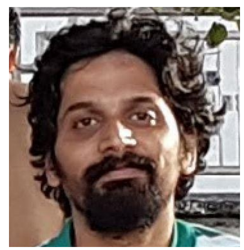
Shared Randomness in AVCs

Sagnik Bhattacharya
EE, IIT Kanpur

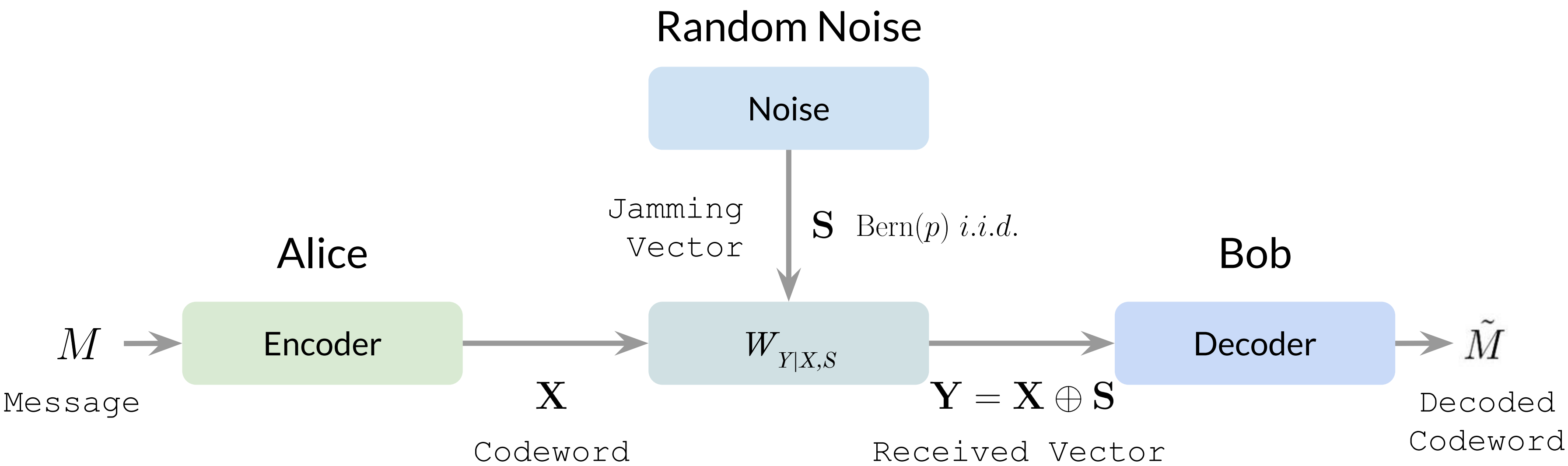
July 8, 2019
ISIT

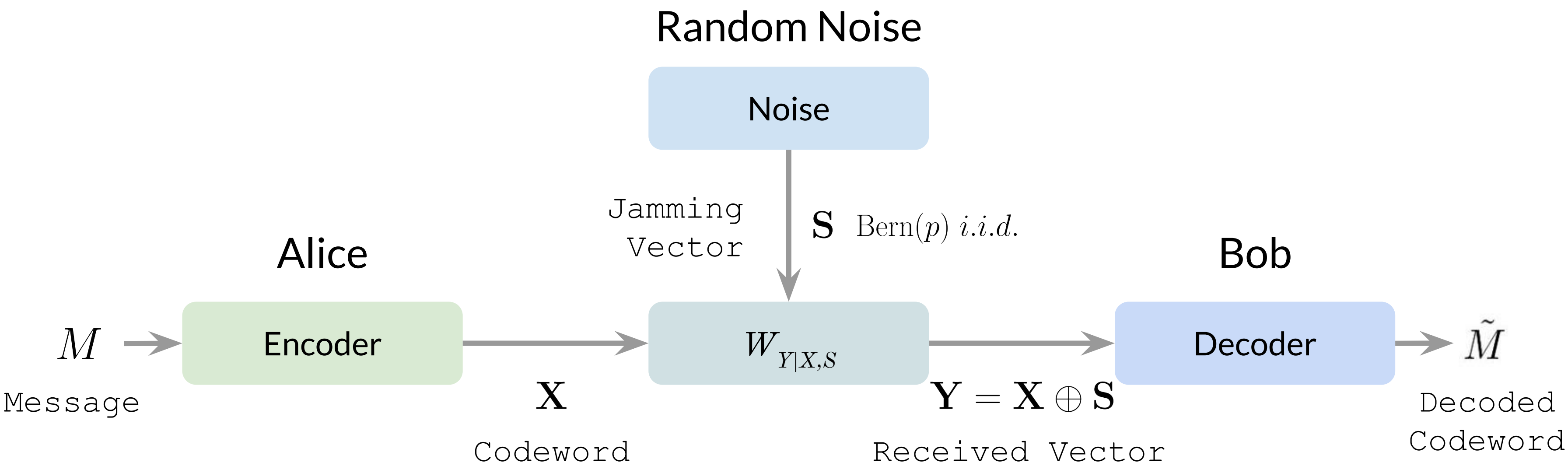

Sagnik Bhattacharya	Amitalok J. Budkuley	Sidharth Jaggi
EE@IIT Kanpur	IE@CUHK	IE@CUHK
sagnik6696@gmail.com	amitalok@ie.cuhk.edu.hk	jaggi@ie.cuhk.edu.hk

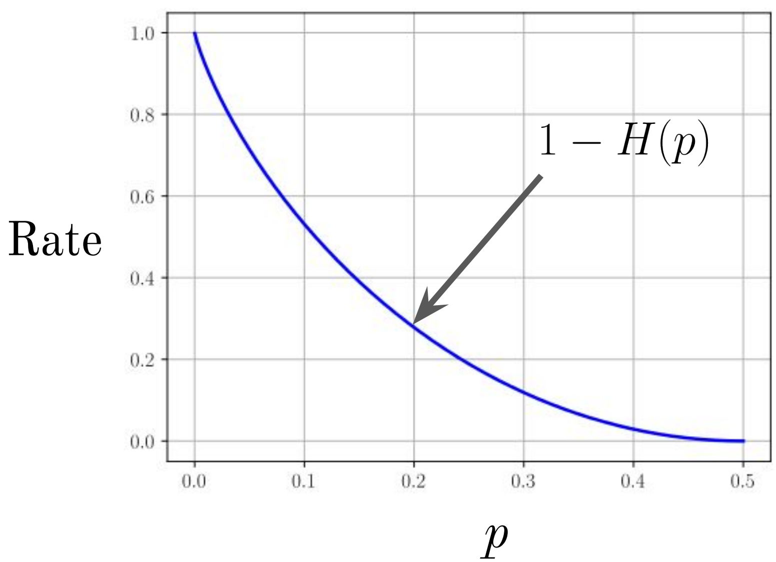
Consider the *stochastic* BSC(p)

Consider the *stochastic* BSC(p)

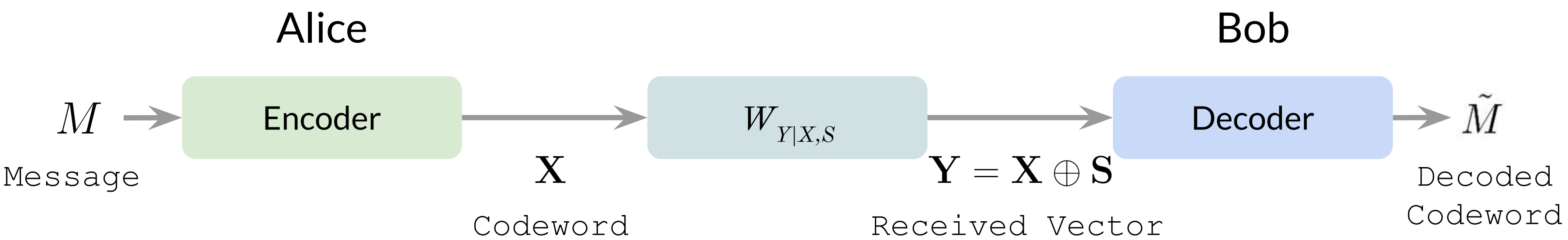


Consider the *stochastic* BSC(p)



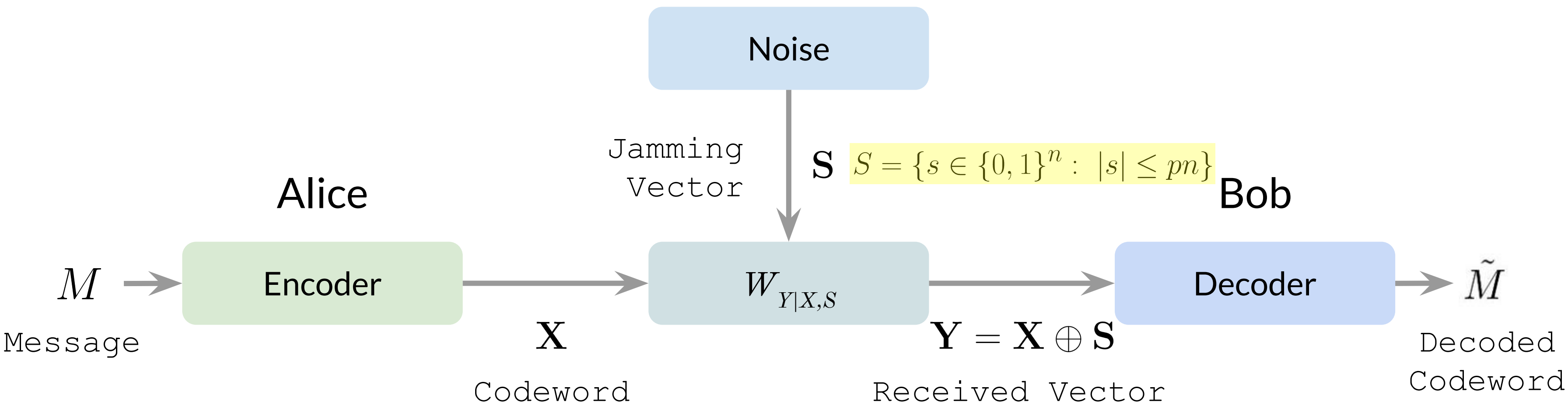


... and the *adversarial* BSC(p)



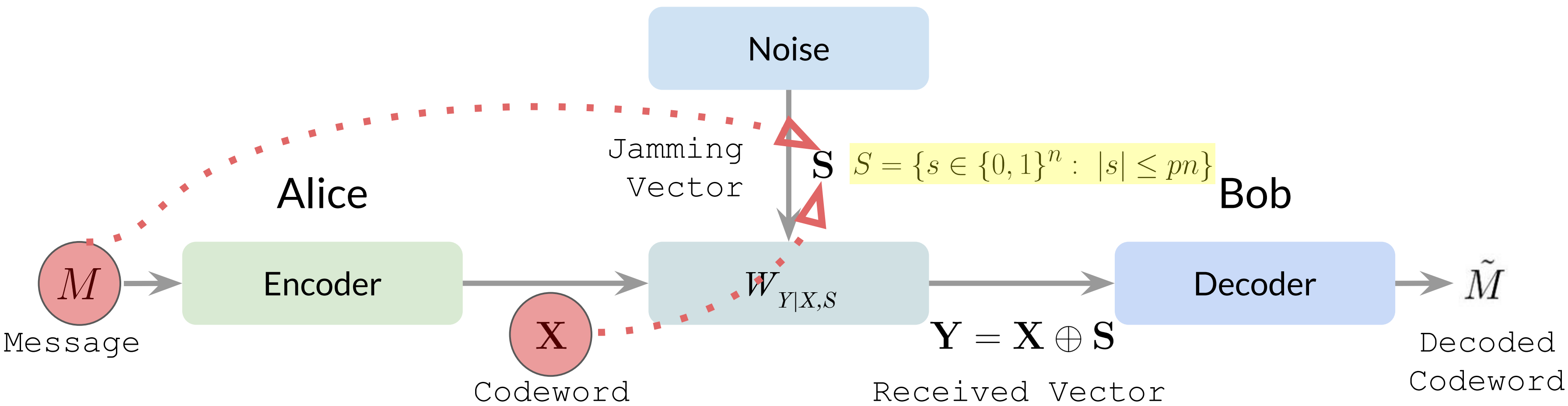
... and the *adversarial* BSC(p)

Omniscient Adversarial Noise (power limited)



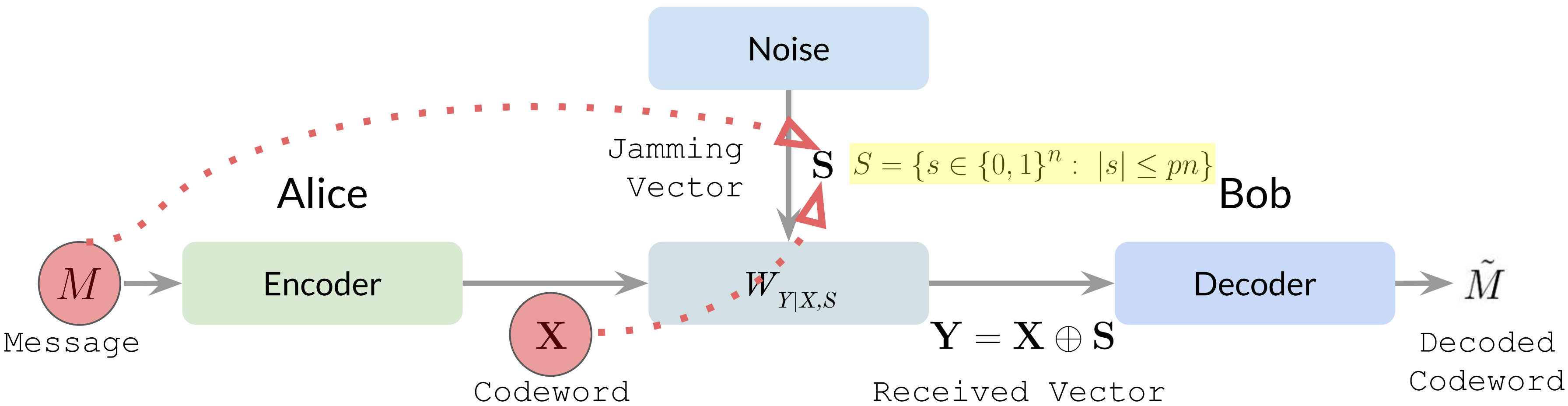
… and the *adversarial* BSC(p)

