

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

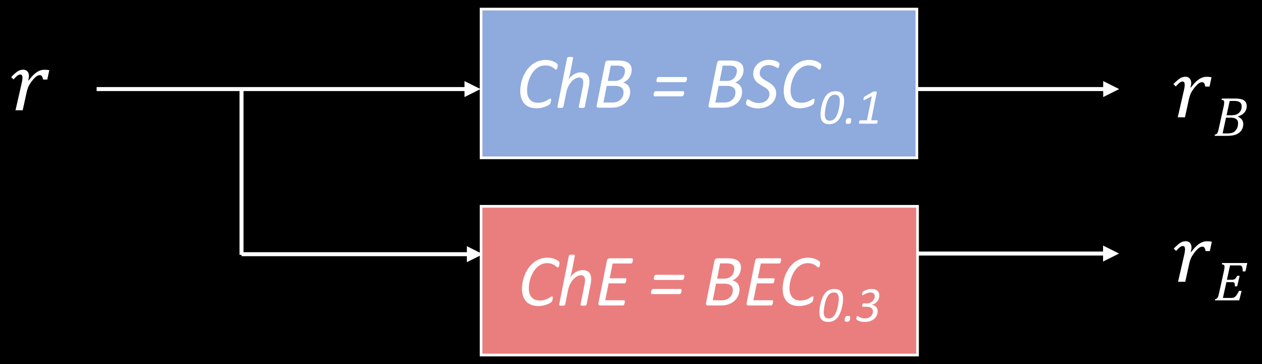


Warm-up: $ChB = BSC_{0.1}$, $ChE = BEC_{0.3}$

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

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

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



Security:
- W.h.p. Eve cannot find an r' such that $f_r(r') = m$.
- Why does $iO(f_r)$ hide m?

Security: What Does Eve See?

Eve sees:

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

Eve does not know:

$r = 1010010110$

$f_r(r')$:

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

Security: What Does Eve See?

Eve sees:

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

Eve does not know:

010110

Using standard hybrid techniques involving iO, can show that this circuit is computationally indistinguishable from a circuit that always outputs $\perp$.

$f_r(r')$:

- 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:

$$10110$$

Using standard hybrid techniques involving iO, can show that this circuit is computationally indistinguishable from a circuit that always outputs $\bot$.

$f_r(r')$:

- Output $\bot$

Security: What Does Eve See?

Eve sees:

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

Eve does not know:

10110

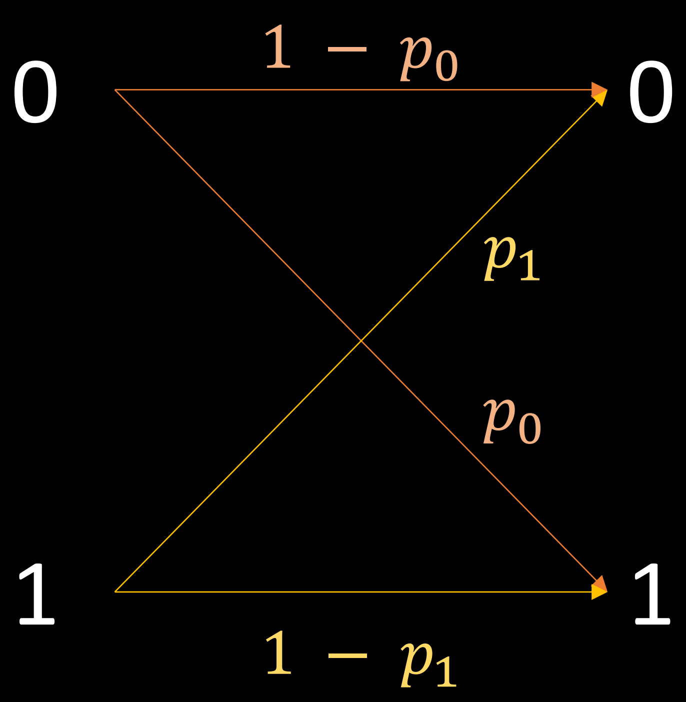
A key step in this argument is that, from Eve's point of view, each erasure is **equally likely** to be 0 or 1.

$f_r(r')$:

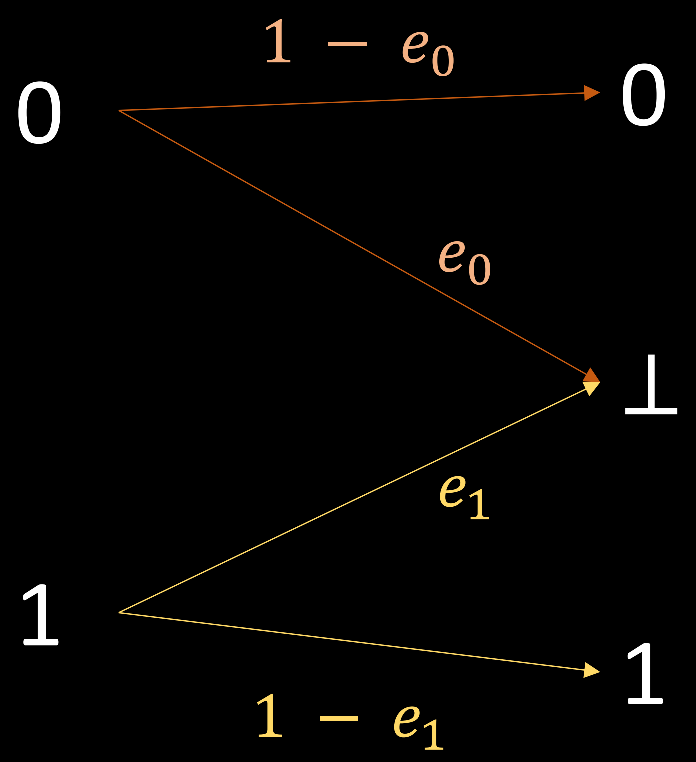
- Output $\perp$

Asymmetric Binary Channels

Binary Asymmetric Channel (BAC)

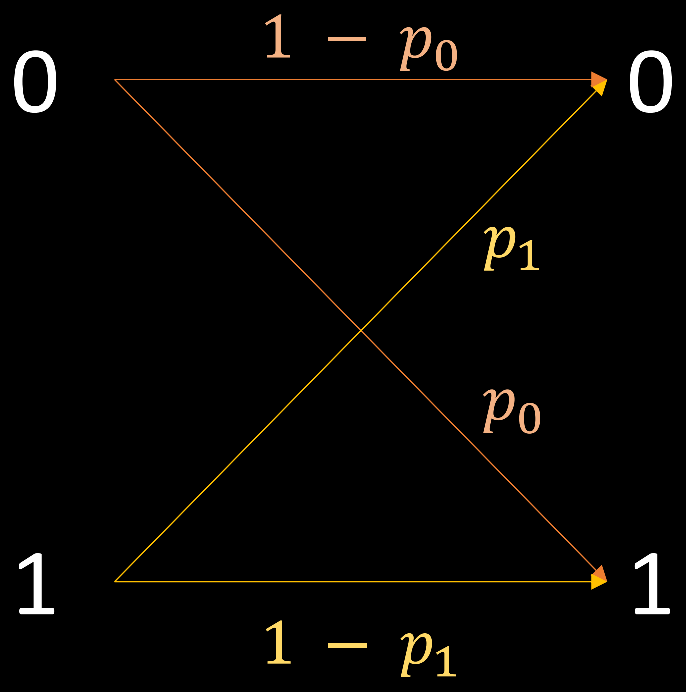


Binary Asymmetric Erasure Channel (BAEC)



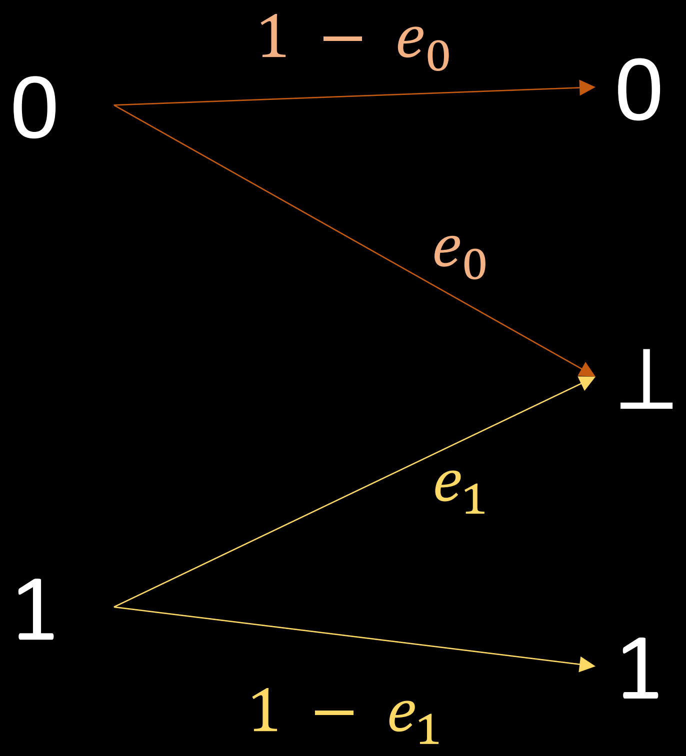
Asymmetric Binary Channels

Binary Asymmetric Channel (BAC)



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

Binary Asymmetric Erasure Channel (BAEC)



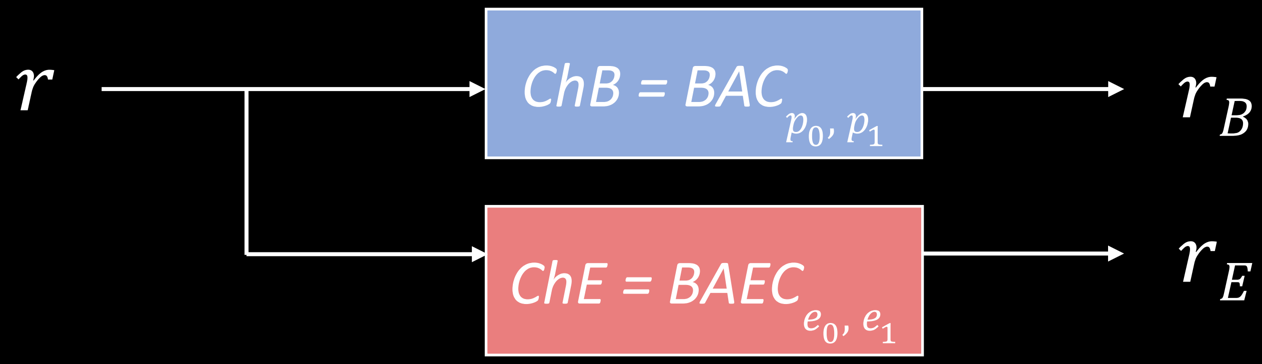
$$\begin{bmatrix} 1-e_0 & 0 & e_0 \\ 0 & 1-e_1 & e_1 \end{bmatrix}$$

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

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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



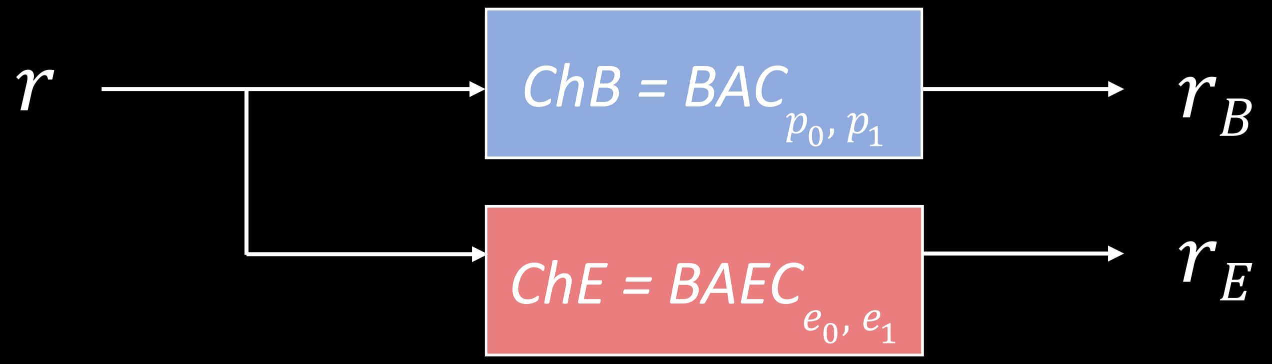
$$ChB = BAC_{p_0, p_1}, \quad ChE = BAEC_{e_0, e_1}$$

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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

What is the expected Hamming distance of $\Delta(r_B, r)$?



What is the expected number of erasures received? And what is Eve's best strategy to recover m?

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

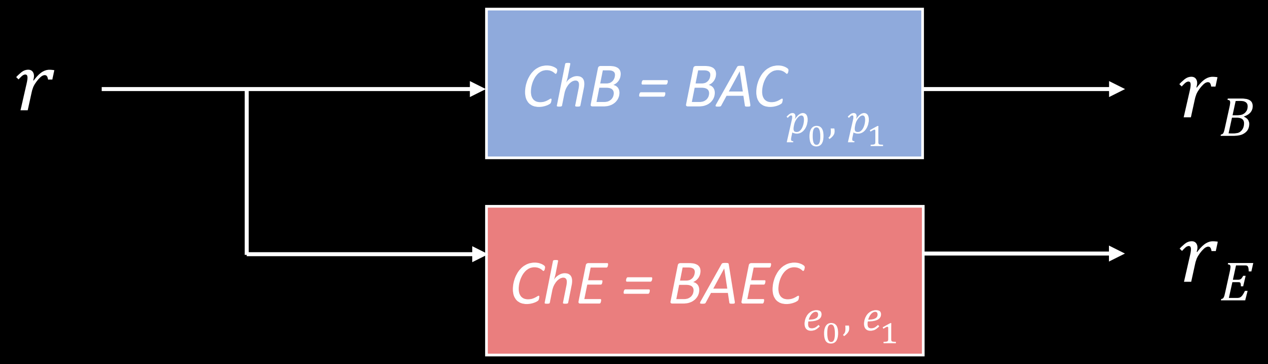
Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

Value should be the expected Hamming distance of $\Delta(r_B, r)$.

$f_r(r')$:

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

What is the expected Hamming distance of $\Delta(r_B, r)$?



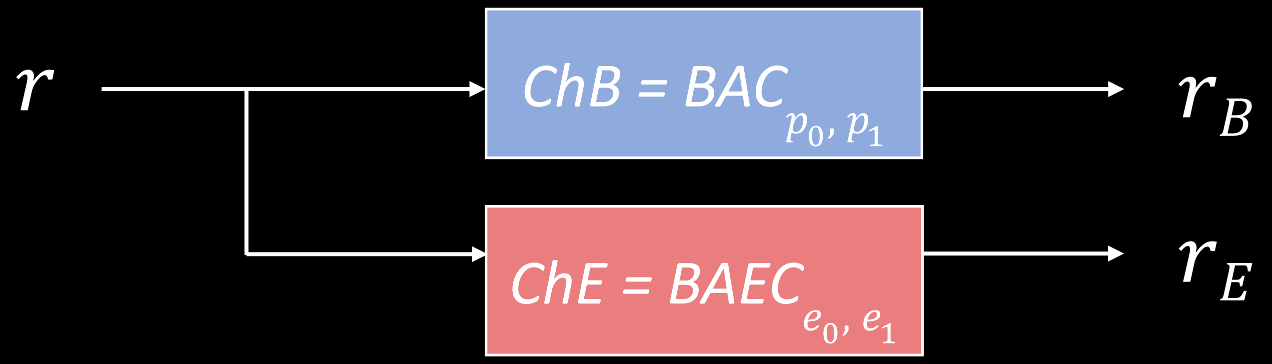
What is the expected number of erasures received? And what is Eve's best strategy to recover m?

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

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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



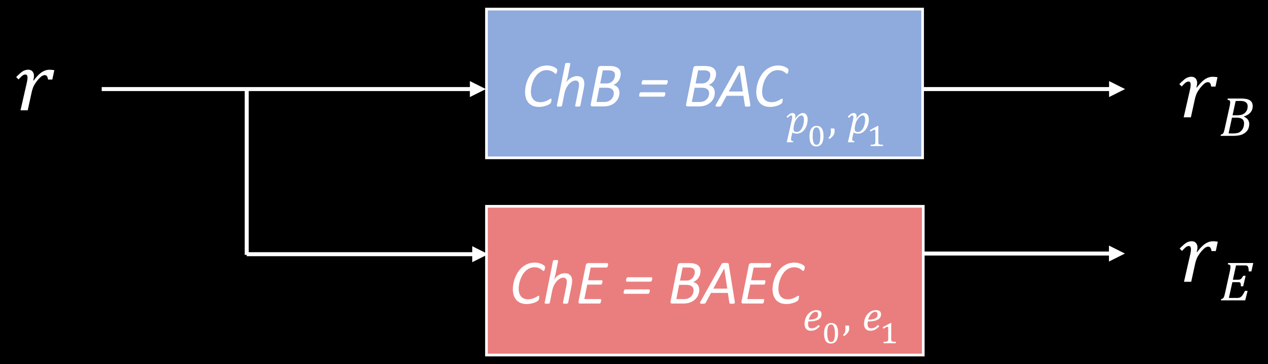
What is Eve's best strategy to recover m? If erasures were equally likely to be 0 or 1, then it's random guessing!

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

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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



Unfortunately, erasures are **not equally likely** to have been 0 or 1.

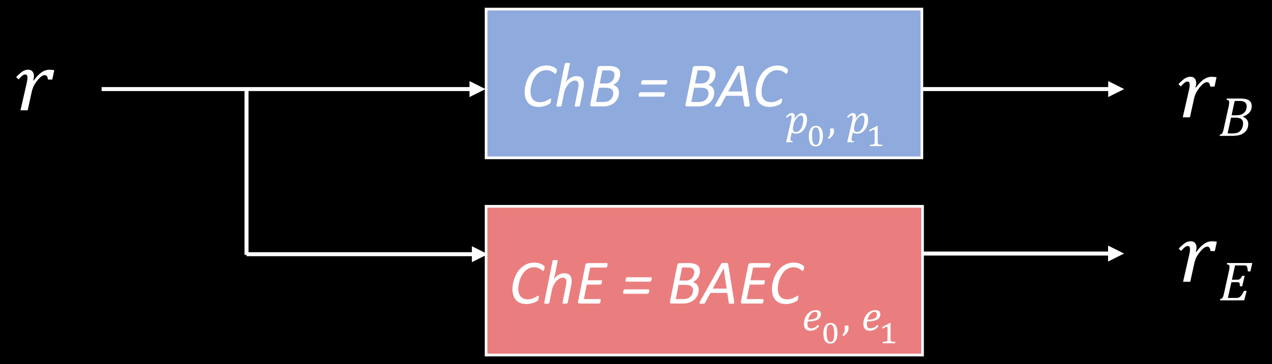
$ChB = BAC_{p_0, p_1}$, $ChE = BAEC_{e_0, e_1}$

Construction: Send a uniform random $r \in \{0,1\}^n$ across the wiretap channel. Then, s[...]fuscation of f_r defined below.

Simple Fix: Pick a distribution such that erasures are equally likely to have been 0 or 1.

$f_r(r')$:

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



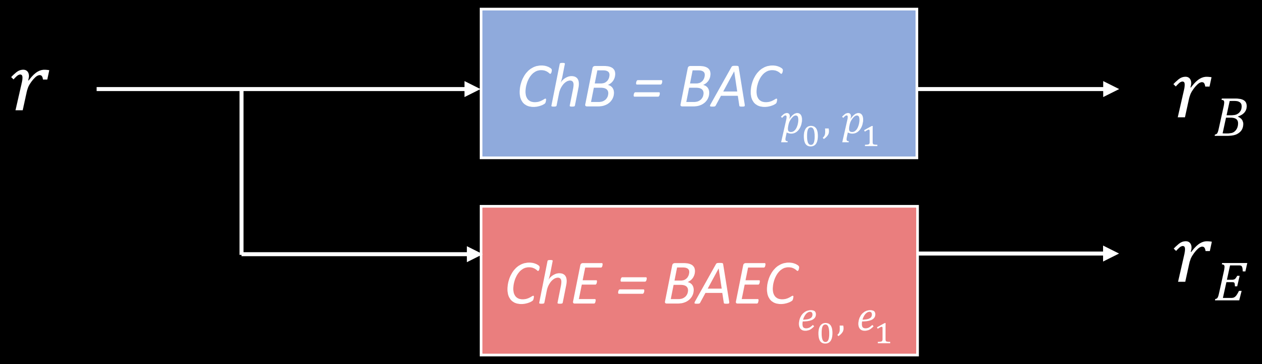
Unfortunately, Erasures are **not equally likely** to have been 0 or 1.

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

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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

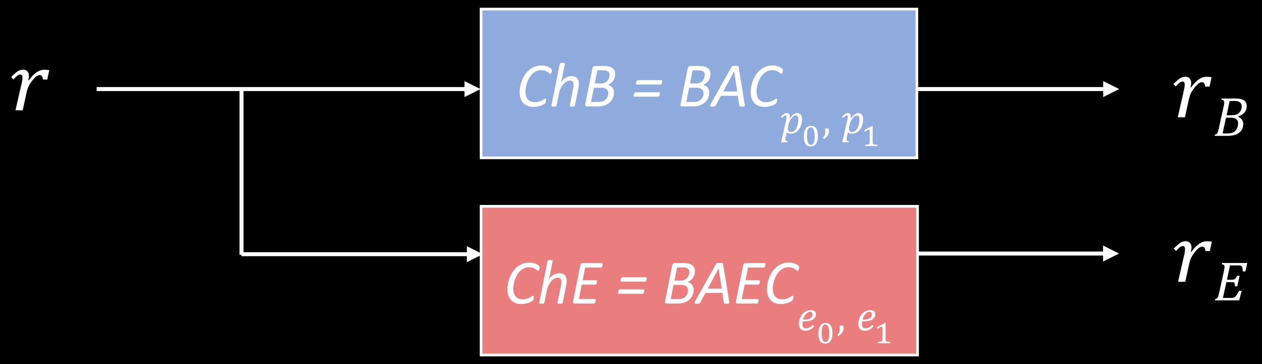


$$ChB = BAC_{p_0, p_1}, \quad ChE = BAEC_{e_0, e_1}$$

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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



Erasures are now **equally likely** to have been 0 or 1.

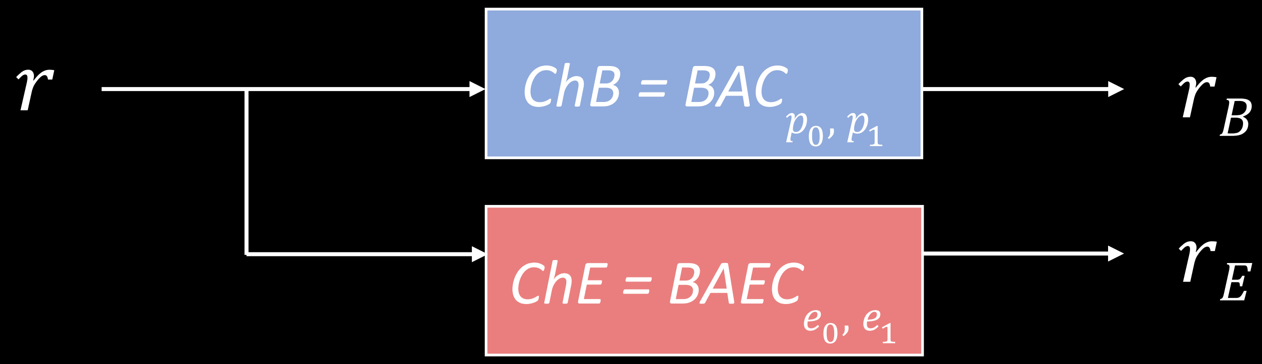
$$ChB = BAC_{p_0,\,p_1},\quad ChE = BAEC_{e_0,\,e_1}$$

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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

What is the expected Hamming distance of $t = \Delta(r_B, r)$?



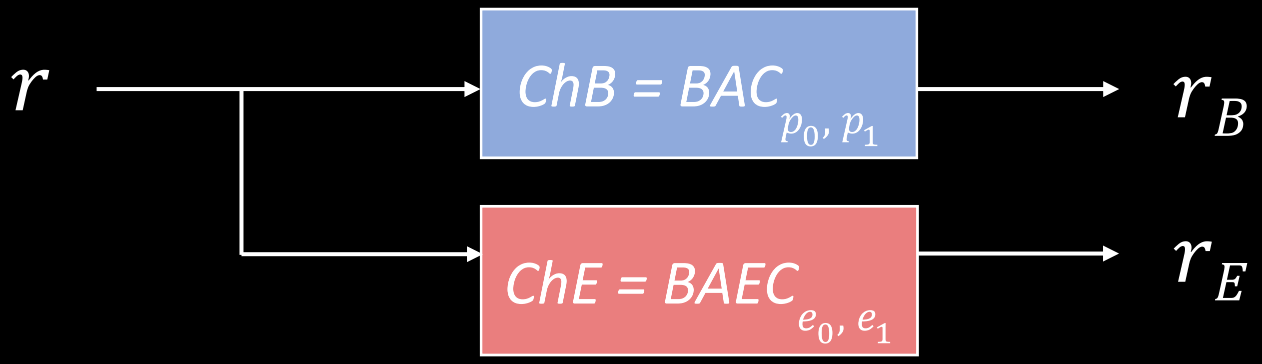
Erasures are now **equally likely** to have been 0 or 1.

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

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

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

$$t = \frac{e_0 p_1 + e_1 p_0}{e_0 + e_1}$$



Erasures are now **equally likely** to have been 0 or 1.

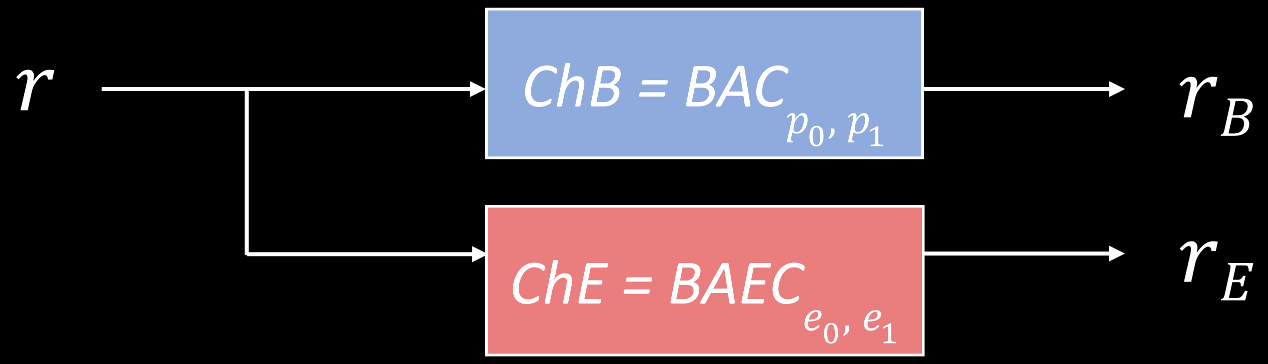
$$ChB = BAC_{p_0,p_1},\ ChE = BAEC_{e_0,e_1}$$

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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

$$t = \frac{e_0 p_1 + e_1 p_0}{e_0+e_1}$$



What is the expected number of erasures received?