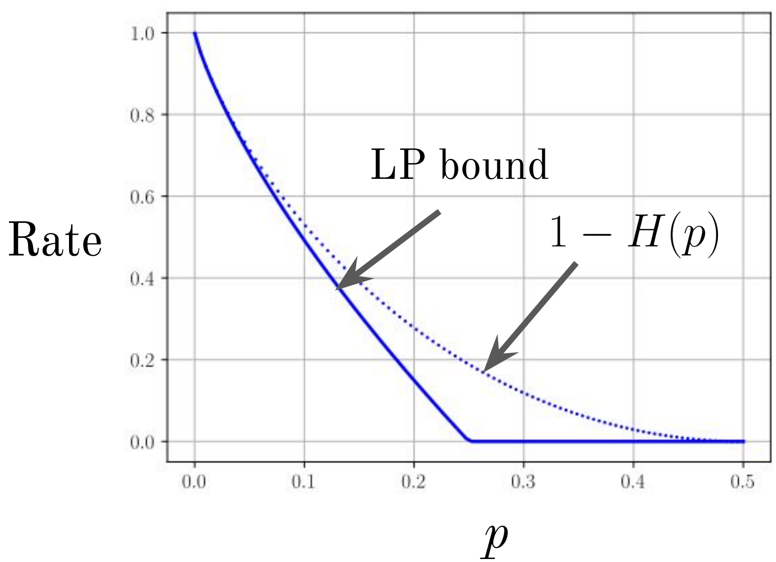
Omniscient Adversarial Noise (power limited)



… and the *adversarial* BSC(p)

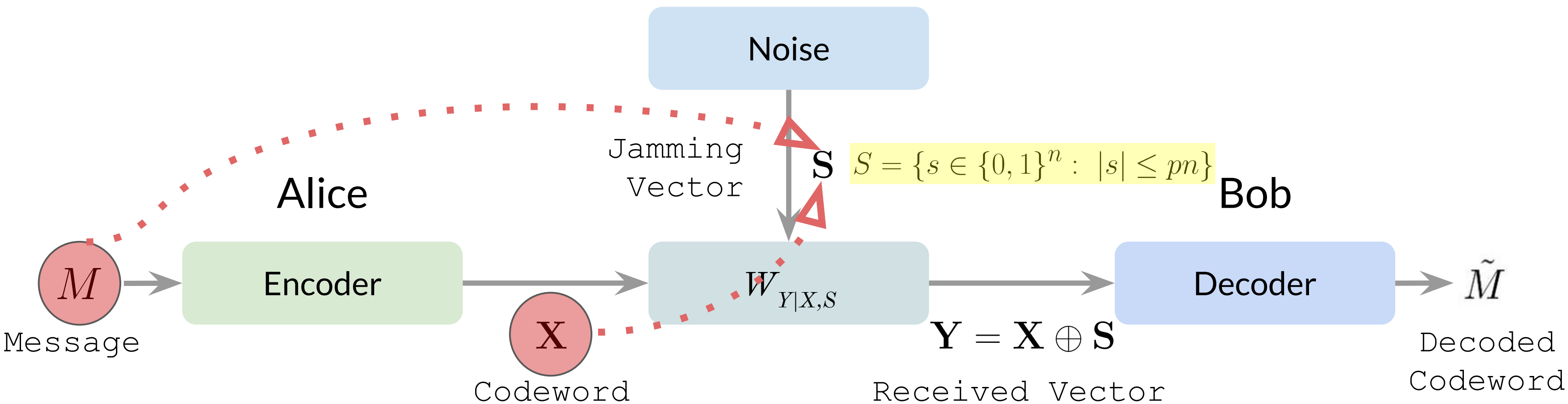
Omniscient Adversarial Noise (power limited)

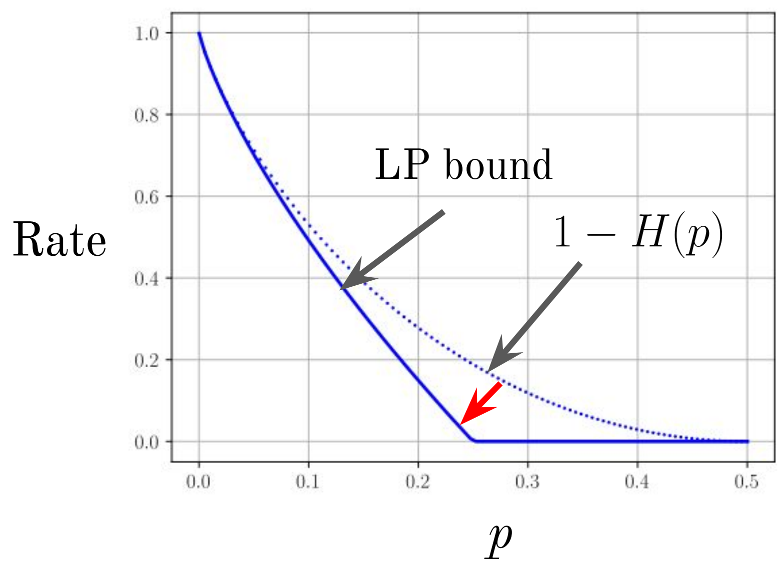




… and the *adversarial* BSC(p)

Omniscient Adversarial Noise (power limited)

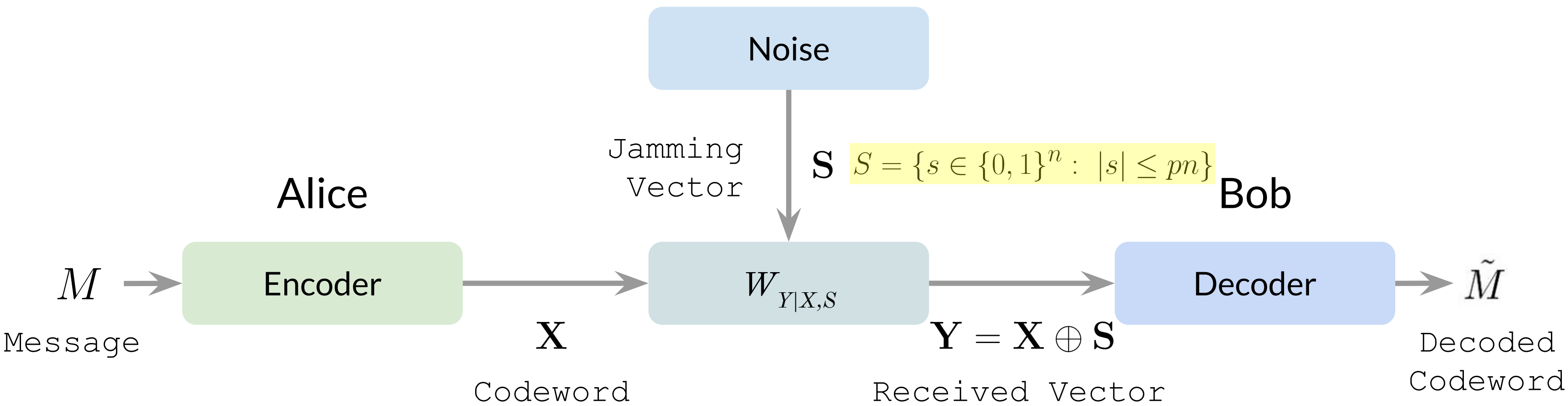


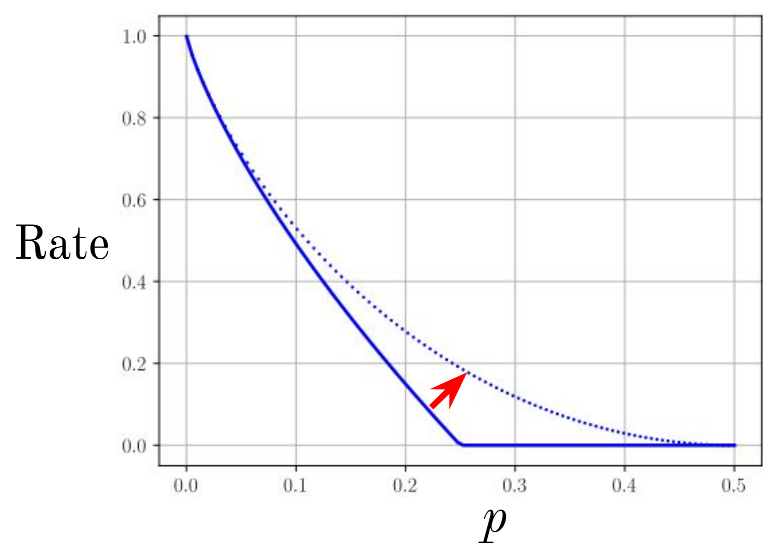


Reduction in Capacity!

How to circumvent the adversary?

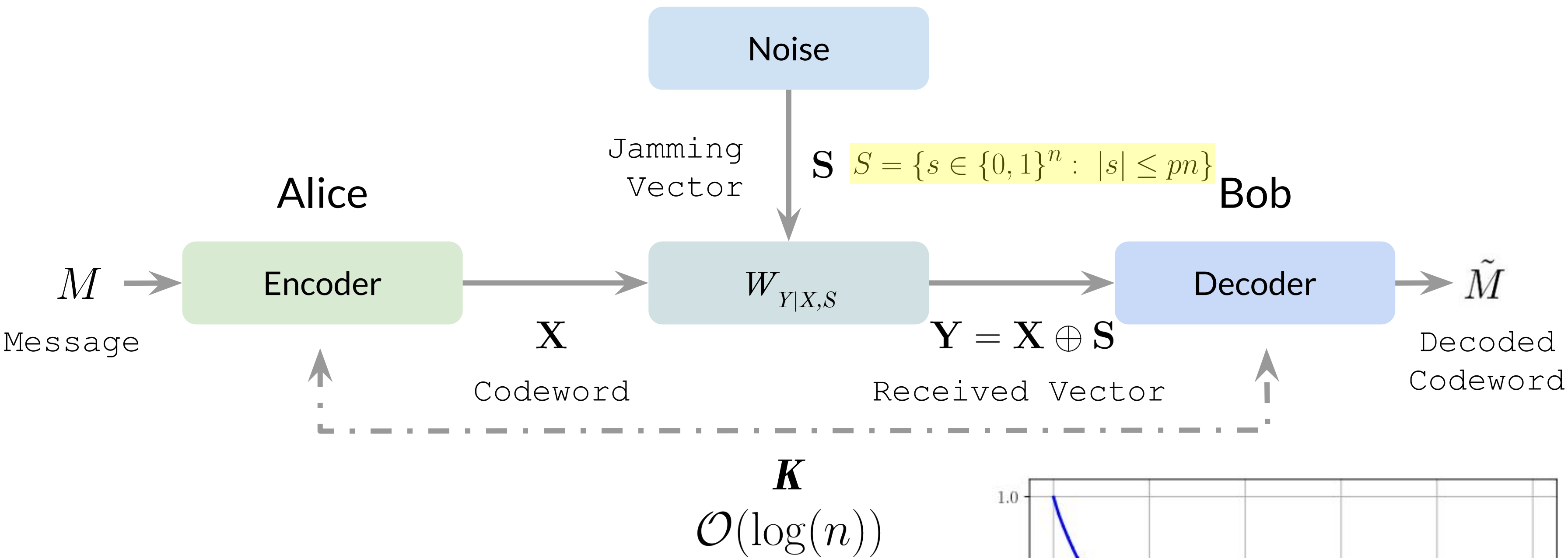
Omniscient Adversarial Noise (power limited)





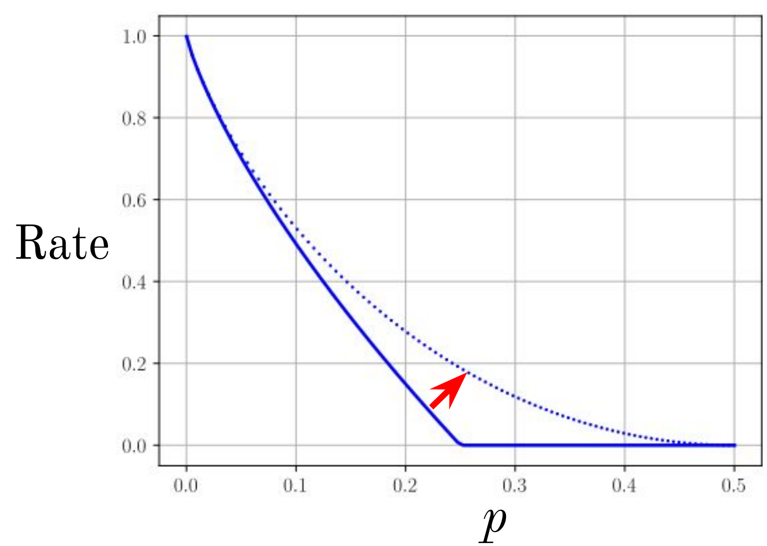
How to circumvent the adversary?

Omniscient Adversarial Noise (power limited)



An **extra** resource:
Common Randomness

Langberg (2004)



Follow up questions

How much common randomness do we require to get back to capacity?

Achievability

$$(1 + \epsilon) \log(n)$$

Converse

$$(1 - \epsilon) \log(n)$$

Follow up questions

How much common randomness do we require to get back to capacity?