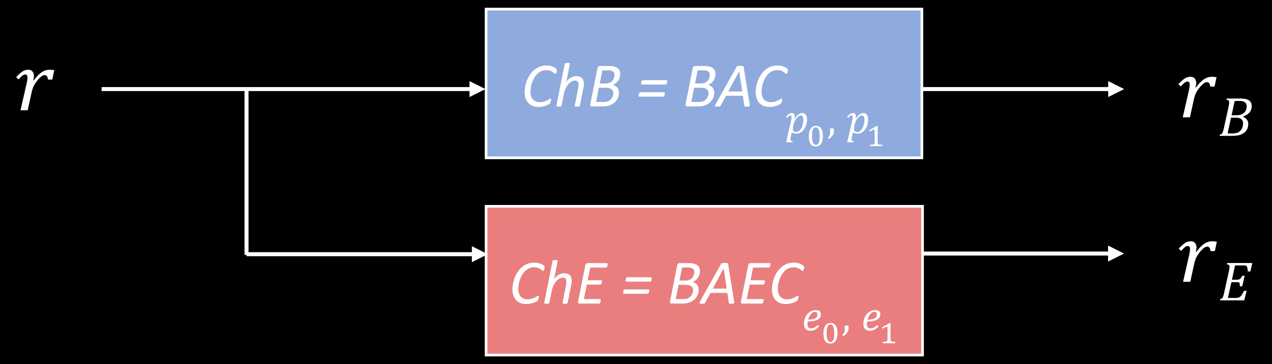
$$ChB = BAC_{p_0,\, p_1},\ ChE = BAEC_{e_0,\, e_1}$$

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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

$t = \frac{e_0 p_1 + e_1 p_0}{e_0+e_1}$



$\frac{2e_0 e_1}{e_0+e_1}$

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

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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

$$t = \frac{e_0 p_1 + e_1 p_0}{e_0 + e_1}$$

$r \rightarrow ChB = BAC_{p_0, p_1} \rightarrow r_B$

$r \rightarrow ChE = BAEC_{e_0, e_1} \rightarrow r_E$

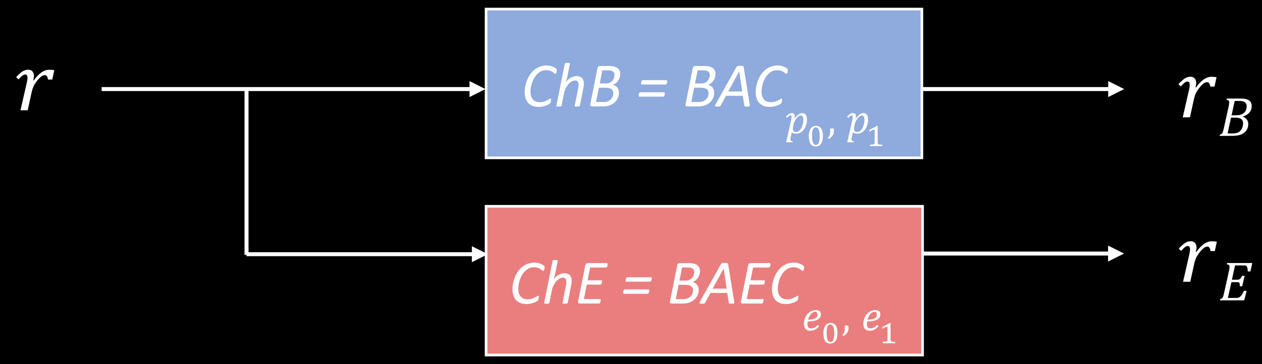
$\frac{2e_0 e_1}{e_0+e_1}$. What about the expected Hamming distance of Eve's best guess?

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

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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



$$t = \frac{e_0p_1 + e_1p_0}{e_0+e_1}$$

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

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

Construction: Sample $r \in \{0,1\}^n$ such that each bi
probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across t
Then, send across an obfuscation of f_r defined be

For Bob to have an advantage, we need $\frac{e_0 p_1 + e_1 p_0}{e_0+e_1} < \frac{e_0 e_1}{e_0+e_1}$.

$f_r(r')$:

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

$r \rightarrow ChB = BAC_{p_0, p_1} \rightarrow r_B$

$r \rightarrow ChE = BAEC_{e_0, e_1} \rightarrow r_E$

$$t = \frac{e_0 p_1 + e_1 p_0}{e_0+e_1}$$

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

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

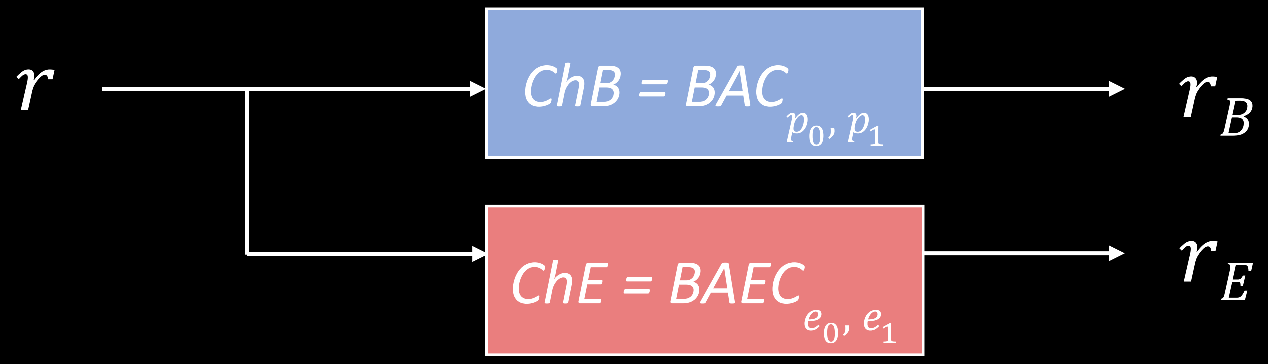
Construction: Sample $r \in \{0,1\}^n$ such that each bi
probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across t
Then, send across an obfuscation of f_r defined be

For Bob to have an advantage, we need $\frac{e_0p_1+e_1p_0}{e_0+e_1} < \frac{e_0e_1}{e_0+e_1}$.

Turns out, non-degradation condition is exactly
$e_0p_1 + e_1p_0 < e_0e_1$.

$f_r(r')$:

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



$t = \frac{e_0p_1 + e_1p_0}{e_0+e_1}$

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

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

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:

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

r_E

Eve's Viewpoint

Having established that Bob has an advantage whenever ChB is a non-degradation of ChE, use the same standard hybrid techniques involving iO as those in the symmetric case.

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

Construction: Sample $r \in \{0,1\}^n$ such that each bit of r is 0 with probability $\frac{e_1}{e_0+e_1}$ and 1 otherwise. Send r across the wiretap channel. Then, send across an obfuscation of f_r defined below.

$f_r(r')$:
- Output $\perp$.

r_E

Eve's Viewpoint

Having established that Bob has an advantage whenever ChB is a non-degradation of ChE, use the same standard hybrid techniques involving iO as those in the symmetric case.

Construction Road Map

1. The setting of the binary **asymmetric** channels (**BAC**) and binary **asymmetric** erasure channels (**BAEC**): an $i\mathcal{O}$ + injective PRG based construction.

2. Polytope formulation of degradation

3. Reducing constructing a computational wiretap coding scheme for any pair of binary input channels to the asymmetric case.

Motivating the Polytope Formulation

1. How did we obtain our degradation condition for the asymmetric setting?

2. Why is constructing a computational wiretap coding scheme for the asymmetric case sufficient for constructing a computational wiretap coding scheme for any pair of non-degraded binary input channels ?

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.
- $\mathcal{P}(A) = \{Av : 0 \leq v \leq 1, v_i \in [0,1]\}$.

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.
- $\mathcal{P}(A) = \{Av : 0 \leq v \leq 1, v_i \in [0,1]\}$.

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.
- $\mathcal{P}(A) = \{Av : 0 \leq v \leq 1, v_i \in [0,1]\}$.

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

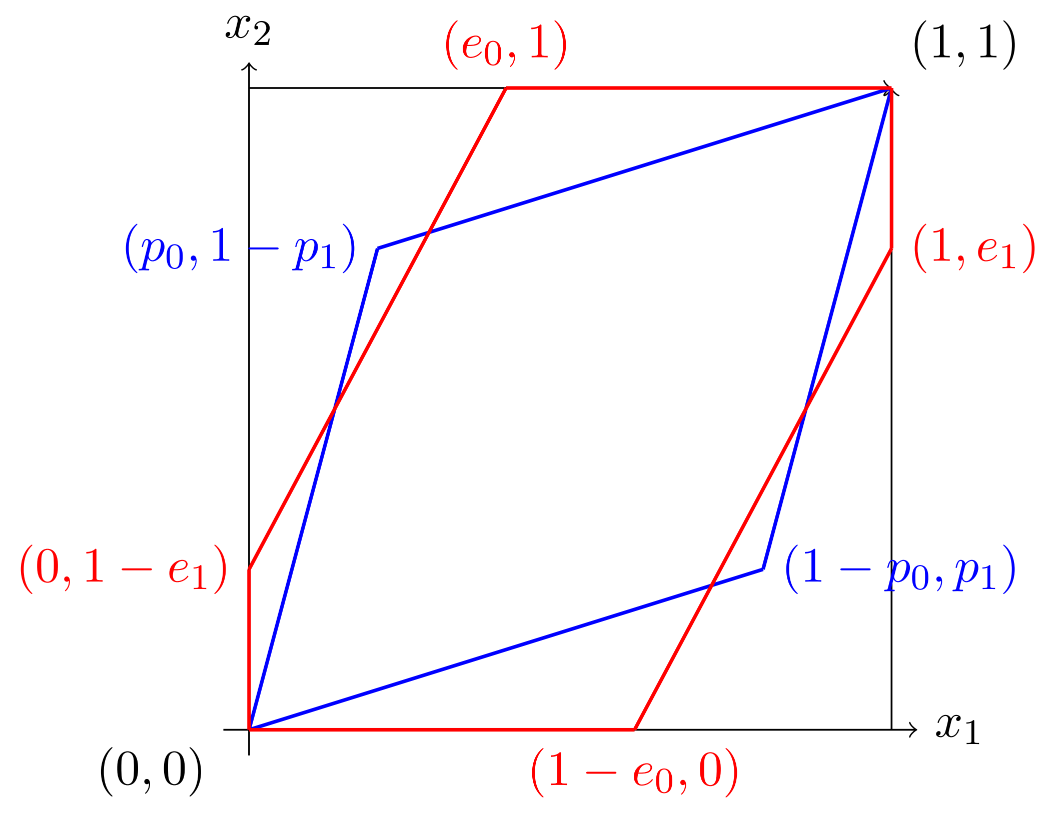
Theorem: Let $B \in \mathbb{R}^{2\times n_B}$ and E $\in \mathbb{R}^{2\times n_E}$ be arbitrary row-stochastic matrices. Then, $B \neq E \cdot S$ for every row stochastic matrix S if and only if $\mathcal{P}(B) \not\subseteq \mathcal{P}(E)$.

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)

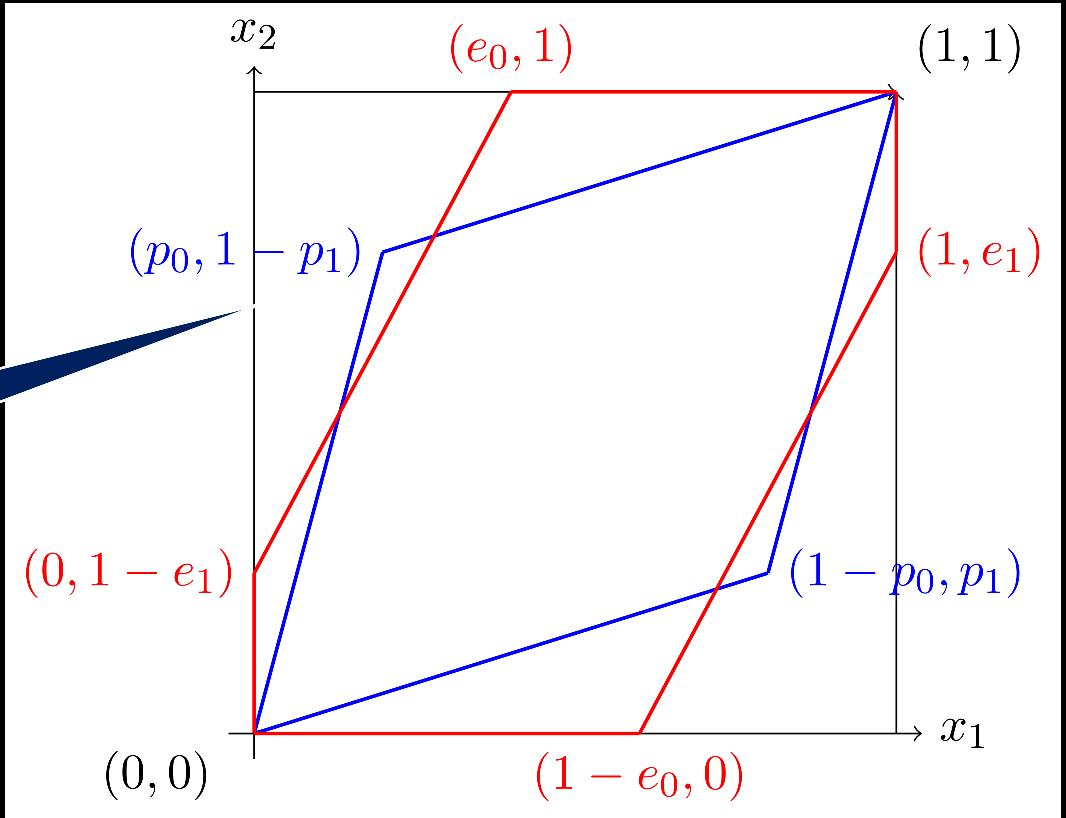


Binary Asymmetric Erasure Channel (BAEC)

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

Binary Asymmetric Channel (BAC)

Non-degradation Formula



This picture exactly gives the non-degradation condition for the BAC-BAEC case:

$$e_0 p_1 + e_1 p_0 < e_0 e_1.$$

Applications of the Polytope Formulation

1. How did we obtain our degradation condition for the asymmetric setting? ✓

2. Why is constructing a computational wiretap coding scheme for the asymmetric case sufficient for constructing a computational wiretap coding scheme for any pair of non-degraded binary input channels ?

Construction Road Map

1. The setting of the binary **asymmetric** channels (**BAC**) and binary **asymmetric** erasure channels (**BAEC**): an iO + injective PRG based construction.

2. Polytope formulation of degradation

3. Reducing constructing a computational wiretap coding scheme for any pair of binary input channels to the asymmetric case.

Pair of Arbitrary Binary Input Channels

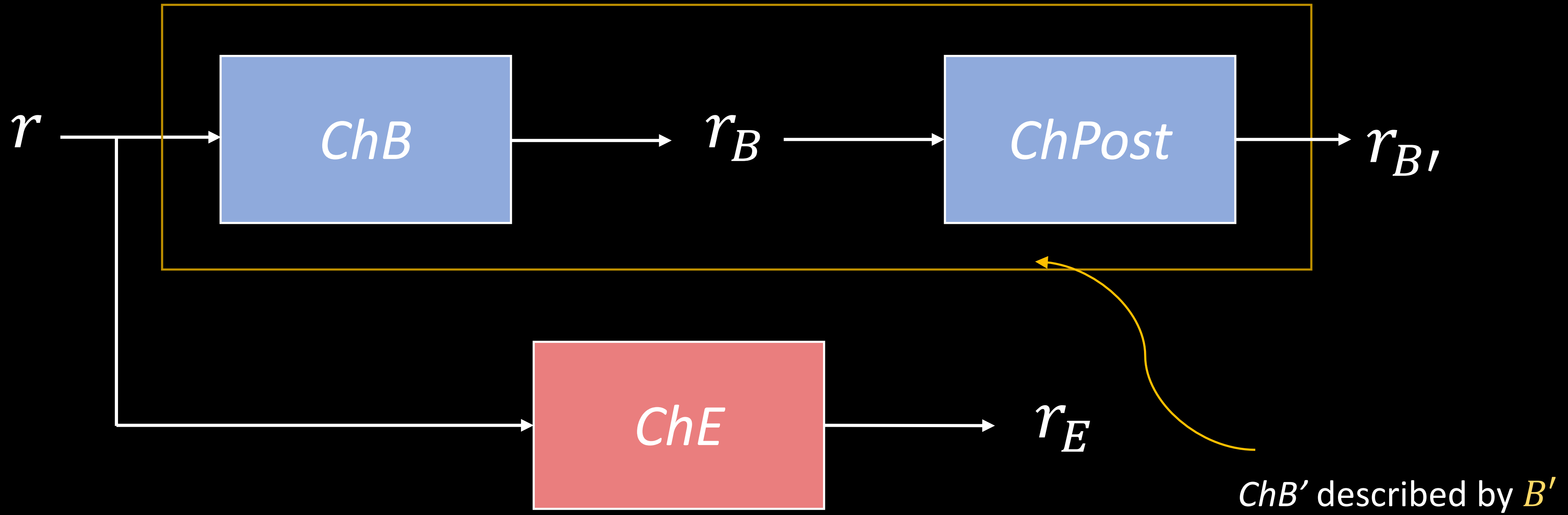
Suppose $\left(B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}, E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}\right)$ such that $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.

We want to…

1. [Bob's Output Alphabet Reduction] find a matrix $B' = BAC_{p_0,p_1}$ such that
   I. $\mathcal{P}(B') \subseteq \mathcal{P}(B)$ (Bob can perfectly simulate receiving an output from the channel described by B').
   II. $\mathcal{P}(B') \nsubseteq \mathcal{P}(E)$ (Eve cannot simulate receiving an output from B').

Pair of Arbitrary Binary Input Channels

Suppose $\left(B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}, E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}\right)$ such that $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.



Bob's Output Alphabet Reduction

Suppose $\left(B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}, E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}\right)$ such that $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.

Take any extreme point u^* (a 0/1 combination of the columns of B) of $\mathcal{P}(B)$ not contained in $\mathcal{P}(E)$.

Bob's Output Alphabet Reduction

Suppose ($B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}$, $E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}$) such that $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.

Take any extreme point u^* (a $0/1$ combination of the columns of B) of $\mathcal{P}(B)$ not contained in $\mathcal{P}(E)$.

Then $B' = \begin{bmatrix} u_1^* & 1 - u_1^* \\ u_2^* & 1 - u_2^* \end{bmatrix}$ is such that both

$$\mathcal{P}(B') \subseteq \mathcal{P}(B) \text{ and } \mathcal{P}(B) \nsubseteq \mathcal{P}(E)$$

Pair of Arbitrary Binary Input Channels

Suppose ($B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}$, $E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}$) such that $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.

We want to…

1. [Bob's Output Alphabet Reduction] find a matrix $B' = BAC_{p_0,p_1}$ such that
   I. $\mathcal{P}(B') \subseteq \mathcal{P}(B)$ (Bob can perfectly simulate receiving an output from the channel described by B').
   II. $\mathcal{P}(B') \nsubseteq \mathcal{P}(E)$ (Eve cannot simulate receiving an output from B').
2. [Reducing Eve to the Erasure Case] find a matrix $E' = BAEC_{e_0,e_1}$ that describes a channel that gives Eve even more information than if her channel was E yet this channel will still not be informative enough to simulate B'.

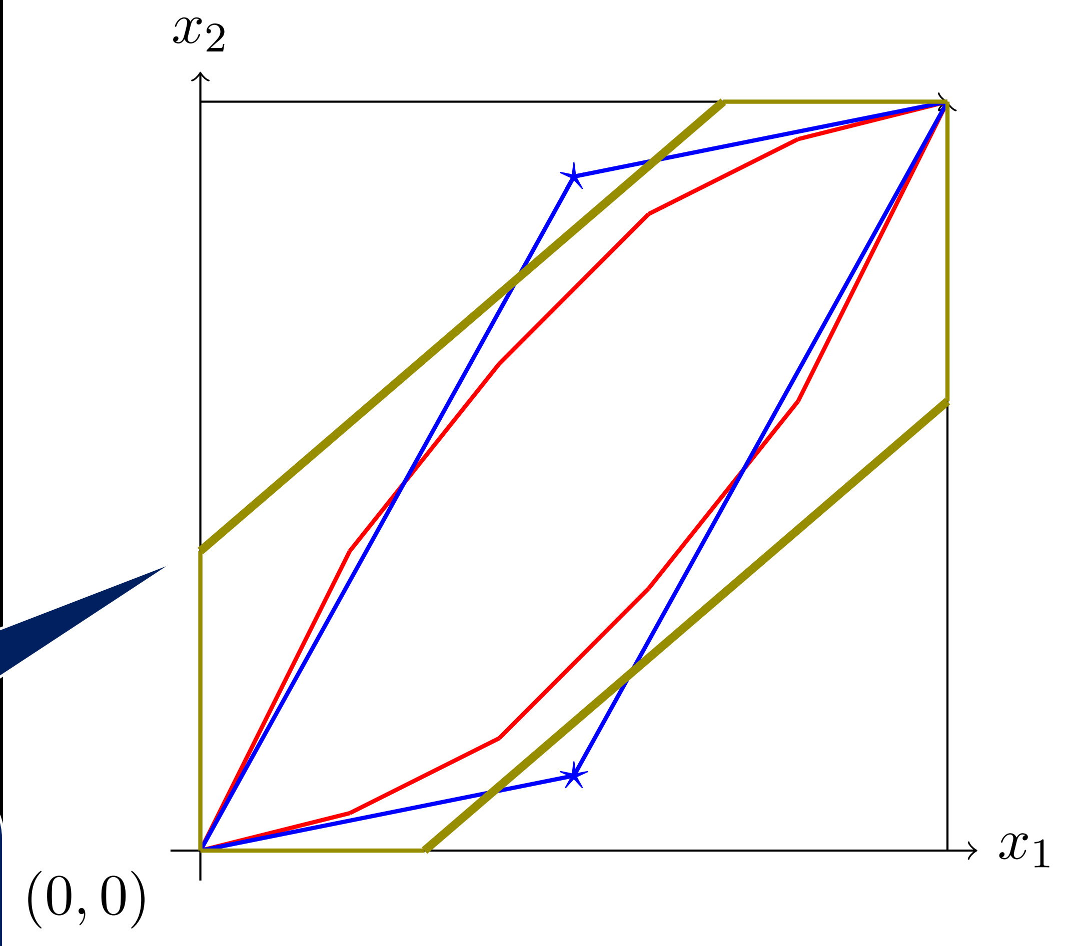
Pair of Arbitrary Binary Input Channels

Suppose $\left(B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}, E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}\right)$ such that $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.

We want to…

1. [Bob's Output Alphabet Reduction] find a matrix $B' = BAC_{p_0,p_1}$ such that
   I. $\mathcal{P}(B') \subseteq \mathcal{P}(B)$ (Bob can perfectly simulate receiving an output from the channel described by B').
   II. $\mathcal{P}(B') \nsubseteq \mathcal{P}(E)$ (Eve cannot simulate receiving an output from B').
2. [Reducing Eve to the Erasure Case] find a matrix $E' = BAEC_{e_0,e_1}$ such that
   I. $\mathcal{P}(E) \subseteq \mathcal{P}(E')$ (Eve that receives an output from E' can perfectly simulate receiving an output from E).
   II. $\mathcal{P}(B') \nsubseteq \mathcal{P}(E')$ (Eve that receives an output from E' cannot simulate receiving an output from B').

Reducing Eve's Channel to a BAEC



Apply the strict separating hyperplane theorem!

This olive polytope is the BAEC that contains Eve's channel's polytope yet is not contained by the BAC.

Pair of Arbitrary Binary Input Channels

Suppose ($B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}$, $E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}$) such that $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.

We want to…

1. [Bob's Output Alphabet Reduction] find a matrix $B' = BAC_{p_0,p_1}$ such that
   I. $\mathcal{P}(B') \subseteq \mathcal{P}(B)$ (Bob can perfectly simulate receiving an output from the channel described by B').
   II. $\mathcal{P}(B') \nsubseteq \mathcal{P}(E)$ (Eve cannot simulate receiving an output from B').
2. [Reducing Eve to the Erasure Case] find a matrix $E' = BAEC_{e_0,e_1}$ such that
   I. $\mathcal{P}(E) \subseteq \mathcal{P}(E')$ (Eve that receives an output from E' can perfectly simulate receiving an output from E).
   II. $\mathcal{P}(B') \nsubseteq \mathcal{P}(E')$ (Eve that receives an output from E' cannot simulate receiving an output from B').

Pair of Arbitrary Binary Input Channels

Suppose $\left(B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}, E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}\right)$ such that $\mathcal{P}(B) \nsubseteq \mathcal{P}(E)$.

A computational wiretap coding scheme for (B, E):

1. $Enc(1^\lambda, b)$: Use any computational wiretap *encoding* algorithm for $(B' = BAC_{p_0,p_1}\ , E' = BAEC_{e_0,e_1})$.

Pair of Arbitrary Binary Input Channels

Suppose $\left(B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}, E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}\right)$ such that $\mathcal{P}(B) \not\subseteq \mathcal{P}(E)$.

A computational wiretap coding scheme for (B, E):

1. $Enc(1^\lambda, b)$: Use any computational wiretap *encoding* algorithm for $(B' = BAC_{p_0,p_1}, E' = BAEC_{e_0,e_1})$.
2. $Dec\left(1^\lambda, ChB\left(Enc(1^\lambda, b)\right)\right)$:
    1. Perfectly simulate $ChB'\left(Enc(1^\lambda, b)\right)$ by using $ChB\left(Enc(1^\lambda, b)\right)$.
    2. Use any computational wiretap *decoding* algorithm for $(B' = BAC_{p_0,p_1}, E' = BAEC_{e_0,e_1})$.

Pair of Arbitrary Binary Input Channels

Suppose $\left(B = \begin{bmatrix} u_{11} & \cdots & u_{1n_B} \\ u_{21} & \cdots & u_{2n_B} \end{bmatrix}, E = \begin{bmatrix} u_{11} & \cdots & u_{1n_E} \\ u_{21} & \cdots & u_{2n_E} \end{bmatrix}\right)$ such that $\mathcal{P}(B) \not\subseteq \mathcal{P}(E)$.

A computational wiretap coding scheme for (B, E):

1. $Enc(1^\lambda, b)$: Use any computational wiretap *encoding* algorithm for $(B' = BAC_{p_0,p_1}, E' = BAEC_{e_0,e_1})$.
2. $Dec\left(1^\lambda, ChB\left(Enc(1^\lambda, b)\right)\right)$:
    1. Perfectly simulate $ChB'\left(Enc(1^\lambda, b)\right)$ by using $ChB\left(Enc(1^\lambda, b)\right)$.
    2. Use any computational wiretap *decoding* algorithm for $(B' = BAC_{p_0,p_1}, E' = BAEC_{e_0,e_1})$.

Correctness: Bob can perfectly simulate ChB'

Security: Eve can perfectly simulate ChE using ChE'. If she can break this coding scheme, she can break the (B', E') coding scheme.

Construction Road Map

1. The setting of the binary **asymmetric** channels (**BAC**) and binary **asymmetric** erasure channels (**BAEC**): an iO + injective PRG based construction.

2. Polytope formulation of degradation

3. Reducing constructing a computational wiretap coding scheme for any pair of binary input channels to the asymmetric case.

Future Directions – Cryptography

1. Can we characterize channel degradation for higher dimensions than two and can we obtain an iO-based solution for all higher dimensions?

2. Do we need program obfuscation to construct computational wiretap coding schemes?

3. More generally, what is the minimum cryptographic assumption that suffices for constructing computational wiretap coding schemes?

4. Does the existence of a computational wiretap coding scheme, say for the pair of channels $(BSC_{0.1}, BEC_{0.3})$ imply key exchange in the plain model?

Future Directions – Coding Theory

For traditional error-correcting codes (ECCs) ,the task of correcting erasures is significantly easier than correcting errors (e.g. bit flips).

In our work, we give a *randomized* encoding procedure such that, for parameters $e > 2p$, correcting a p fraction of *random* errors can be efficiently done while correcting a e fraction of random} erasures cannot be efficiently done.

1. Can we *directly* (not through program obfuscation) construct codes with these properties?
2. Moreover, can we construct one with a *deterministic* encoder?

- Can we design computational wiretap coding schemes from hard average-case problems (e.g. a planted random CSP or a planted graph problem)?

Future Directions – Average-case Complexity Theory

Can we design computational wiretap coding schemes from hard average-case problems (e.g. a planted random CSP or a planted graph problem)?

- We require sharp thresholds at which the problem phase changes from easy to computationally difficult.
- For example, for (BSC_p, BEC_e), we have an inversion problem $\mathrm{P}_{p,e}(x)$ where one is given some "side information" x'.

  We desire that if x' has a *random* p fraction of errors, then recovering x is easy, and instead if x' has a *random* e fraction of erasures, then recovering x is hard.

Thank you!

Appendix: Statistically Evasive Circuit Families

Statistically Evasive Circuit Collection with Auxiliary Input

Let D be a distribution of circuits.