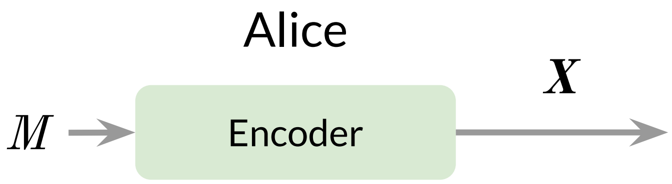
Achievability

Converse

...which we answer for more general adversarial channels

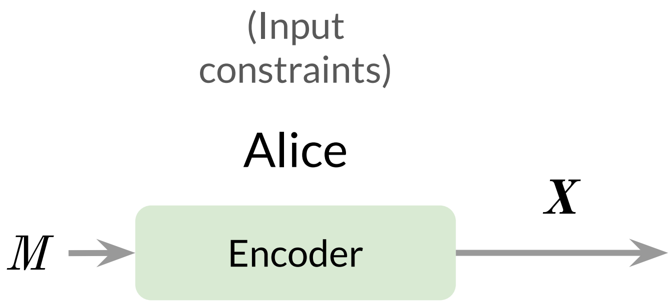
From an achievability perspective!

The Setup

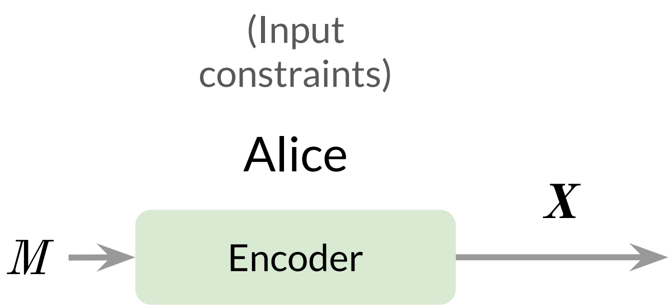


From an achievability perspective!

The Setup



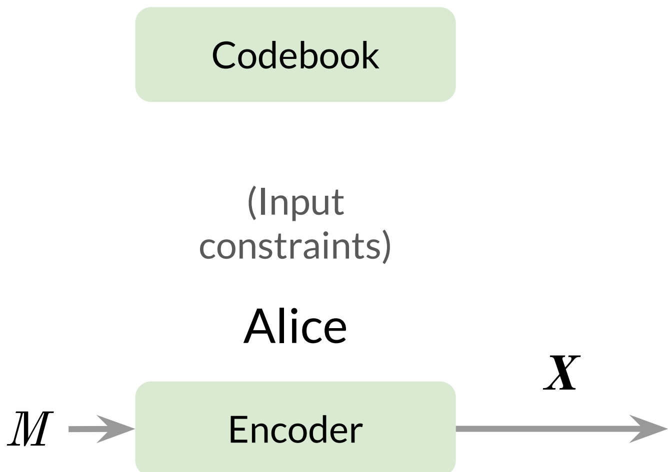
The Setup



$$\Lambda_X = \{x \in \{0,1\}^n : |x| \leq wn\}$$

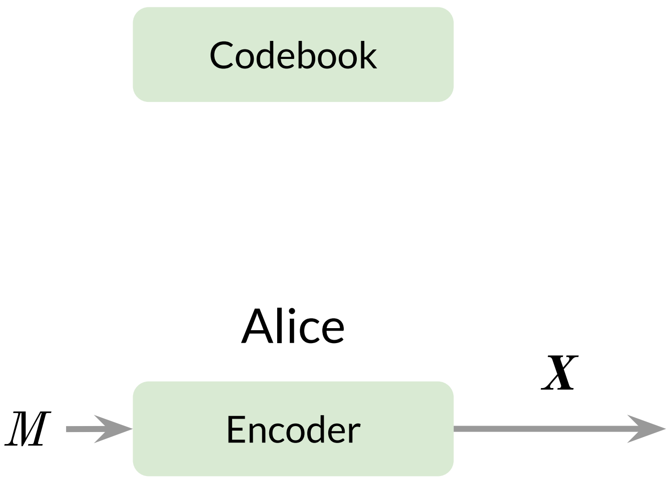
The Setup

From an achievability perspective!



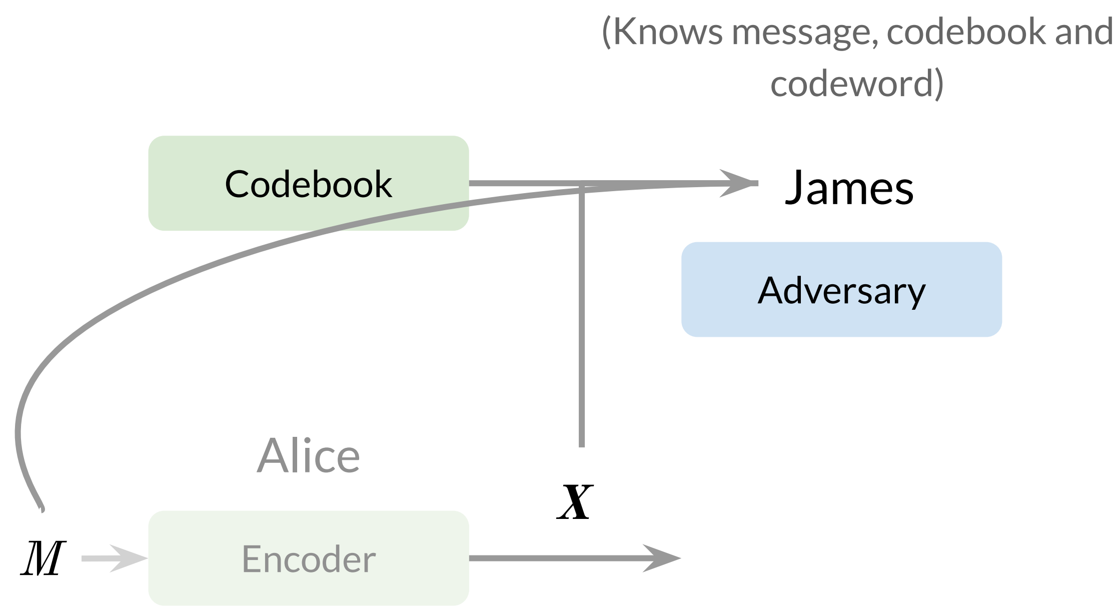
From an achievability perspective!

The Setup



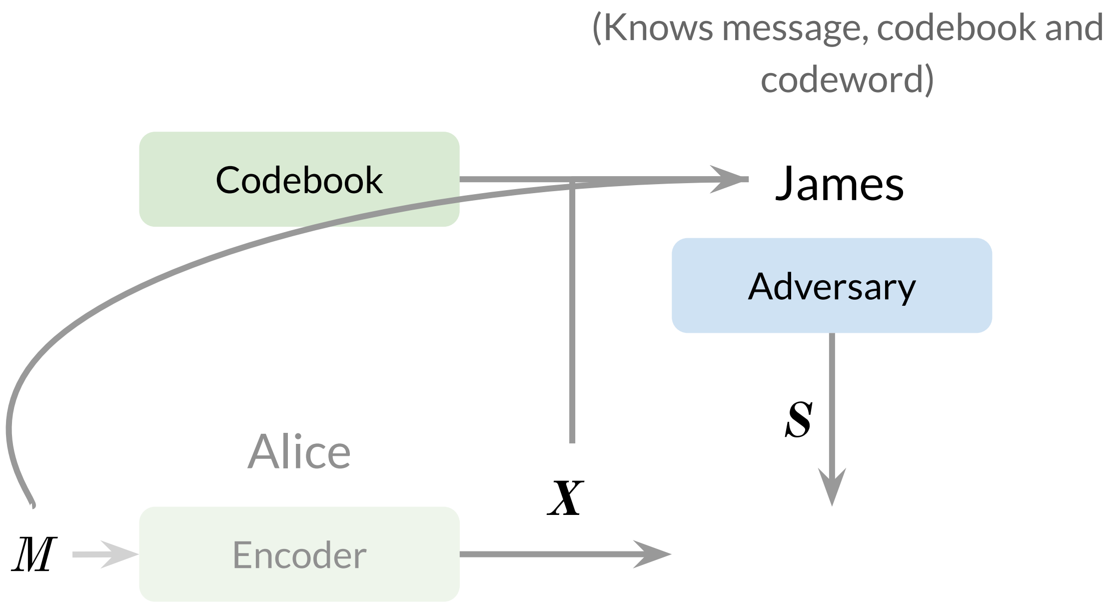
From an achievability perspective!

The Setup



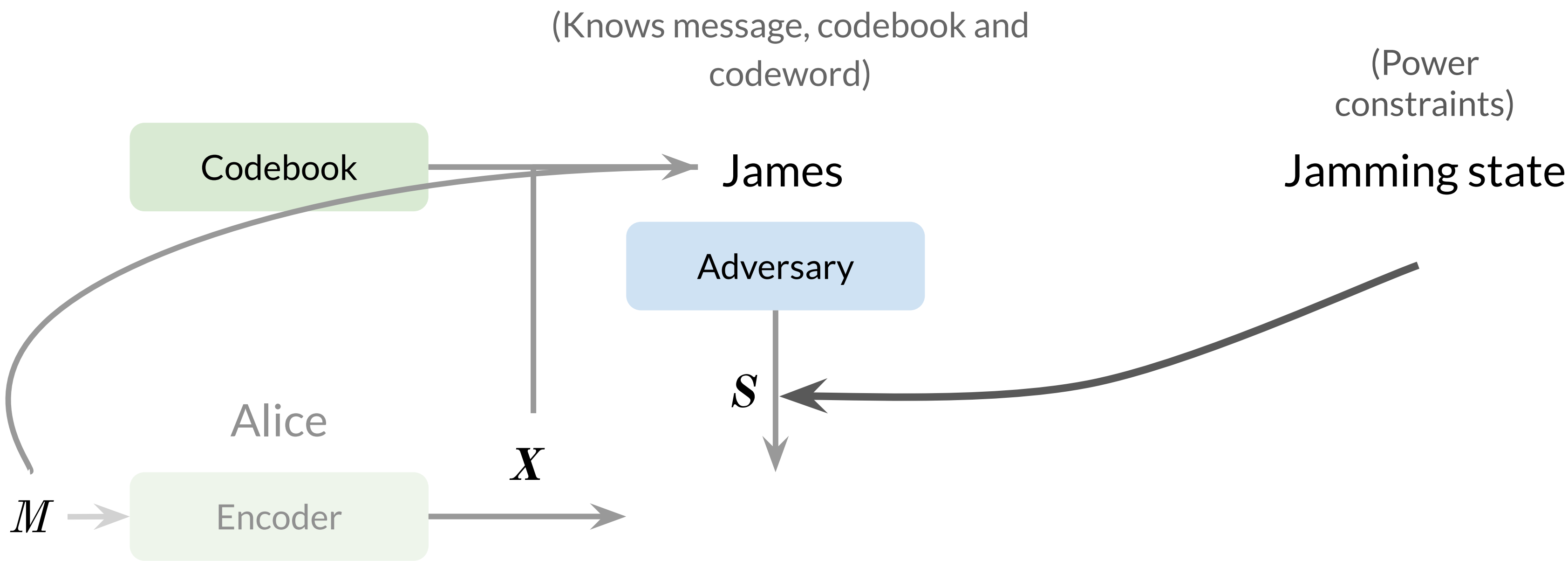
From an achievability perspective!

The Setup



From an achievability perspective!

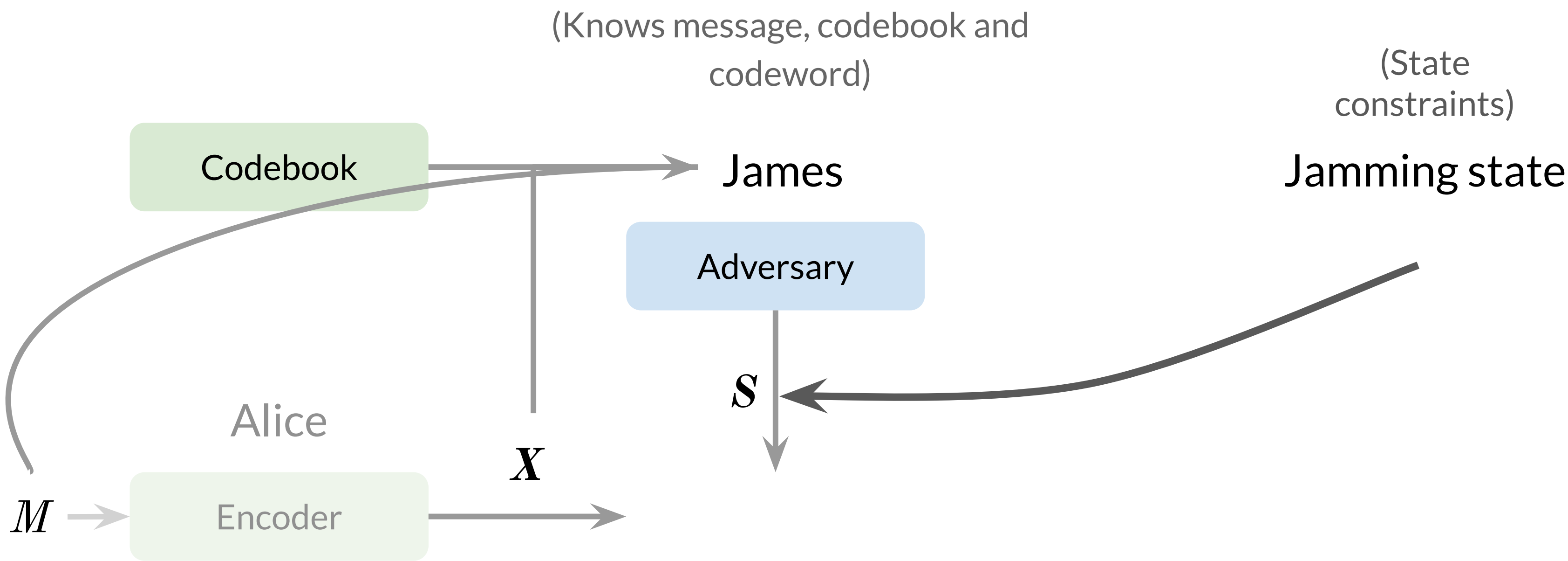
The Setup



The Setup

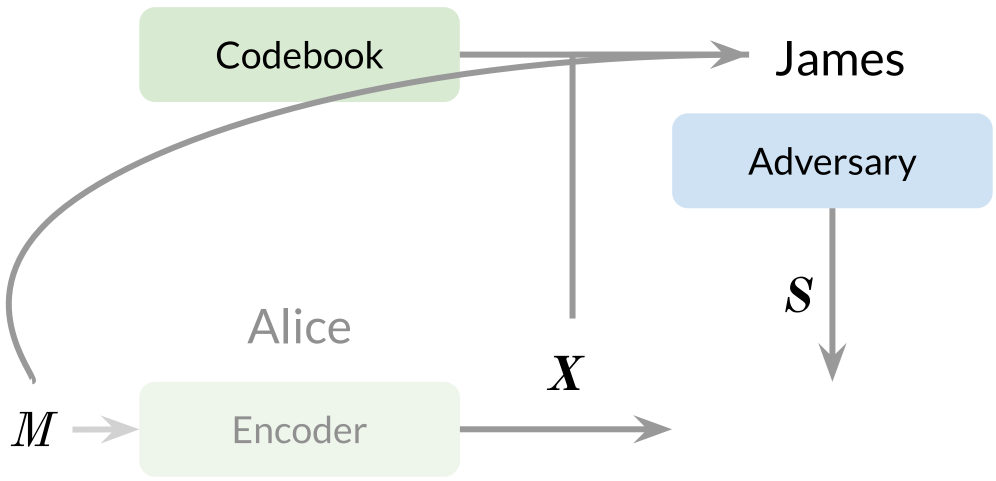
From an achievability perspective!