Let Aux be an auxiliary input generator.

For all <u>unbounded</u> oracle machines A that are limited to polynomially many queries to their oracle and for all λ,

$$\Pr\left[C\left(A^C(1^\lambda, Aux(1^\lambda, C))\right) = 1; C \leftarrow D(1^\lambda)\right] \leq negl(\lambda)$$

Statistically Evasive Circuit Collection with Auxiliary Input

Let D be a distribution of circuits.

Let Aux be an auxiliary input generator.

D will be a class of generalized fuzzy point functions with a randomly chosen center r.

For all unbounded oracle machines A that are limited to polynomially many queries to their oracle and for all λ,

$$\Pr\left[C\left(A^C(1^\lambda, Aux(1^\lambda, C))\right) = 1; C \leftarrow D(1^\lambda)\right] \leq negl(\lambda)$$

Statistically Evasive Circuit Collection with Auxiliary Input

Let *D* be a distribution of circuits.

Let *Aux* be an auxiliary input generator.

D will be a class of generalized fuzzy point functions with a randomly chosen center *r*.

Aux = ChE(r)

For all <u>unbounded</u> oracle machines *A* that are li… many queries to their oracle and for all λ,

$$\Pr\left[C\left(A^C(1^\lambda, Aux(1^\lambda, C))\right) = 1; C \leftarrow D(1^\lambda)\right] \le negl(\lambda)$$

Statistically Evasive Function Obfuscation

Let *(D, Aux)* be a statistically evasive circuit collection with auxiliary input.

Correctness: For all λ, all $C \leftarrow D(1^\lambda)$,

$$\Pr\left[\forall x, Obf(1^\lambda, C)(x) \neq C(x)\right] \leq negl(\lambda)$$

VBB Security: For all polytime *A*, there exists a polytime oracle machine *Sim* such that for all λ,

$$\Big|\Pr\left[A\left(1^\lambda, Obf(1^\lambda, C), Aux(1^\lambda, C)\right) = 1; C \leftarrow D\left(1^\lambda\right)\right]$$
$$-\Pr\left[Sim^C\left(1^\lambda, 1^{|C|}, Aux(1^\lambda, C)\right) = 1; C \leftarrow D\left(1^\lambda\right)\right]\Big| \leq negl(\lambda)$$

Statistically Evasive Function Obfuscation

- No impossibility results known for VBB obfuscation of <u>statistically</u> evasive circuits!
  - Previous impossibility results for evasive circuits require the auxiliary input to statistically reveal non-trivial inputs.
- Plausible conjecture that *iO* achieves statistically evasive function obfuscation since *iO* is a best possible obfuscator [GR07].
- [BSMZ16] gives a construction with security in an idealized weak multilinear map model with no known attacks.

Appendix: Security Proof for $i\mathcal{O}$-based construction.

Brief Sketch of Security: What Does Eve See?

Eve sees:

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

Eve does not know:

$r = 1010010110$

$f_r(r')$:

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

Security: What Does Eve See?

Eve sees:

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

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

$f_r(r')$:

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

Eve does not know:

$r = 1010010110$

Security: An Indistinguishable Viewpoint (1)

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, $S_\perp$.

- 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 (1)

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:

$r = 1010010110$

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

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

Constants: r, $S_\perp$.

- 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.

Getting to the Null Circuit: The iO “PRG Trick”

Consider a length-tripling PRG G.

Sample a random element α.

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

Constants: $r, S_\perp$

- 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.

Getting to the Null Circuit: The iO "PRG Trick"

Consider a length-tripling PRG G.

Sample a random element α.

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

Constants: $r, S_{\perp}, G.$

- Add a conditional branch that doesn't change the functionality of the form: "If $G(?) \neq G(\alpha)$, then output $\perp$".
- 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.

Getting to the Null Circuit: The iO "PRG Trick"

Consider a length-tripling PRG G.

Sample a random element α.

Goal: Switch $G(\alpha)$ with a uniform random R. With overwhelming probability, R is not in the image of G. After the switch, the branch will always execute, resulting in a null circuit.

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

Constants: $r, S_\perp, G$.

- Add a conditional branch that doesn't change the functionality of the form: "If $G(?) \neq G(\alpha)$, then output $\perp$".
- 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.

“Code Offset” construction

Injective length-tripling PRG G.

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

$f^{(1)}(r')$:
Constants: r, $S_\perp$

- 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.

"Code Offset" construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

$f^{(2)}(r')$:
Constants: $r_{S_{0,1}}$, $G(\alpha)$, z, $S_\perp$

- 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.

"Code Offset" construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

If $r'_{S_\perp}$ and $r_{S_\perp}$ are close to each other, then $z + r'_{S_\perp} + r_{S_\perp} \approx \mathcal{C}(\alpha)$.

By list-decoding, L will therefore contain α.

$f^{(2)}(r')$:
Constants: $r_{S_{0,1}}$, $G(\alpha)$, z, $S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
- 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.

“Code Offset” construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

By injectivity of G, there’s a unique preimage of $G(\alpha)$, so we can recover α.

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

Constants: $r_{S_{0,1}}, G(\alpha), z, S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.

- 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.

“Code Offset” construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

Why did we hide $\mathcal{C}(\alpha)$?

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

Constants: $r_{S_{0,1}}, G(\alpha), z, S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.
- 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.

"Code Offset" construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

Why did we hide $\mathcal{C}(\alpha)$?

Ultimately, we'll want to switch $G(\alpha)$ with uniform random R, so there cannot be other constants correlated with α.

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

Constants: $r_{S_{0,1}}, G(\alpha), z, S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
    - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.

- 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.

"Code Offset" construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

Why is z uncorrelated with α?

Because from Eve's point of view, every erasure is *equally likely* to have been a 0 or 1, so $r_{S_\perp}$ is uniform random.

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

Constants: $r_{S_{0,1}}, G(\alpha), z, S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.

- 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.

"Code Offset" construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

Why is z uncorrelated with α?

Because from Eve's point of view, every erasure is *equally likely* to have been a 0 or 1, so $r_{S_\perp}$ is uniform random.

We removed $r_{S_\perp}$!

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

Constants: $r_{S_{0,1}}, G(\alpha), z, S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.
- 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.

“Code Offset” construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

Why is z uncorrelated with α?

Because from Eve’s point of view, every erasure is *equally likely* to have been a 0 or 1, so $r_{S_\perp}$ is uniform random.

We removed $r_{S_\perp}$!

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

Constants: $r_{S_{0,1}}, G(\alpha), z, S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.
- 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.

We still need to recover $r_{S_\perp}$ to maintain functionality!!

"Code Offset" construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

Why is z uncorrelated with α?

Because from Eve's point of view, every erasure is *equally likely* to have been a 0 or 1, so $r_{S_\perp}$ is uniform random.

We removed $r_{S_\perp}$!

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

Constants: $r_{S_{0,1}}, G(\alpha), z, S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.
- Recover $r_{S_\perp} \leftarrow \mathcal{C}(\alpha) + z$.
- 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.

We still need to recover $r_{S_\perp}$ to maintain functionality!!

“Code Offset” construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

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

Constants: $r_{S_{0,1}}$, $G(\alpha)$, z, $S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.
- Recover $r_{S_\perp} \leftarrow \mathcal{C}(\alpha) + z$.
- 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.

"Code Offset" construction

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.

Why is this functionally equivalent (w.h.p.) to $f^{(1)}(\cdot)$?

By the degradation condition, there will exist a choice of constant ε such that whenever the Hamming distance check passes, the initial recovery process succeeds.

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

Constants: $r_{S_{0,1}}$, $G(\alpha)$, z, $S_{\perp}$

- Attempt to recover α: List-decode
  - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.
- Recover $r_{S_{\perp}} \leftarrow \mathcal{C}(\alpha) + z$.
- 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 Pseudorandomness

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$

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

Constants: $r_{S_{0,1}}, G(\alpha), z, S_\perp$

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq G(\alpha)$, then output $\perp$.
- Recover $r_{S_\perp} \leftarrow \mathcal{C}(\alpha) + z$.
- 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 Pseudorandomness

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$
3. Sample a uniform random R.

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

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

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq R$, then output $\perp$.
- Recover $r_{S_\perp} \leftarrow \mathcal{C}(\alpha) + z$.
- 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 Pseudorandomness

Injective length-tripling PRG G.

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

1. Sample a random element α in domain of $\mathcal{C}$.
2. Hide $\mathcal{C}(\alpha)$ by producing $z \leftarrow \mathcal{C}(\alpha) + r_{S_\perp}$
3. Sample a uniform random R.

With all but negligible probability (due to length-tripling), R will not be in the image of G.

Therefore, the circuit will always output $\perp$

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

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

- Attempt to recover α: List-decode $z + r'_{S_\perp}$ to obtain a list L.
  - If for all $s \in L$, $G(s) \neq R$, then output $\perp$.
- Recover $r_{S_\perp} \leftarrow \mathcal{C}(\alpha) + z$.
- 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 Pseudorandomness

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

- Output $\perp$.

Appendix: Optimal Rate

Optimal Rate

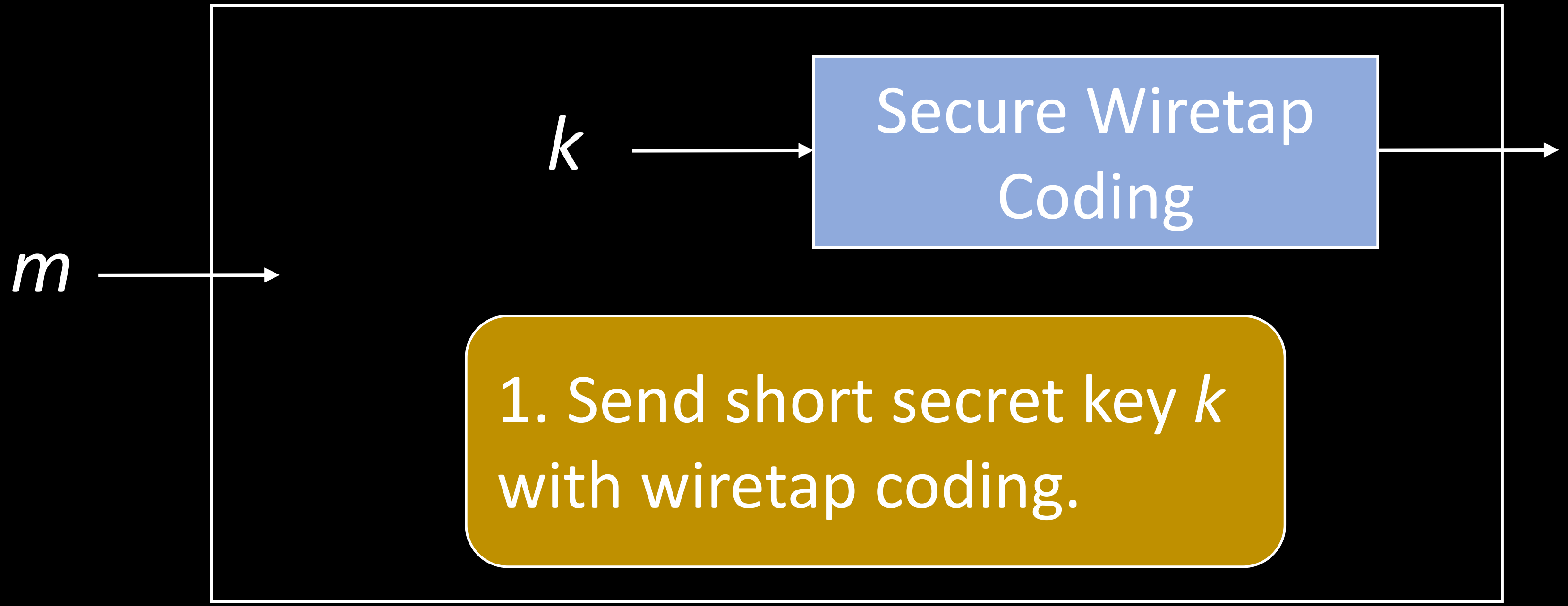
We can achieve optimal rate in the computational setting.

(Rate approaching capacity of *ChB*)

Optimal Rate

We can achieve optimal rate in the computational setting.

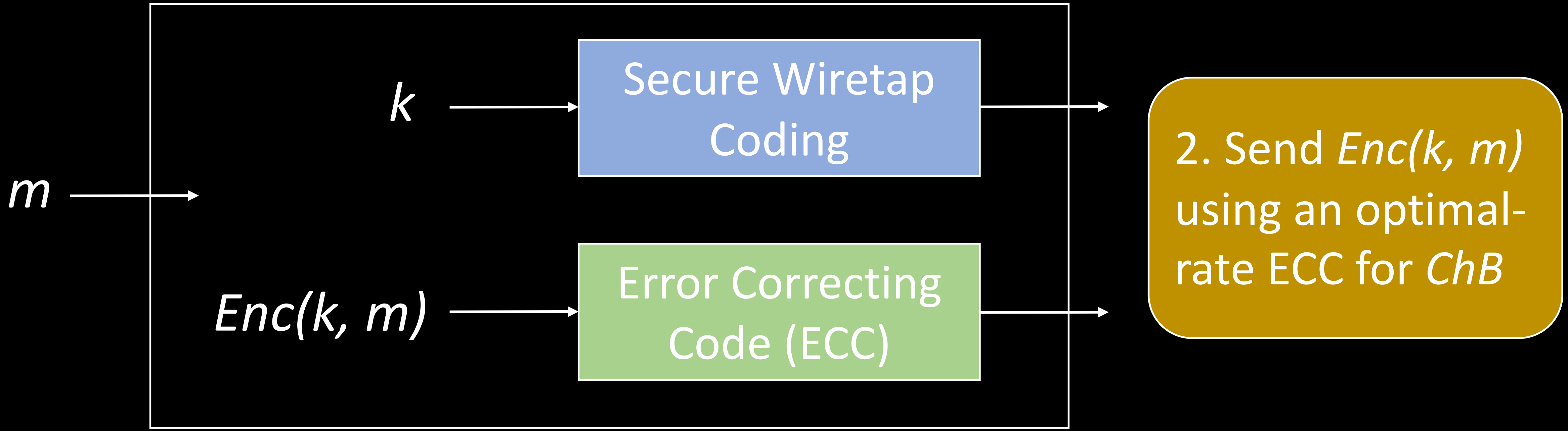
(Rate approaching capacity of *ChB*)



Optimal Rate

We can achieve optimal rate in the computational setting.

(Rate approaching capacity of ChB)



Appendix: Main Hybrid Argument Details

Starting Point: Optimal Strategy g*

Let g* be any deterministic strategy that maximizes
$$\Pr\left[f_r\left(g^*(r_E)\right) = m\right]$$

(WLOG, we can assume an optimal g* to be deterministic.)

H_0: Add structure to g*

Let g* be any deterministic strategy that maximizes

$$\Pr\left[f_r\left(g^*(r_E)\right) = m\right]$$

Key Observation: We can exploit symmetry!

$f_r(r')$:

- Output m if for all (x,y), $|\{i \ \in [n]: r_i = x \; and \; r'_i = y\}|$ is as expected for an r’ = ChB(r).
- Output ⊥ otherwise.

H_0: Add structure to g*

Let g* be any deterministic strategy that maximizes
$$\Pr\left[f_r\left(g^*(r_E)\right) = m\right]$$

Key Observation: We can exploit symmetry!

$f_r(r')$:
- Output m if for all (x,y), $|\{i \in [n]: r_i = x \text{ and } r'_i = y\}|$ is as expected for an r' = ChB(r).
- Output $\perp$ otherwise.

f_r only looks at the counts!

g* equally likely to win on $r_E = \pi(s)$ as $r_E = s$ where π is a permutation.

H_0: Add structure to g*

Let Eve_0 be the following strategy:

$r_E \rightarrow$

b	**a**	**c**	**c**	**b**	**a**	**c**

H_0: Add structure to g*

Let Eve_0 be the following strategy:

$r_E \rightarrow$

b	a	c	c	b	a	c

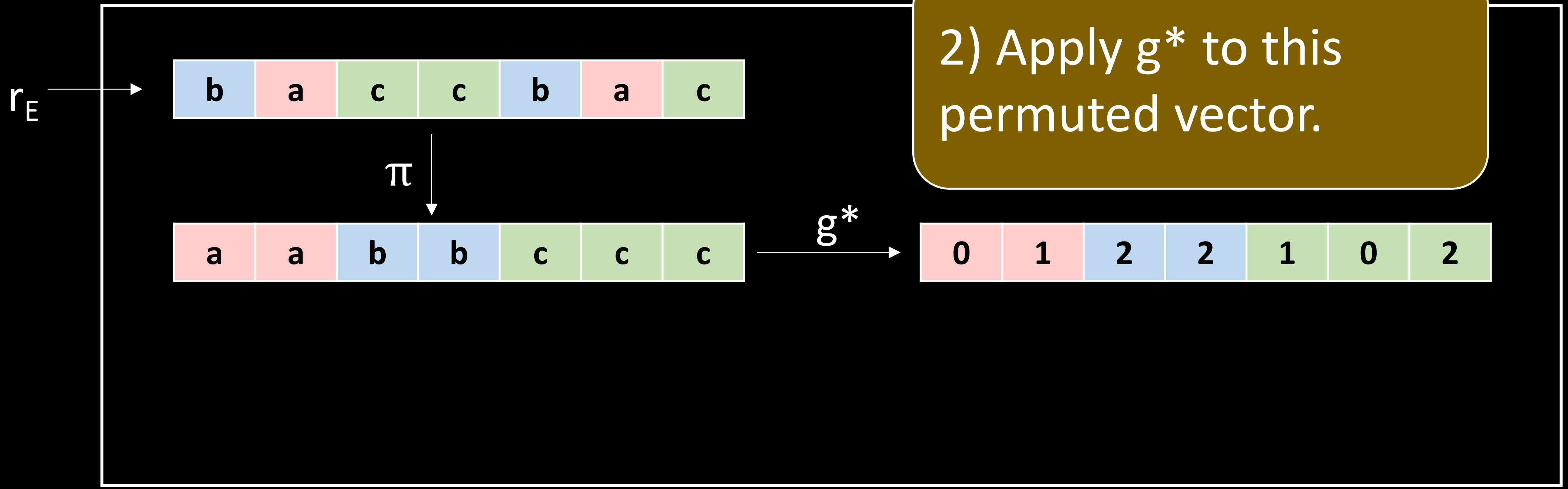
$\downarrow \pi$

a	a	b	b	c	c	c

1) Permute r_E into the lexicographically first vector of same weight.

H_0: Add structure to g*

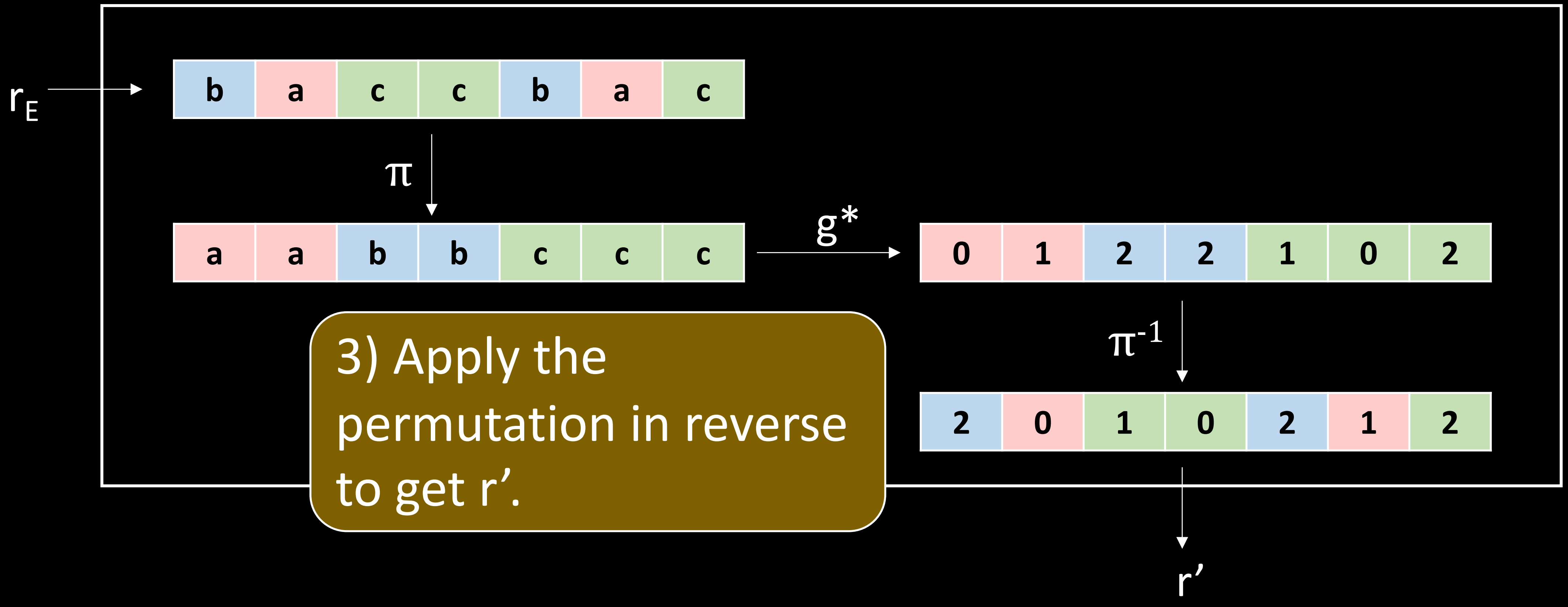
Let Eve_0 be the following strategy:



2) Apply g* to this permuted vector.

H_0: Add structure to g*

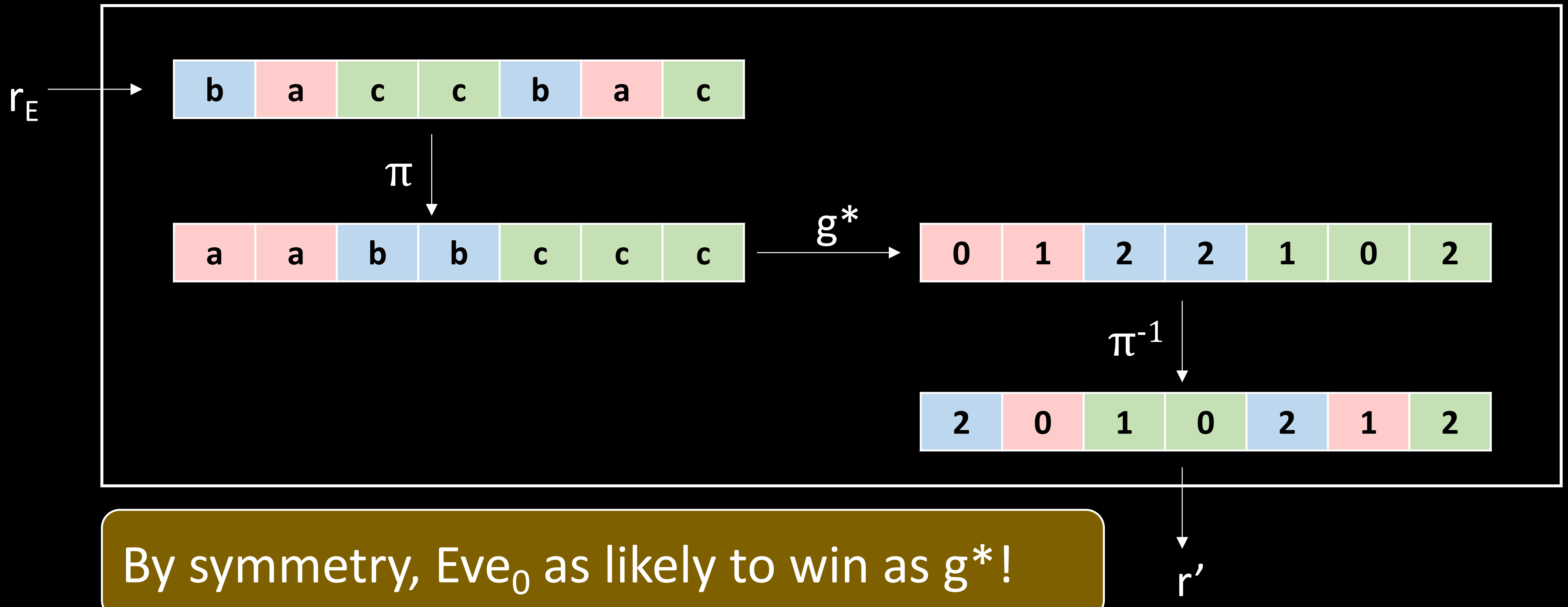
Let Eve_0 be the following strategy:



3) Apply the permutation in reverse to get r'.

H_0: Add structure to g*

Let Eve_0 be the following strategy:



By symmetry, Eve_0 as likely to win as g*!

H_0: Add structure to g*

By symmetry, Eve_0 is also an optimal strategy!

Nice Property: For any permutation π,

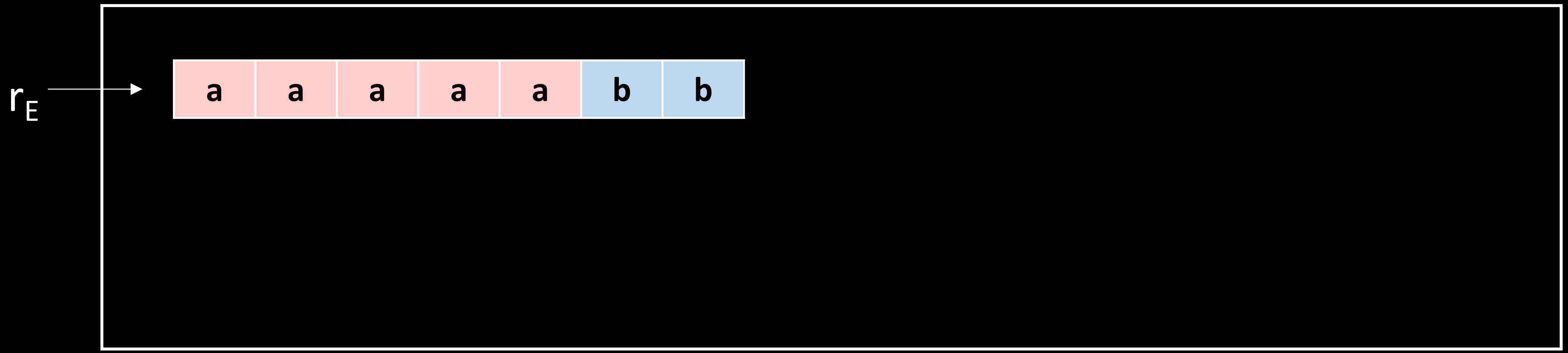
$$\pi(\mathrm{Eve}_0(r_E)) = \mathrm{Eve}_0(\pi(r_E))$$

Def: r and s have the same weight if there exists a permutation π such that $\pi(r) = s$

Eve_0 acts similarly on all vectors of the same weight.