$\Lambda_S = \{s \in \{0,1\}^n : |s| \leq pn\}$



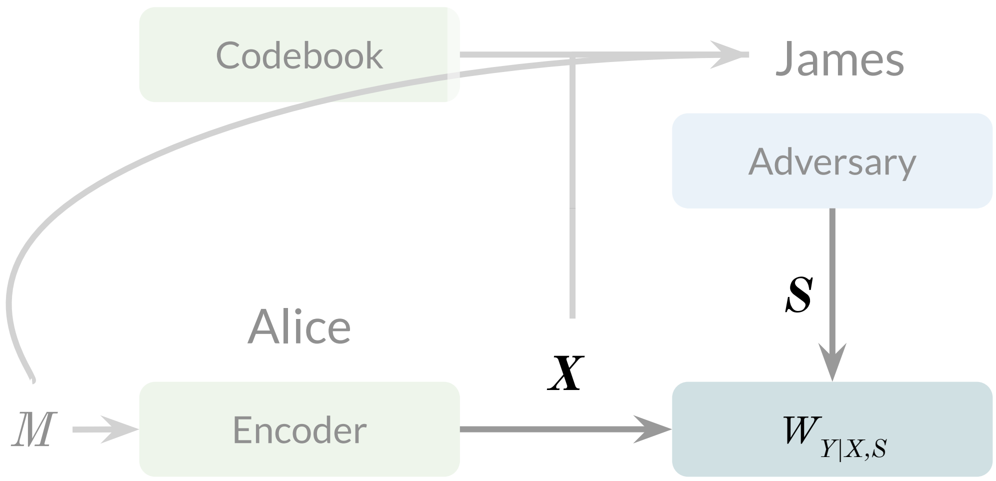
From an achievability perspective!

The Setup



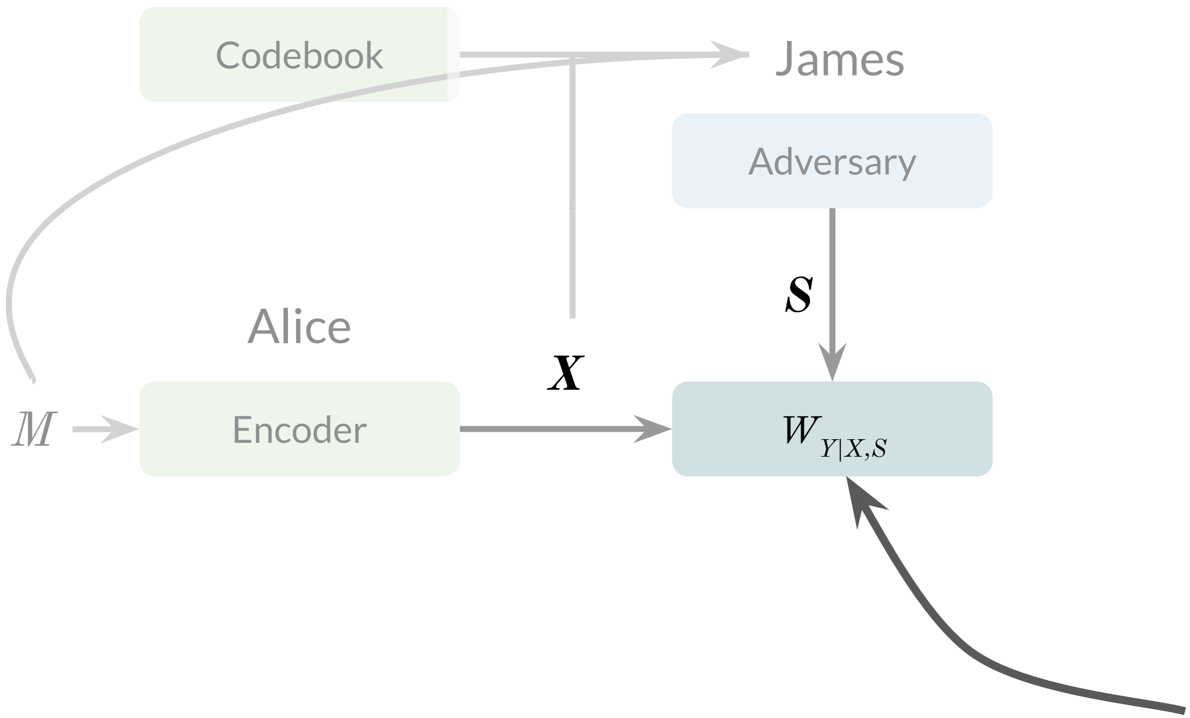
From an achievability perspective!

The Setup



From an achievability perspective!

The Setup



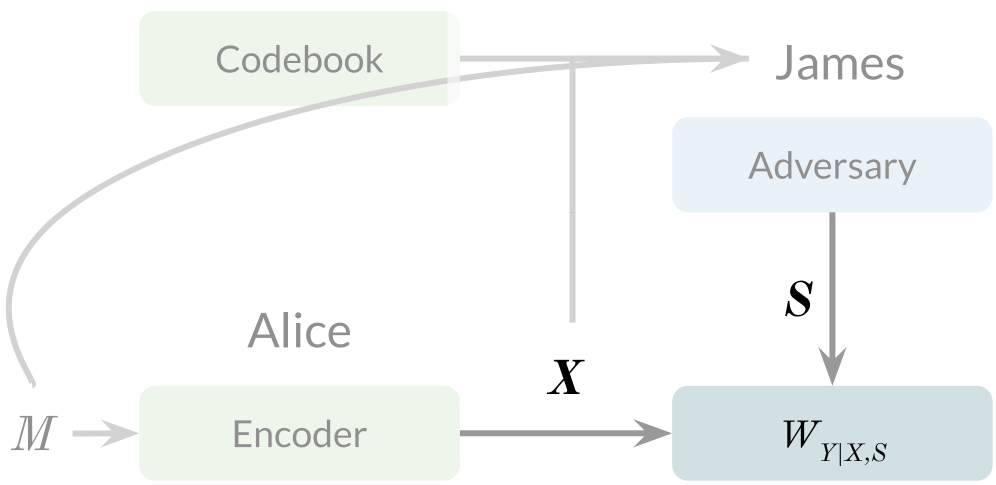
State-deterministic

(S completely specifies the channel noise)

symbol-wise

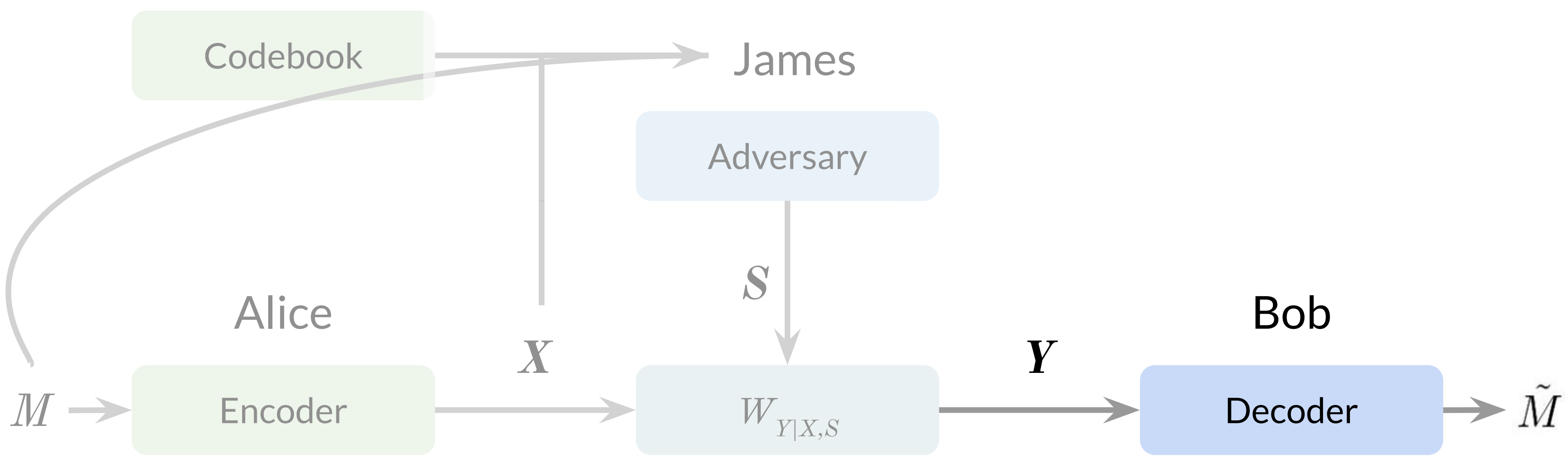
From an achievability perspective!

The Setup



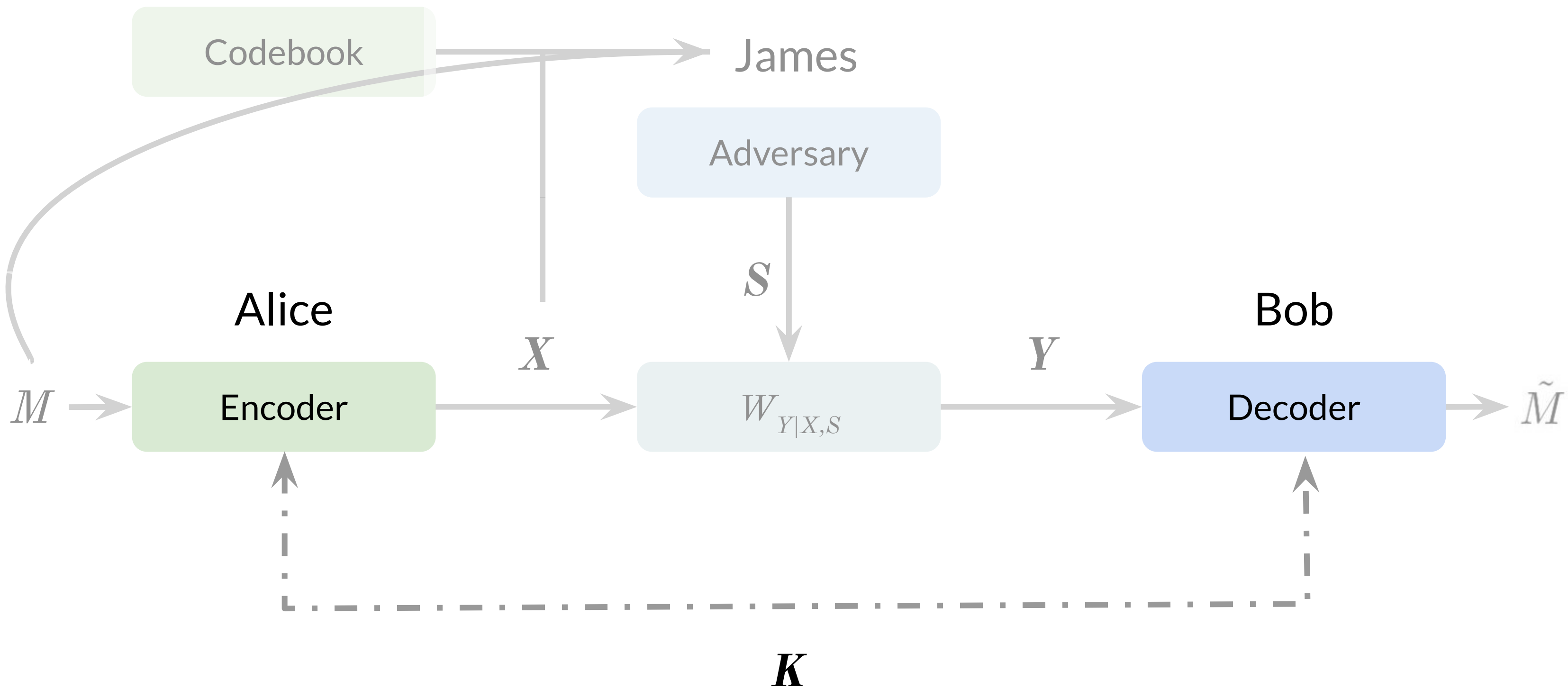
The Setup

From an achievability perspective!



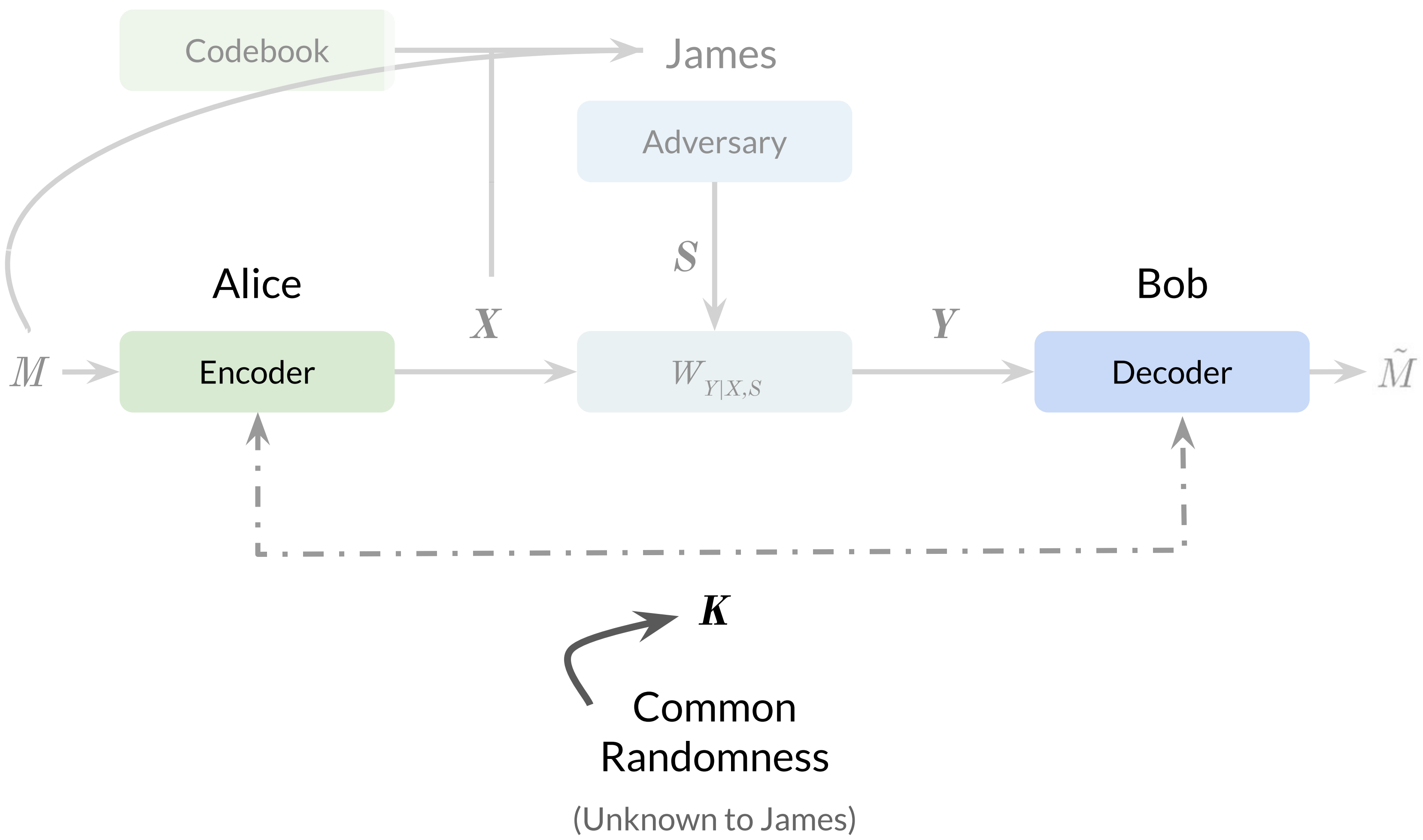
From an achievability perspective!

The Setup



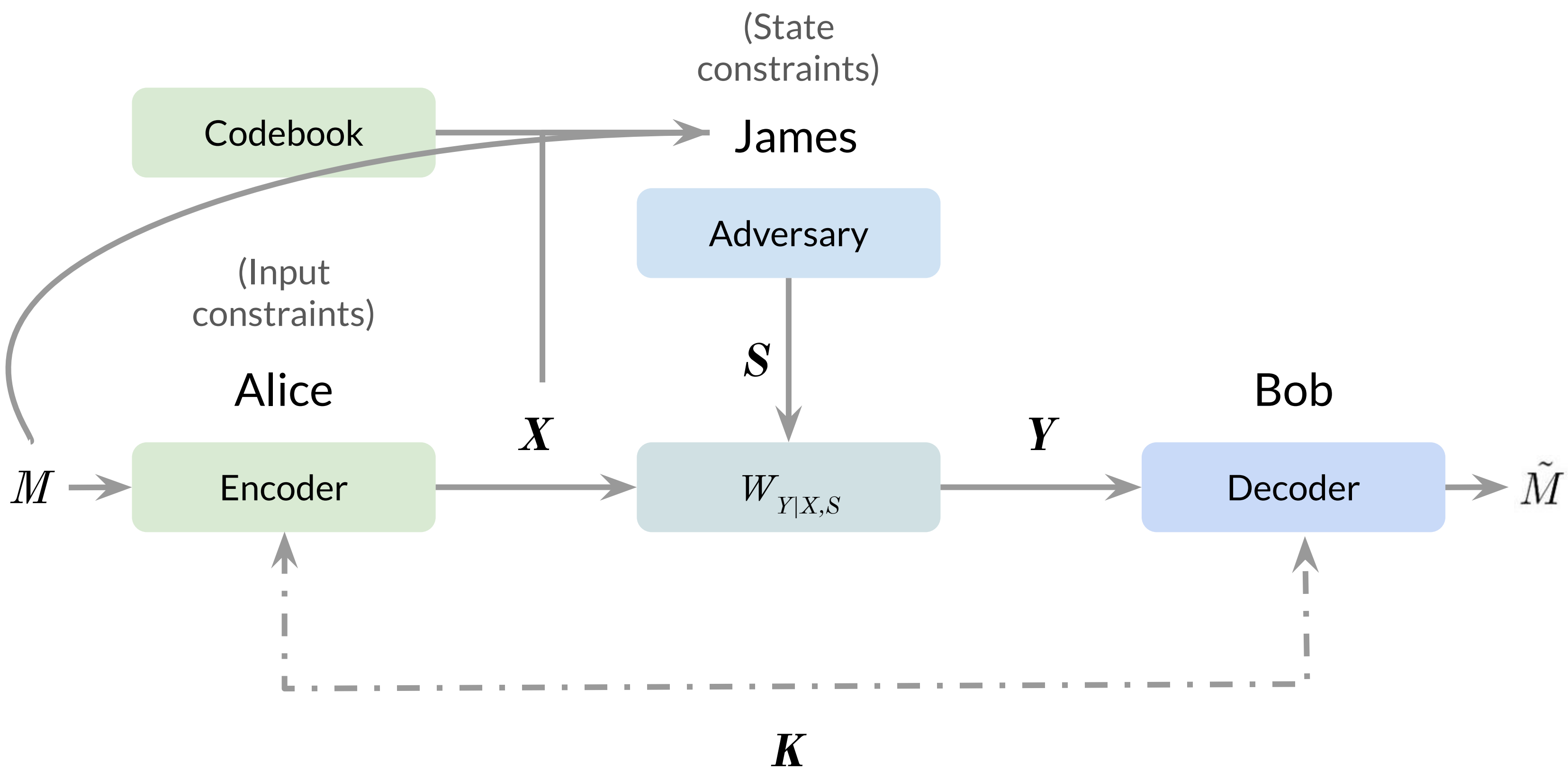
From an achievability perspective!

The Setup



The Setup

From an achievability perspective!



AVC

Input
Constraints

Channel

$$\mathcal{A} = \left(\Lambda_X, \Lambda_S, W_{Y|X,S}, \kappa\right)$$

State
Constraints

Available
Randomness

Randomised Coding Capacity

Definition: Maximum rate when unbounded common randomness is available

Theorem [Ahlswede 1986]

$$\overline{C}_r(\mathcal{A}) := \max_{P_X \in \Lambda_X} \min_{P_{S|X} : [P_X P_{S|X}]_S \in \Lambda_S} I(X;Y)$$

Reminder - Key Question

How much common randomness do we require to get to randomised coding capacity on a general AVC?

Some answers exist...

$\mathcal{O}(\log(n))$

$\mathcal{O}(\log(n))$ bits are sufficient for a fairly wide class of AVCs `[Ahlswede, 1986][Langberg, 2004]`

- Approach also used in `[Smith, 2007]`
- extended to a wide class in `[Sarwate, 2008]`

$\Omega(\log(n))$

$\Omega(\log(n))$ bits are necessary for an adversarial BSC(p) `[Langberg, 2004]`

Our contributions

Sufficiency

Using $(1+\epsilon)\log(n)$ bits of common randomness achieves capacity.

Necessity

$(1-\epsilon)\log(n)$ bits of common randomness are necessary to achieve capacity for 'adversary-weakened' AVCs.

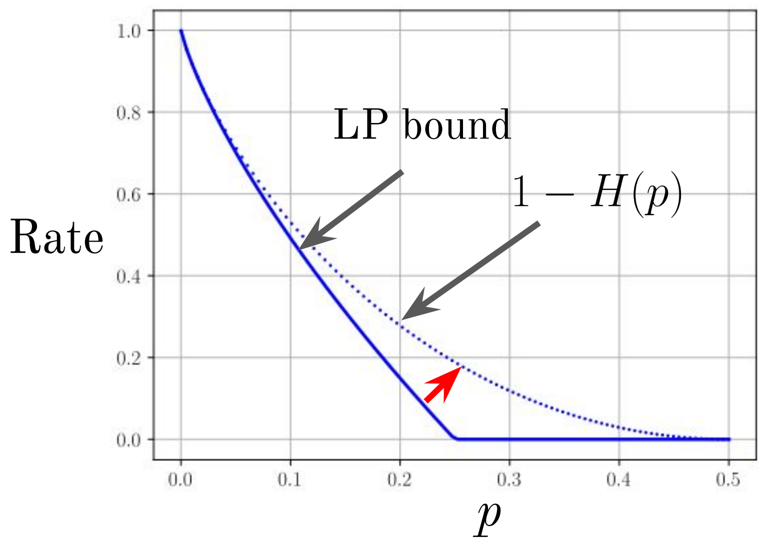
Therefore, ***precise characterisation of threshold.***

Max probability of error
metric

The Achievability

Overview

We want to increase the rate in presence of an adversary



Divide and Conquer

1 Achieving the rate

List
Decoding

2 Disambiguation

Polynomial
Hashing

List coding

Key known result:

For any $\epsilon > 0$ there exists a deterministic list code with rate

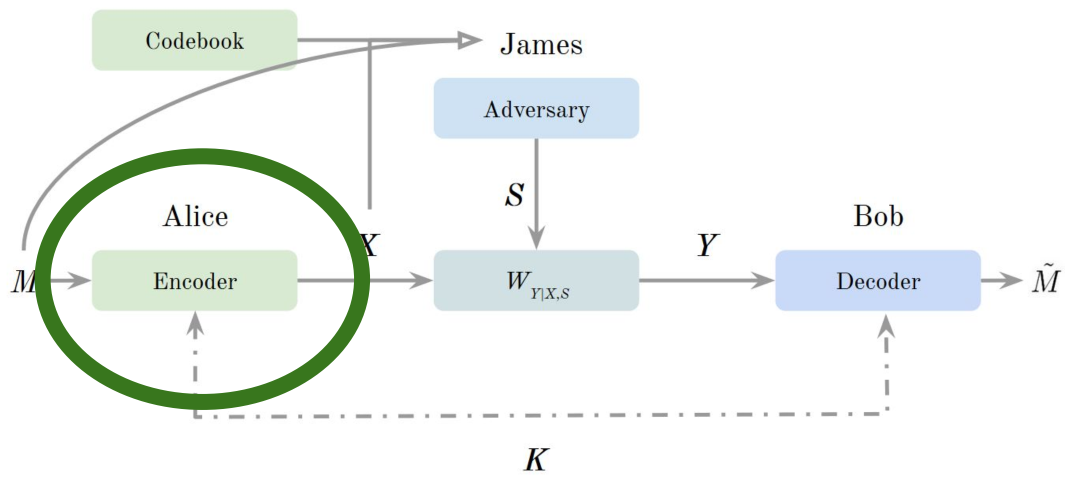
$$R = \overline{C}_r(\mathcal{A}) - \epsilon$$

and list size

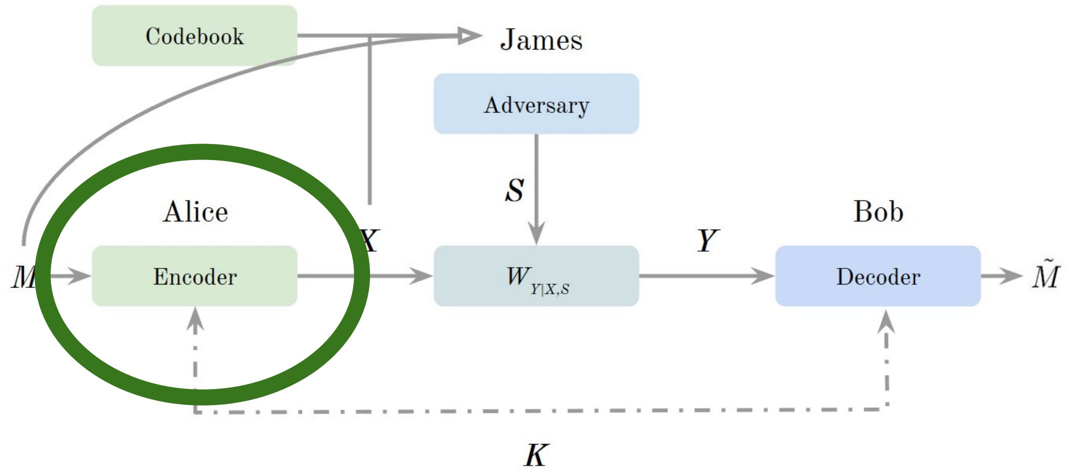
$$2 \cdot \frac{\log\left(|\text{output alphabet}|\right)}{\epsilon}$$