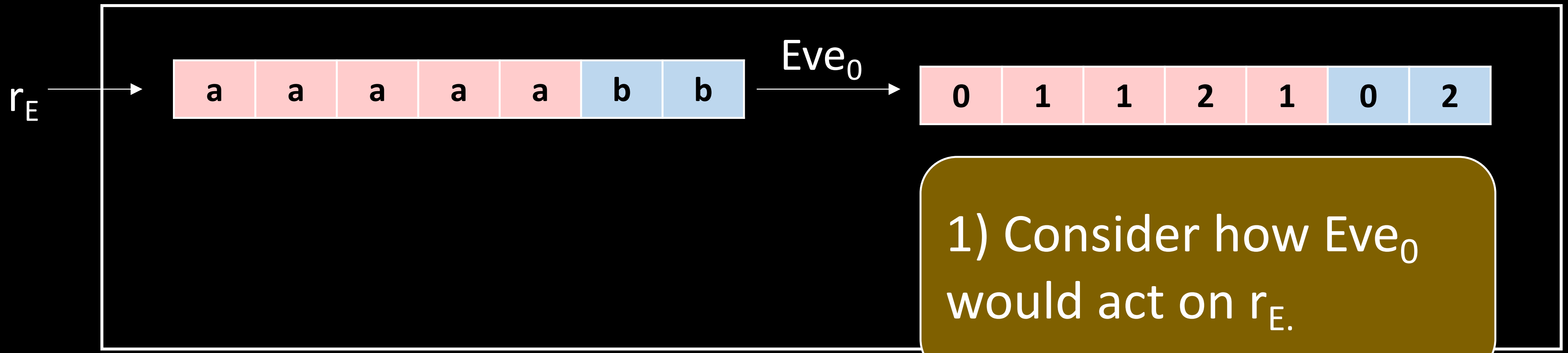
H1: Randomize Eve_0

Let Eve_1 be the following strategy:



H1: Randomize Eve_0

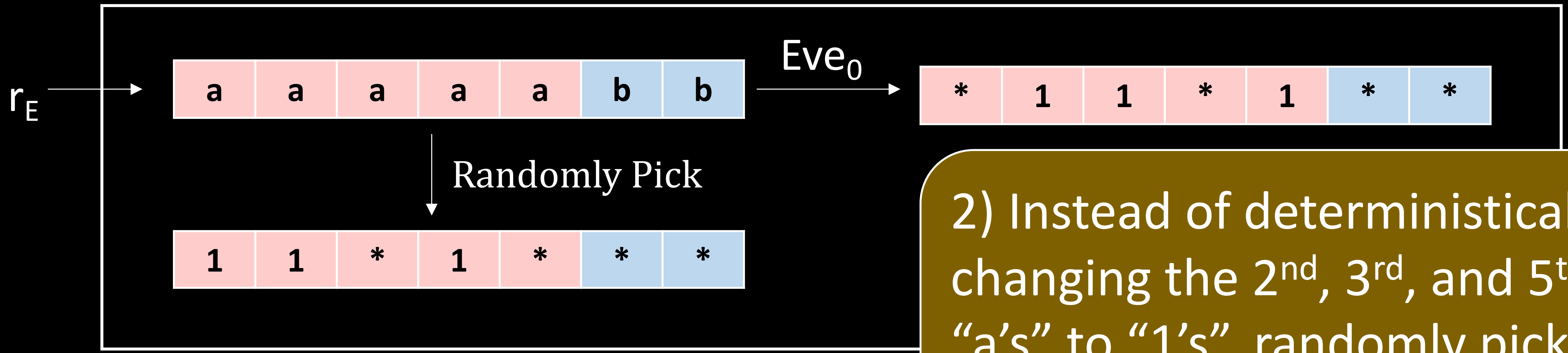
Let Eve_1 be the following strategy:



1) Consider how Eve_0 would act on $r_{E.}$

H1: Randomize Eve_0

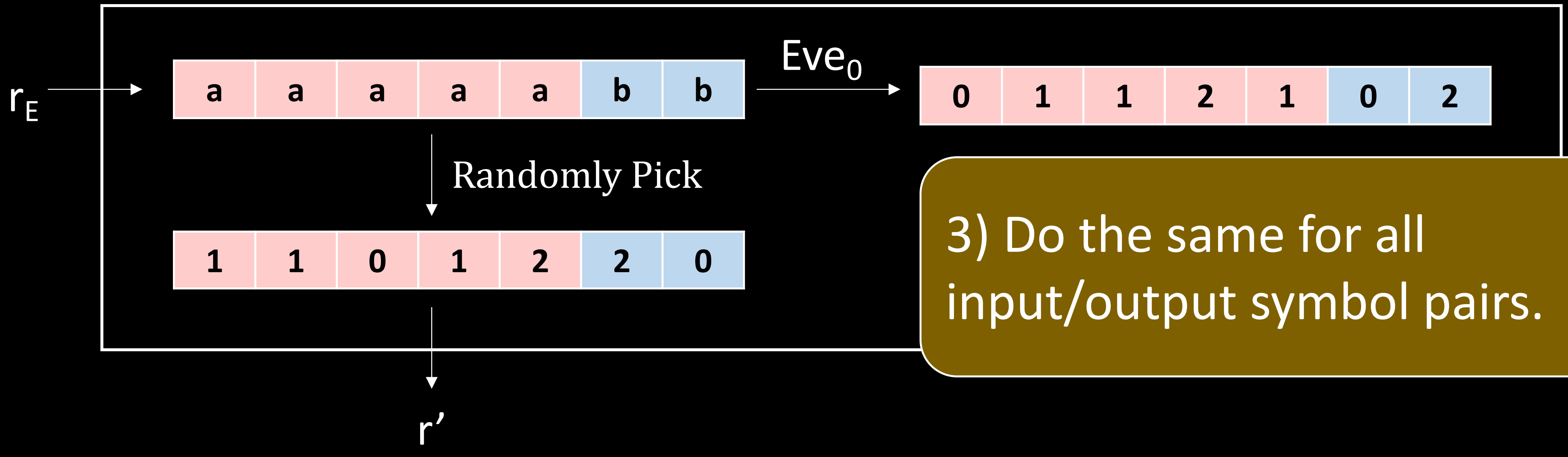
Let Eve_1 be the following strategy:



2) Instead of deterministically changing the 2nd, 3rd, and 5th "a's" to "1's", randomly pick three "a's" to change to "1's"

H1: Randomize Eve_0

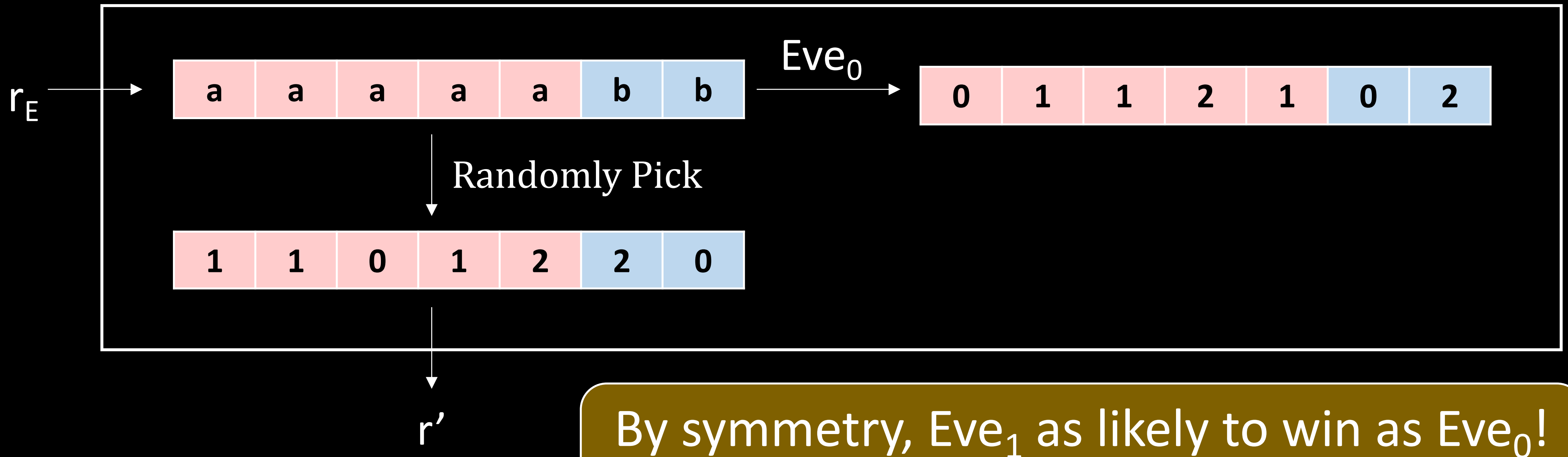
Let Eve_1 be the following strategy:



3) Do the same for all input/output symbol pairs.

H1: Randomize Eve_0

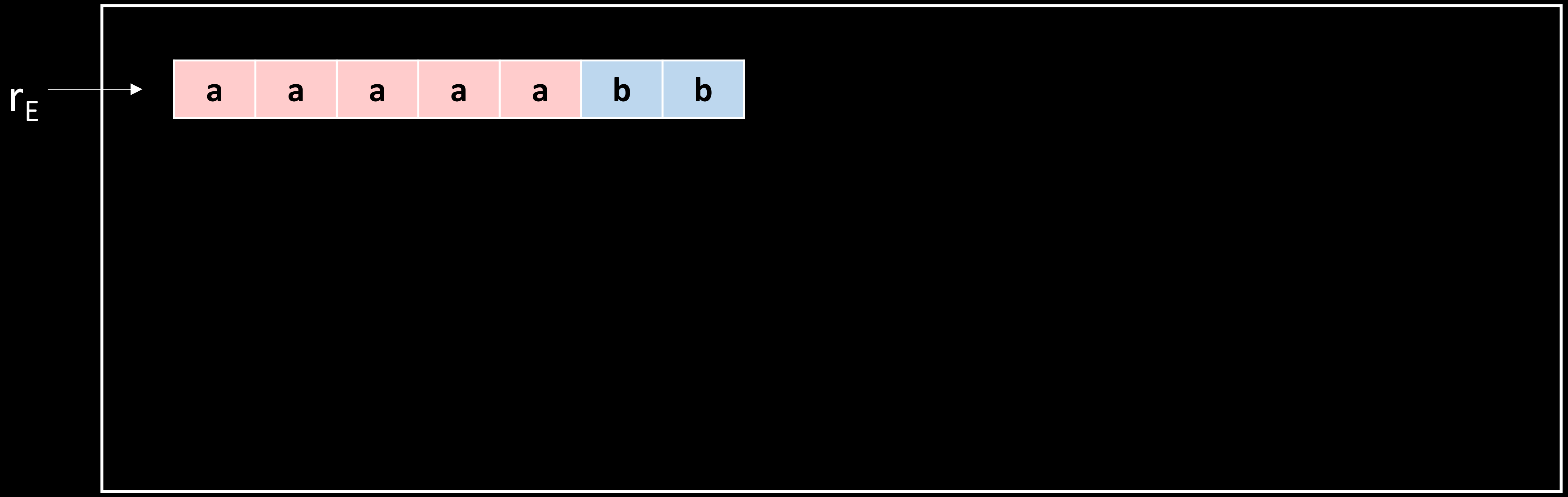
Let Eve_1 be the following strategy:



By symmetry, Eve_1 as likely to win as Eve_0!
Eve_1 is an optimal strategy!

H2: Use an Input-Dependent Channel

Let Eve_2 be the following strategy:



H2: Use an Input-Dependent Channel

Let Eve_2 be the following strategy:

$r_E \rightarrow$

a	a	a	a	a	b	b

$\xrightarrow{\text{Eve}_0}$

0	1	1	2	1	0	2

1) Consider how Eve_0 would act on $r_{E.}$

H2: Use an Input-Dependent Channel

Let Eve_2 be the following strategy:

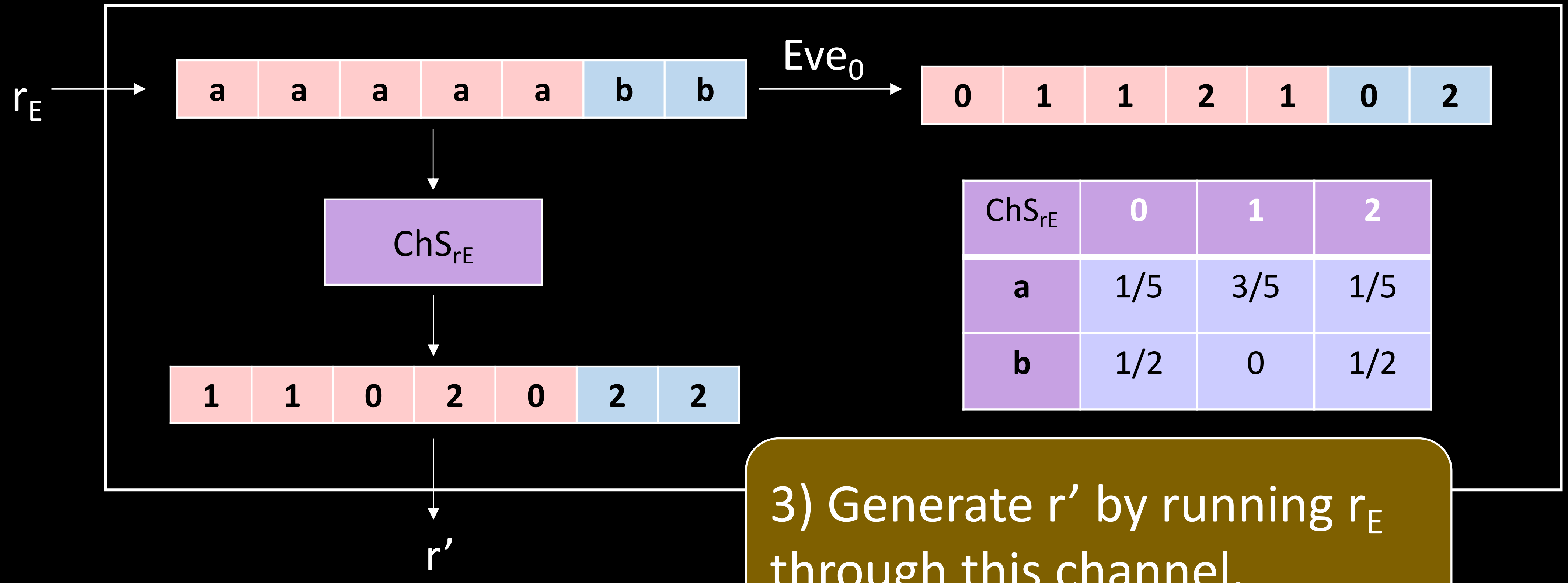
$r_E \rightarrow$ a a a a a b b $\xrightarrow{\text{Eve}_0}$ 0 1 1 2 1 0 2

2) Make a stochastic matrix (channel) representing the input/output probabilities.