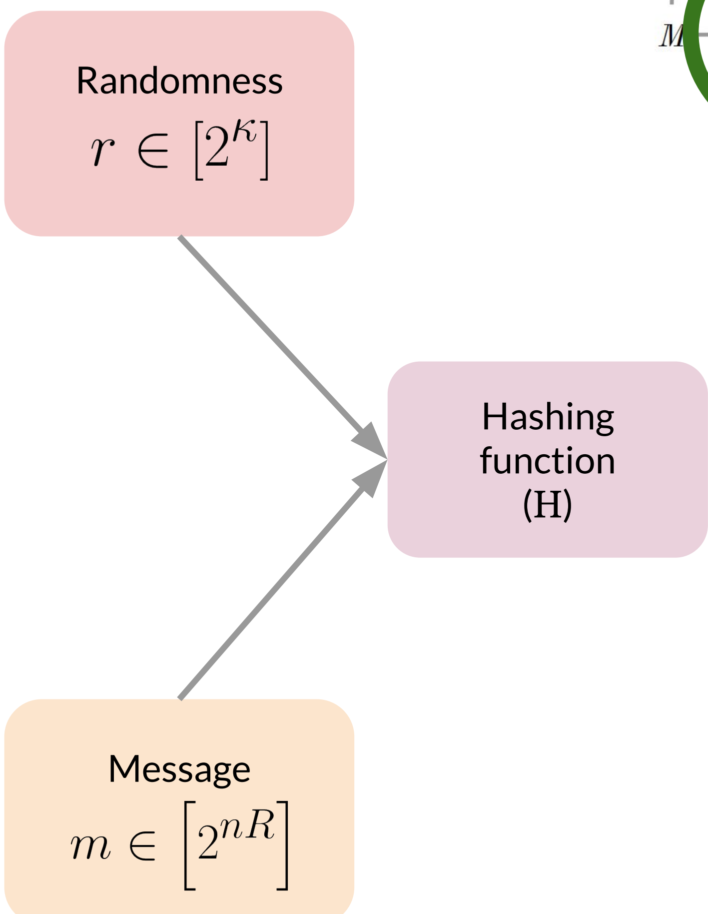
Let the code be Φ

Using list code

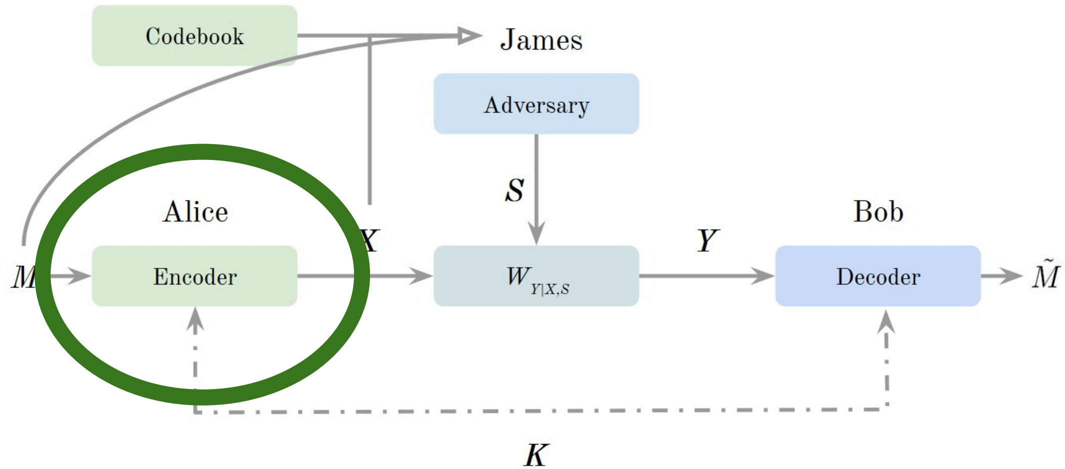


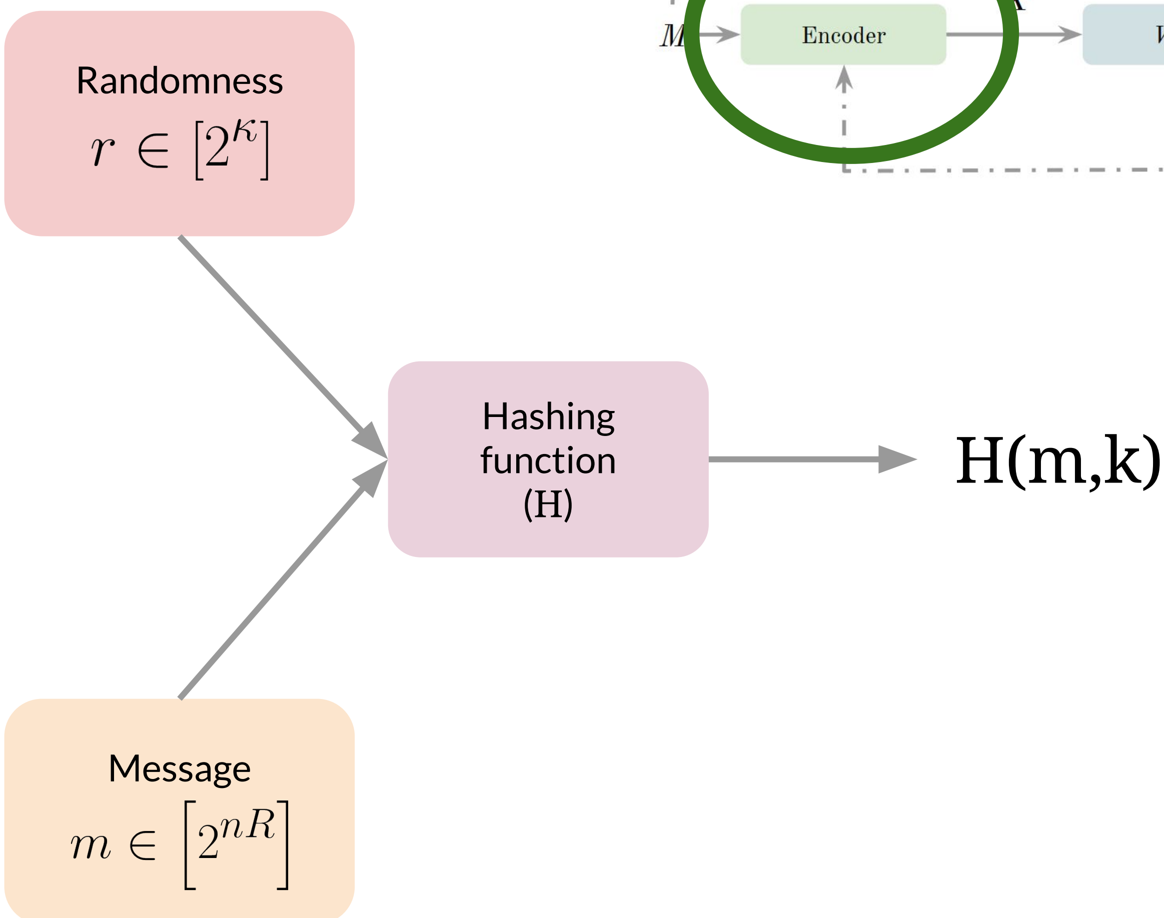
Using list code



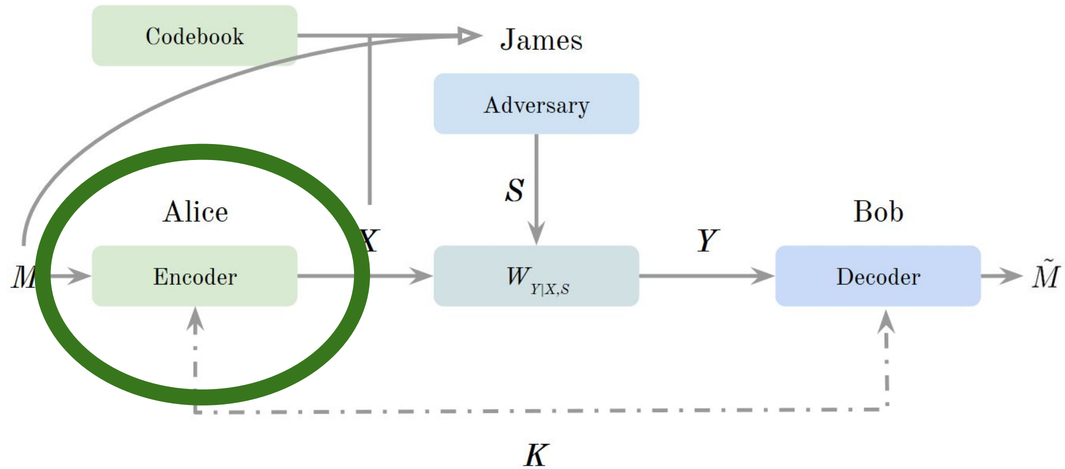


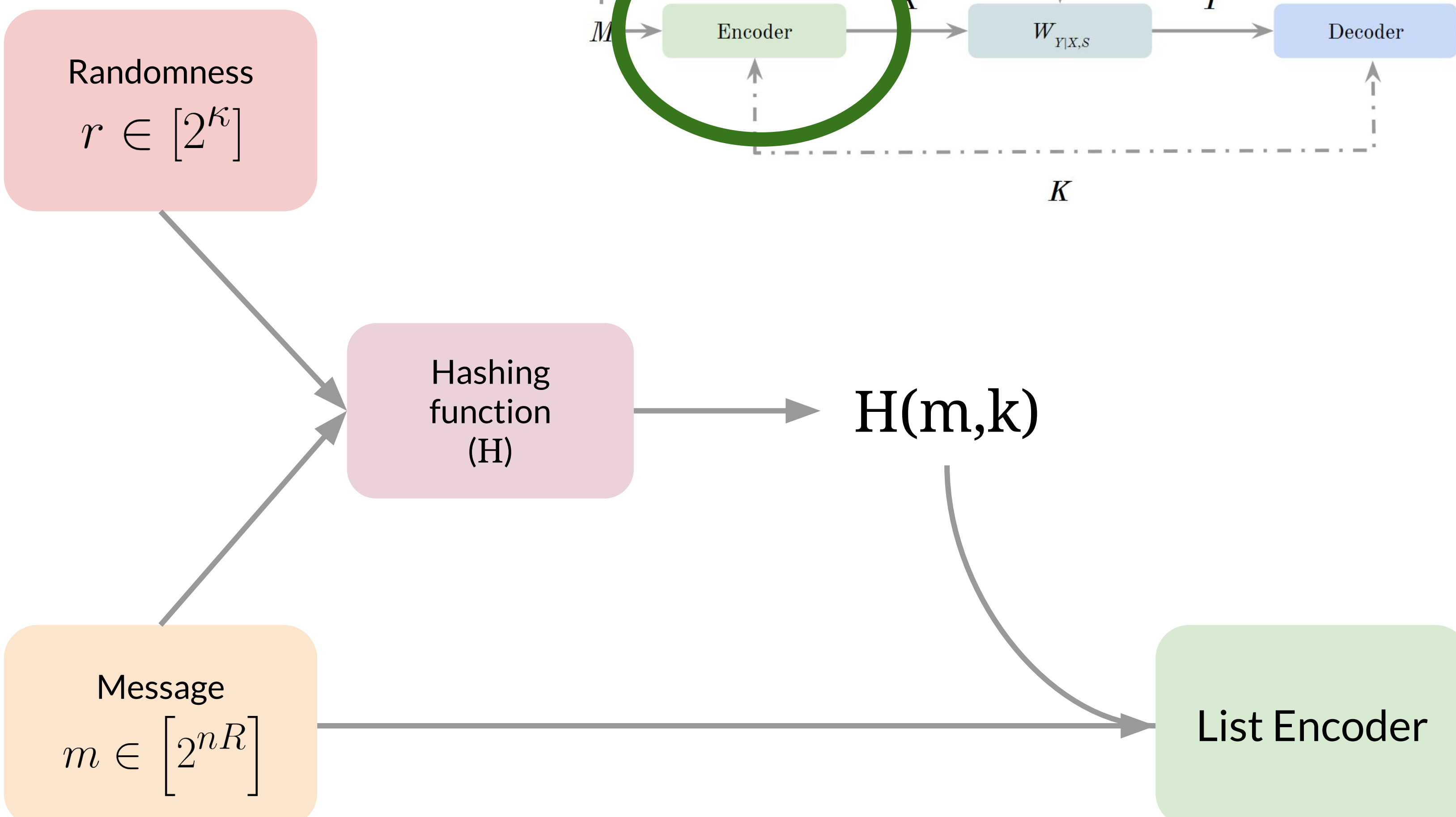
Using list code



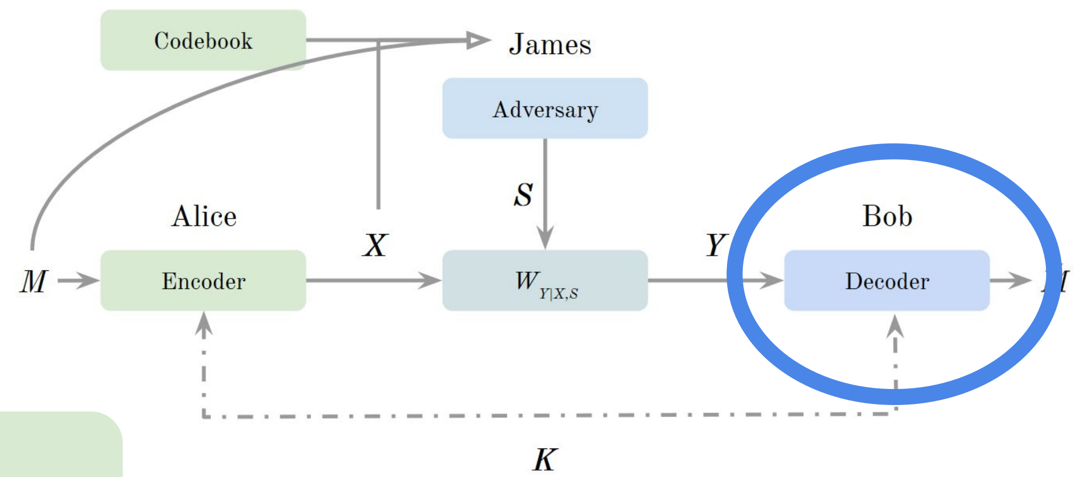


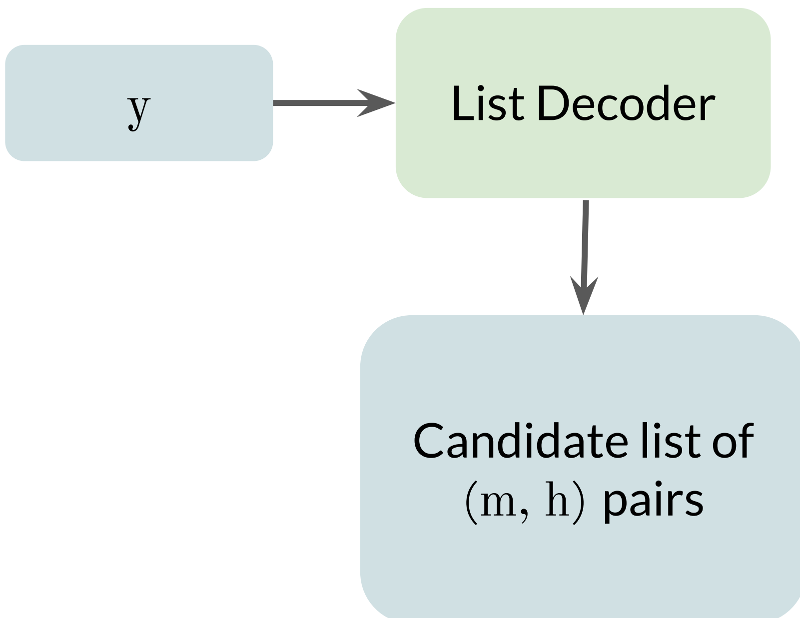
Using list code



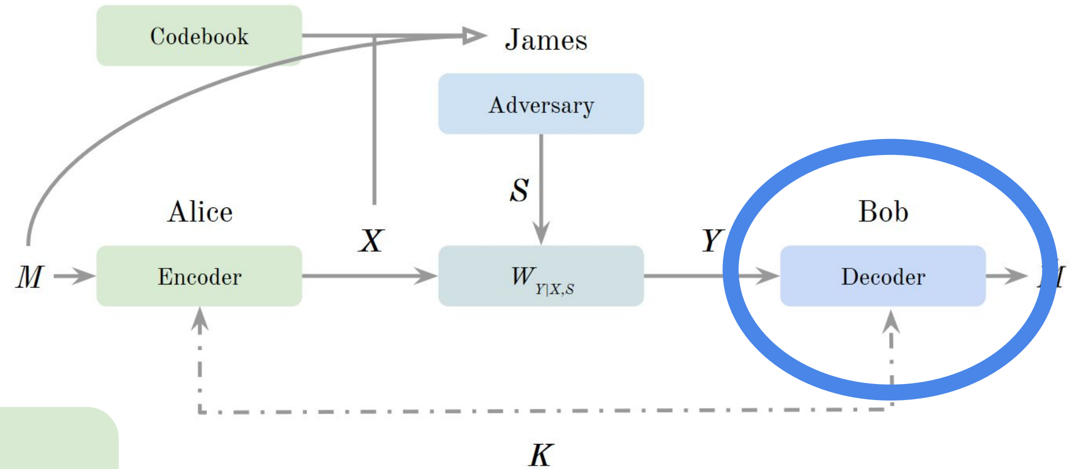


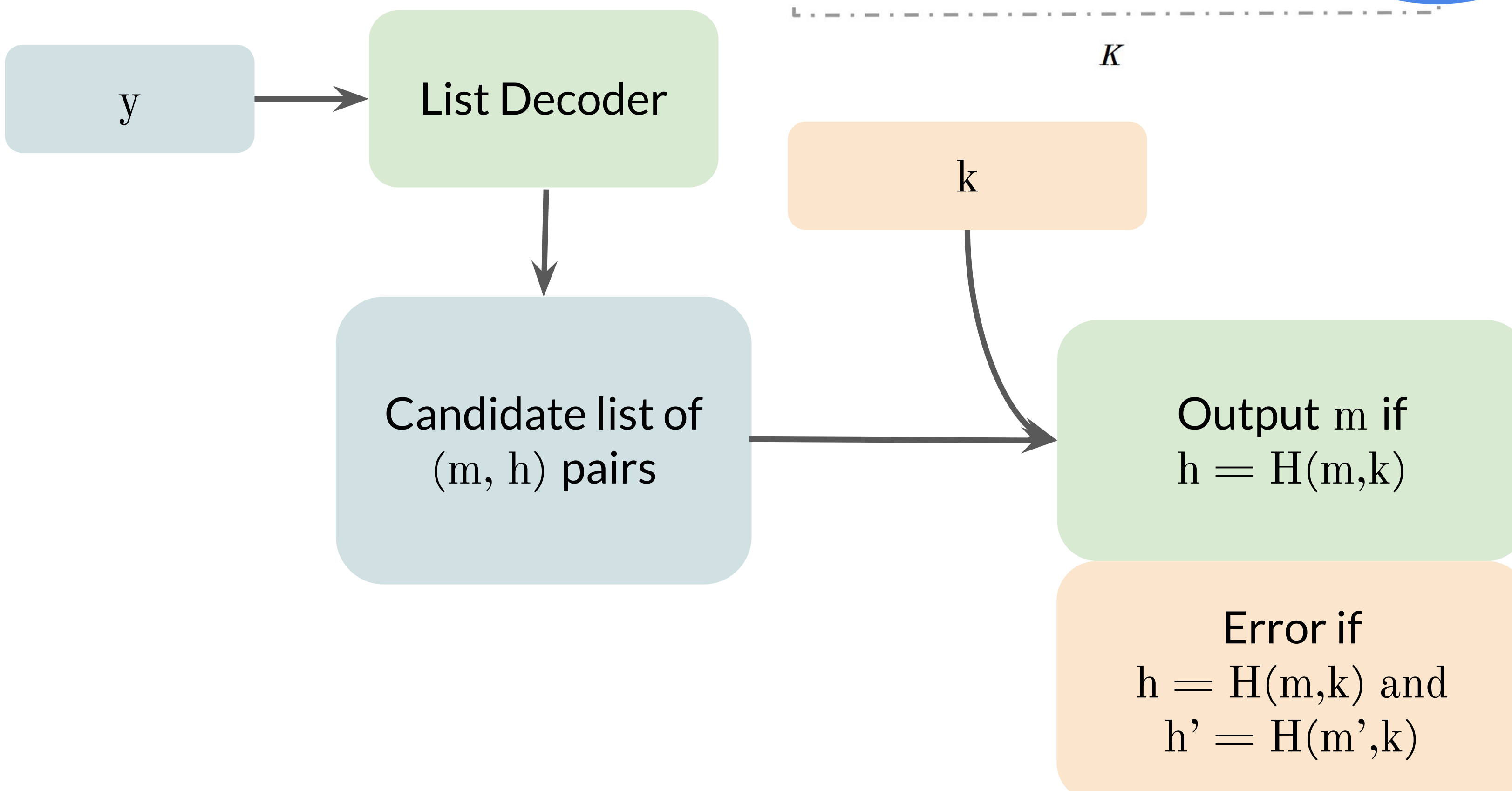
Bob’s decoder





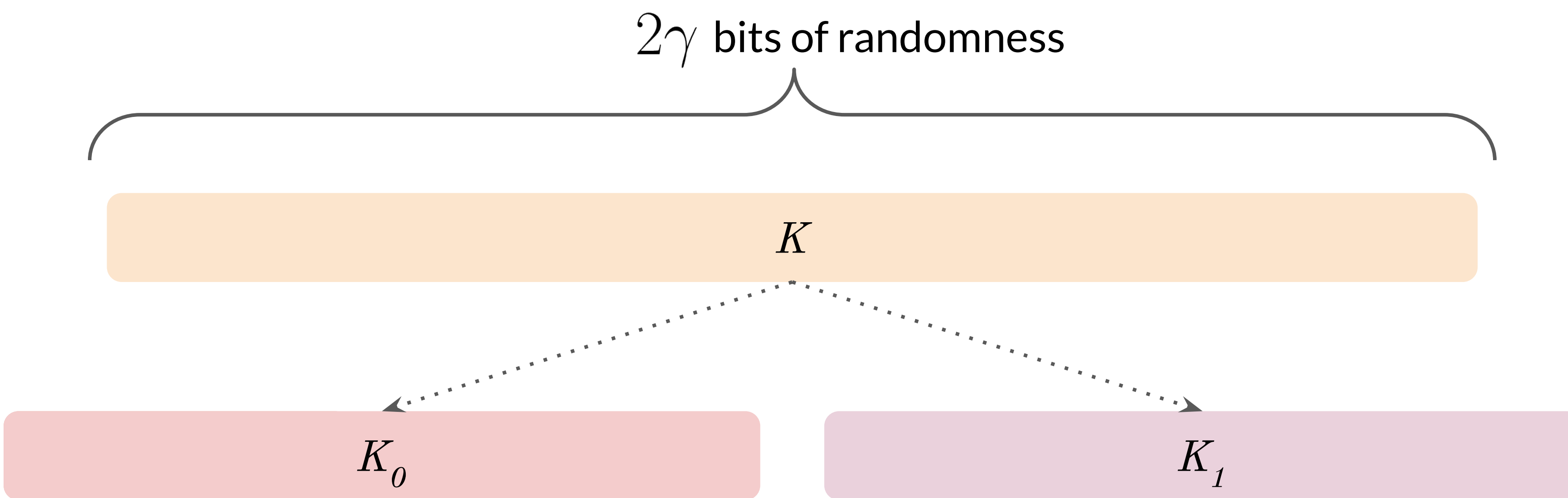
Bob's decoder





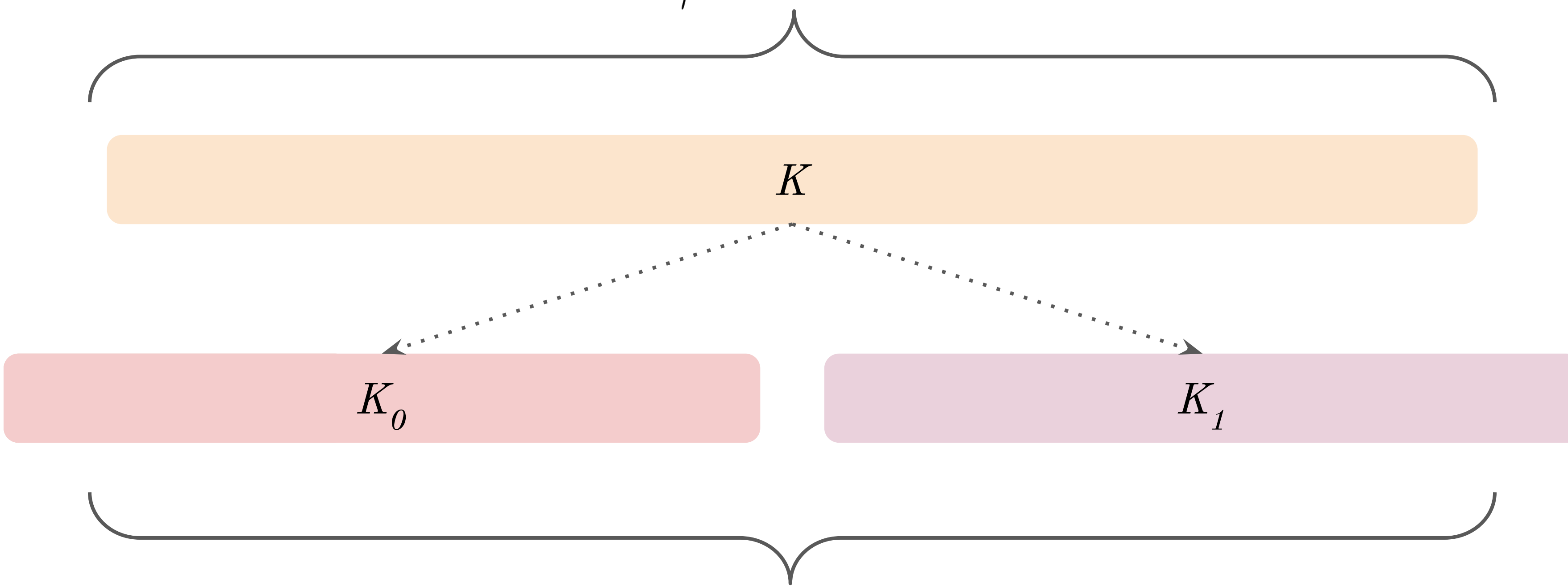
Randomness generation

$$\gamma := \log(n)$$



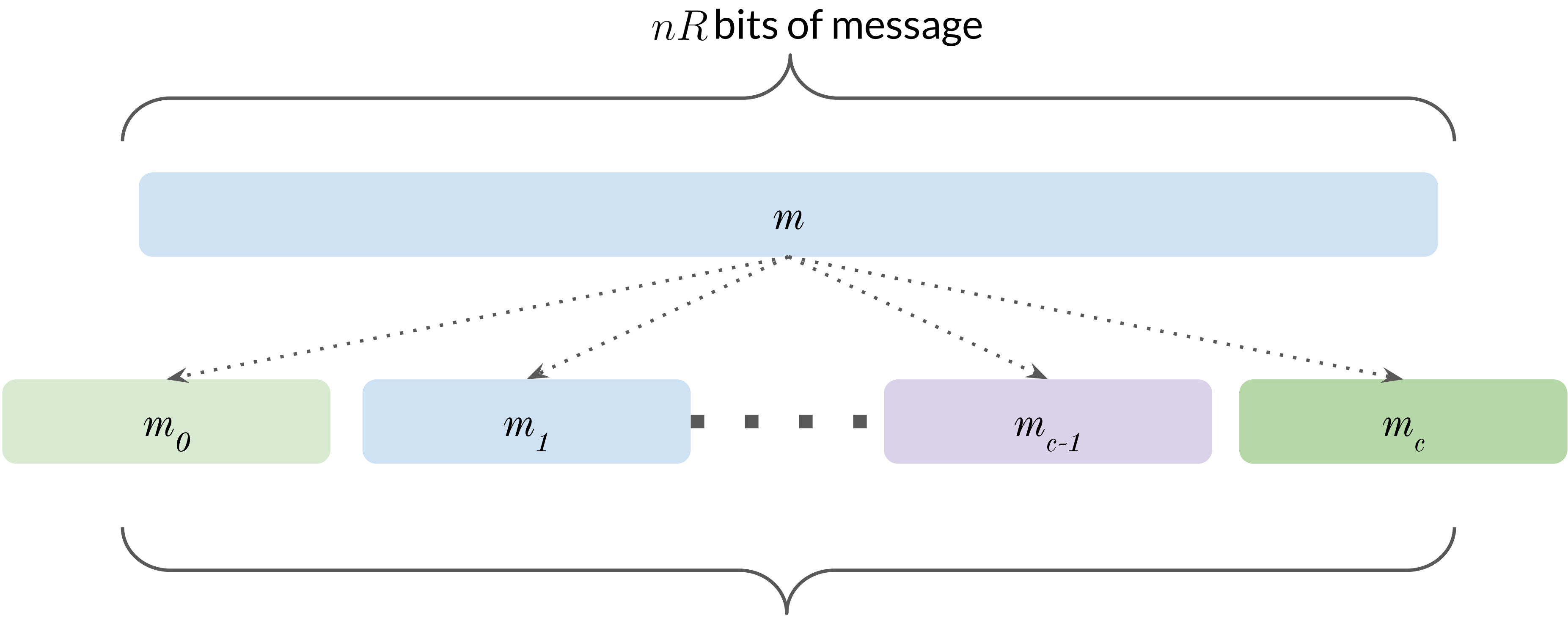
Randomness generation

2γ bits of randomness



Each of these belong to a field **F** of size **n**

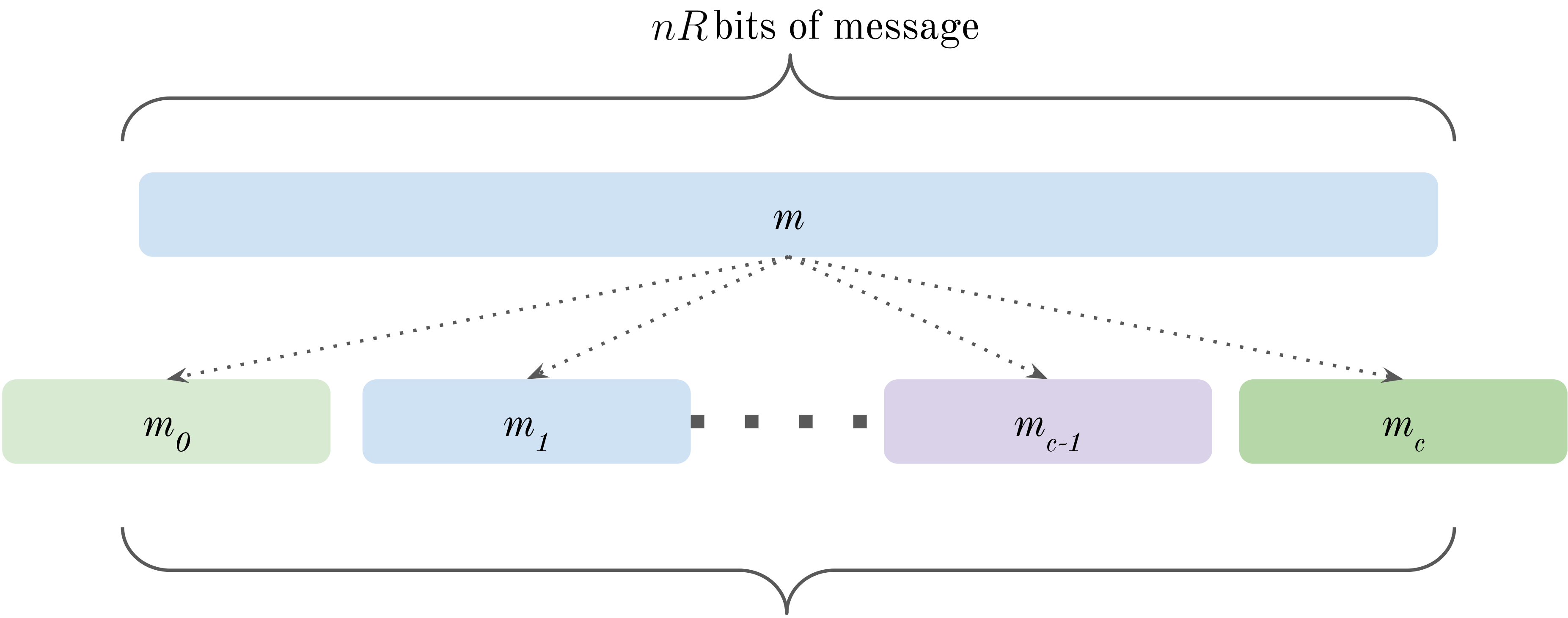
Alice’s encoder - hashing



Message broken up into $c = nR/\log(n)$ pieces.

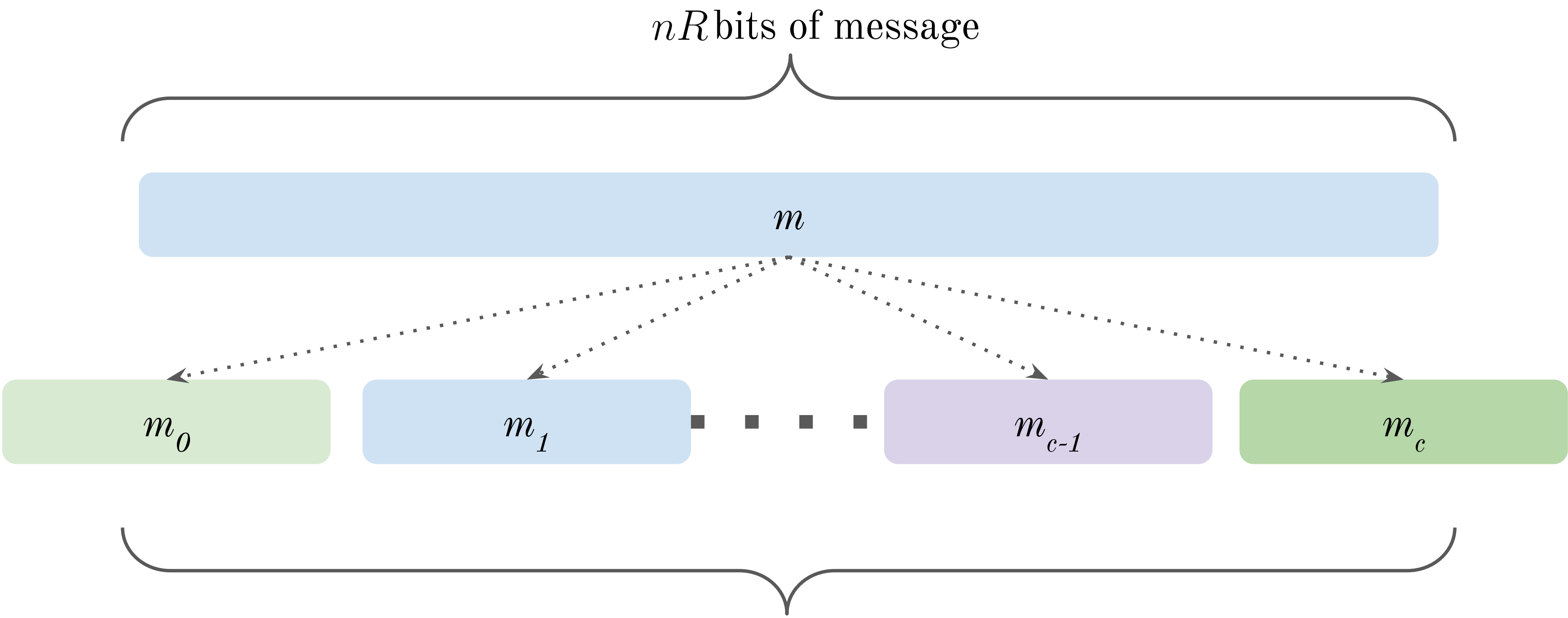
Each belongs to the field F

Alice's encoder - hashing