ChS_{rE}	**0**	**1**	**2**
a	1/5	3/5	1/5
b	1/2	0	1/2

H2: Use an Input-Dependent Channel

Let Eve_2 be the following strategy:

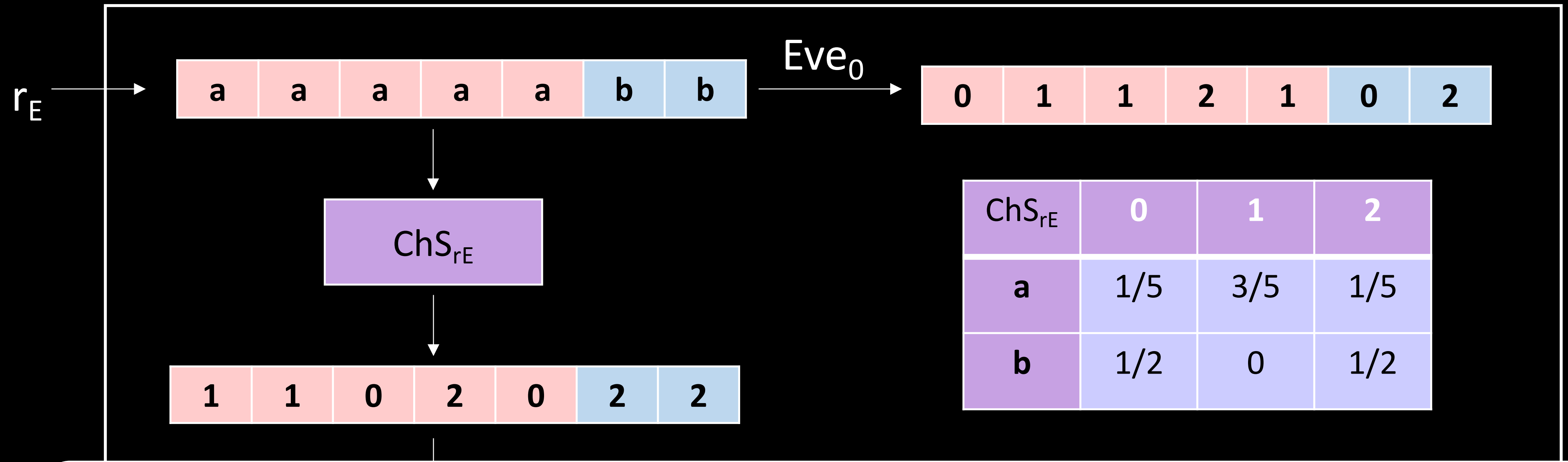


ChS_{rE}	**0**	**1**	**2**
a	1/5	3/5	1/5
b	1/2	0	1/2

3) Generate r’ by running r_E through this channel.

H2: Use an Input-Dependent Channel

Let Eve_2 be the following strategy:



ChS_{rE}	**0**	**1**	**2**
a	1/5	3/5	1/5
b	1/2	0	1/2

Security: With probability 1/poly(n), Eve_2 acts exactly the same as Eve_1!
(Probability that each input/output pair hits its expected value.)

H3: Use an Input-Independent Channel

Key Observation 1: For any permutation π,

$$\mathrm{ChS}_{rE} = \mathrm{ChS}_{\pi(rE)}$$

Eve_0 acts similarly on all vectors of the same weight!

H3: Use an Input-Independent Channel

Key Observation 1: For any permutation π,

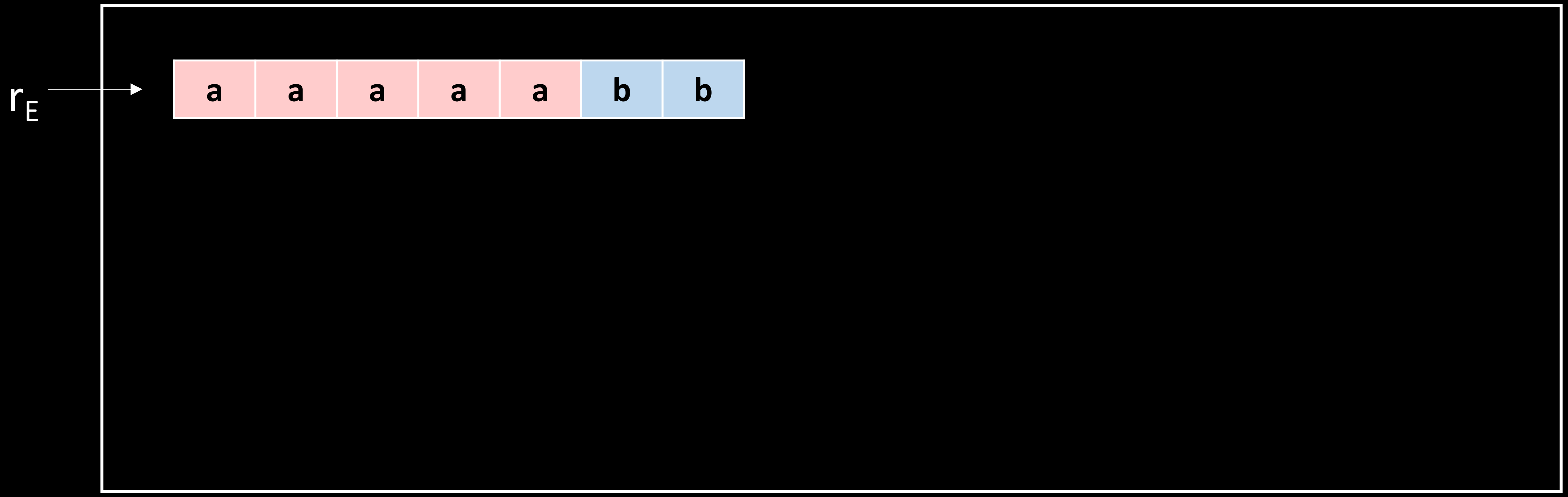
$$\mathrm{ChS}_{rE} = \mathrm{ChS}_{\pi(rE)}$$

Eve_0 acts similarly on all vectors of the same weight!

Key Observation 2: There are only poly(n) different weights.

H3: Use an Input-Independent Channel

Let Eve_3 be the following strategy:



H3: Use an Input-Independent Channel

Let Eve_3 be the following strategy:

$r_E \rightarrow$

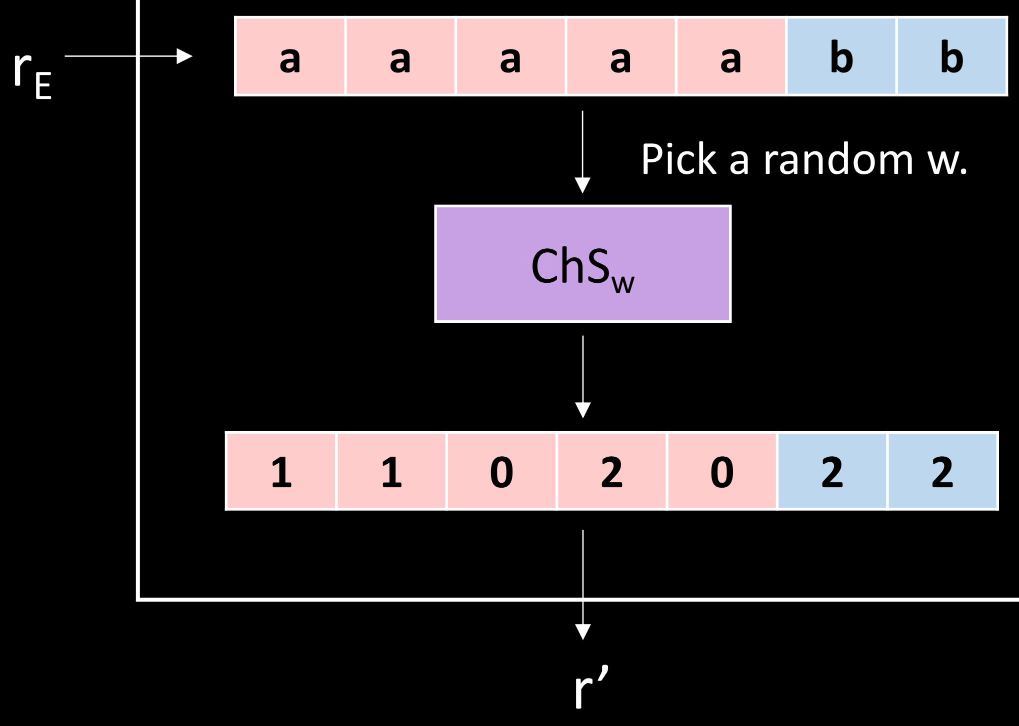
a	a	a	a	a	b	b

Pick a random w.

1) Randomly pick a weight w.

H3: Use an Input-Independent Channel

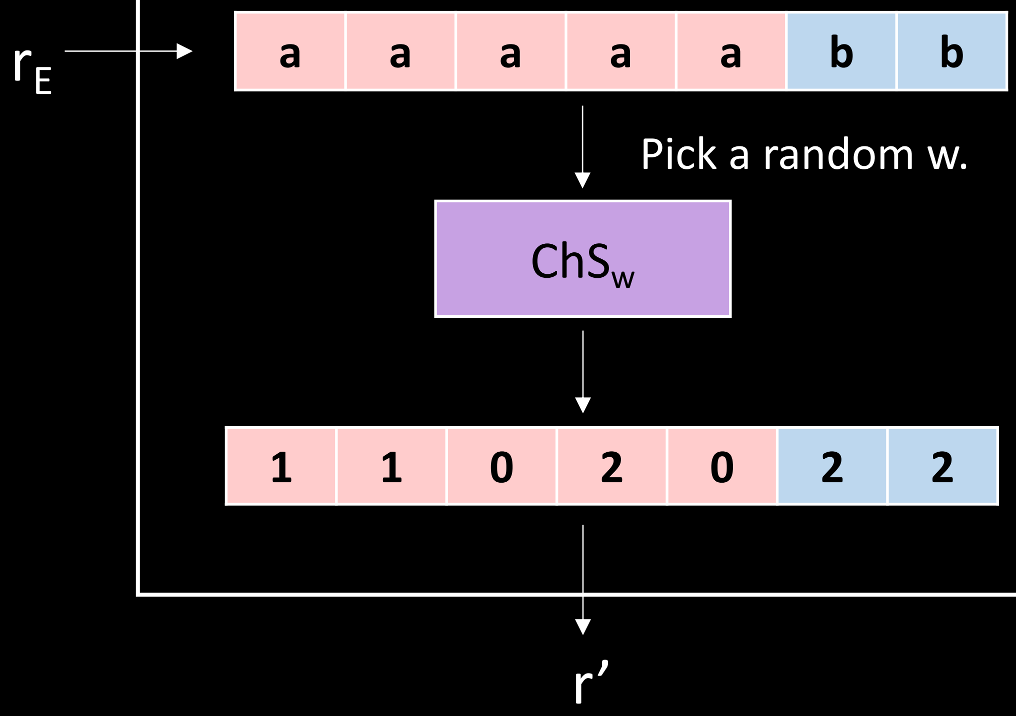
Let Eve_3 be the following strategy:



2) Generate r' by running r_E through ChS_w where $\text{ChS}_w = \text{ChS}_t$ for any string t of weight w.

H3: Use an Input-Independent Channel

Let Eve_3 be the following strategy:



Security: Since there are only poly(n) many weights, then with probability 1/poly(n),

$$ChS_w = ChS_{rE}$$

Eve_3 acts like Eve_2 in this case!

Security Summary

Goal: Show that for any strategy g, there exists a DMC ChS and a polynomial p such that

$$\Pr\left[f_r\left(g(r_E)\right) = m\right] \leq \mathrm{p}(\mathrm{n}) * \Pr\left[f_r\left(ChS(r_E)\right) = m\right] + negl(n)$$

Eve cannot do much better by using g than by using ChS!

Hybrids:

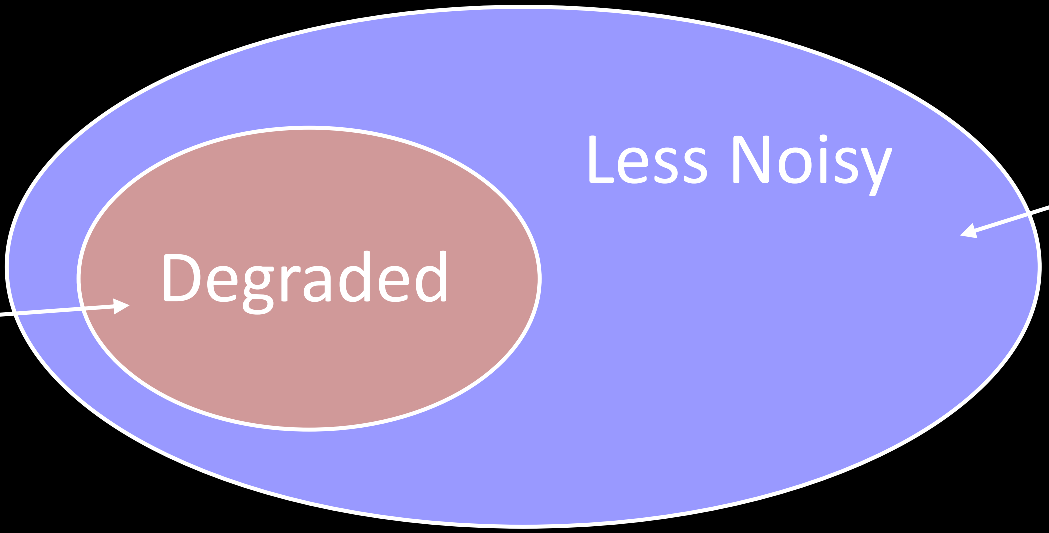
- Optimal Deterministic Strategy g*
- H_0: Add structure to g*
- H_1: Randomize
- H_2: Use an input-dependent channel.
- H_3: Use an input-independent channel (ChS).

Conclusion

Main Theorem: Assuming secure evasive function obfuscation for the class of generalized fuzzy point functions,

wiretap coding schemes are possible if and only if
ChB is not a degradation of ChE.

Impossible (even computationally)



Now possible computationally!
(Information theoretically impossible)

Conclusion

Main Theorem: Assuming secure evasive function obfuscation for the class of generalized fuzzy point functions,

wiretap coding schemes are possible if and only if ChB is not a degradation of ChE.

Extensions:

- Extends to general message spaces
- Optimal rate
- Universal encoding (encoding only depends on ChB, not ChE)

Conclusion

Main Theorem: Assuming secure evasive function obfuscation for the class of generalized fuzzy point functions,

wiretap coding schemes are possible if and only if
ChB is not a <u>degradation</u> of ChE.

Extensions:

- Extends to general message spaces
- Optimal rate
- Universal encoding (encoding only depends on ChB, not ChE)