Arranged in a 1D array of length *c*

Alice's encoder - hashing



Arranged in a 1D array of length *c*

$$\text{Hash} = K_0 + \sum_{i=1}^{c} K_1^i m_i$$

Analysis

We make James **stronger**

- by **revealing** the hash and
- allowing him to send an **arbitrary short** list to Bob, as long as the correct pair is in the list.

If we show the rate is achievable, it will still be achievable with the weaker James.

Analysis

$$h = K_0 + \sum_{i=1}^{c} K_1^i m_i \qquad h' = K_0 + \sum_{i=1}^{c} K_1^i m_i'$$

$$h - h' = \sum_{i=1}^{c} K_1^i (m_i' - m_i)$$

Conditioned on everything that James knows, K_1 is uniformly distributed…

James has to guess **some** K_1 that will be consistent with the ***true*** message-hash pair.

Analysis

$$h - h' = \sum_{i=1}^{c} K_1^i (m_i' - m_i)$$

- Reduces to guessing an assignment of variables that makes a polynomial evaluate to zero.

Analysis

$$h - h' = \sum_{i=1}^{c} K_1^i (m_i' - m_i)$$

- Reduces to guessing an assignment of variables that makes a polynomial evaluate to zero.
- Probability is small by the **Schwartz-Zippel lemma**.

Analysis

$$h - h' = \sum_{i=1}^{c} K_1^i (m_i' - m_i)$$

- Reduces to guessing an assignment of variables that makes a polynomial evaluate to zero.
- Probability is small by the **Schwartz-Zippel lemma**.
- It can be shown to give a polynomially decreasing error rate that goes down to zero - which proves the claim.

$$\text{Error Rate} = \frac{nR}{n\log(n)}$$

Average probability of error criterion

Holds even if adversary doesn’t know the message

Therefore a **strong** converse

The Converse

‘Achievability’ for the adversary

Two parts to the converse

1. Rate Converse

2. Randomness Converse

Two parts to the converse

1 Rate Converse

Like a standard DMC converse

2 Randomness Converse

Two parts to the converse

1 Rate Converse

Like a standard DMC converse

2 Randomness Converse

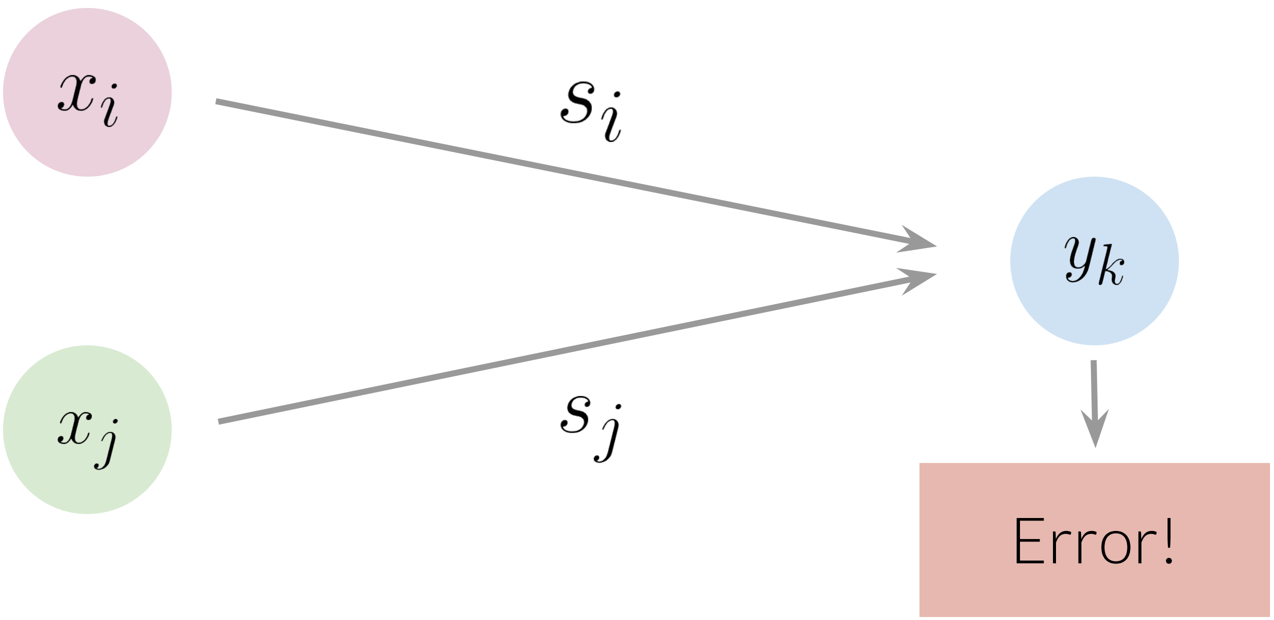
Same basic approach as in `[Langberg, 2004]` but extended

Proof

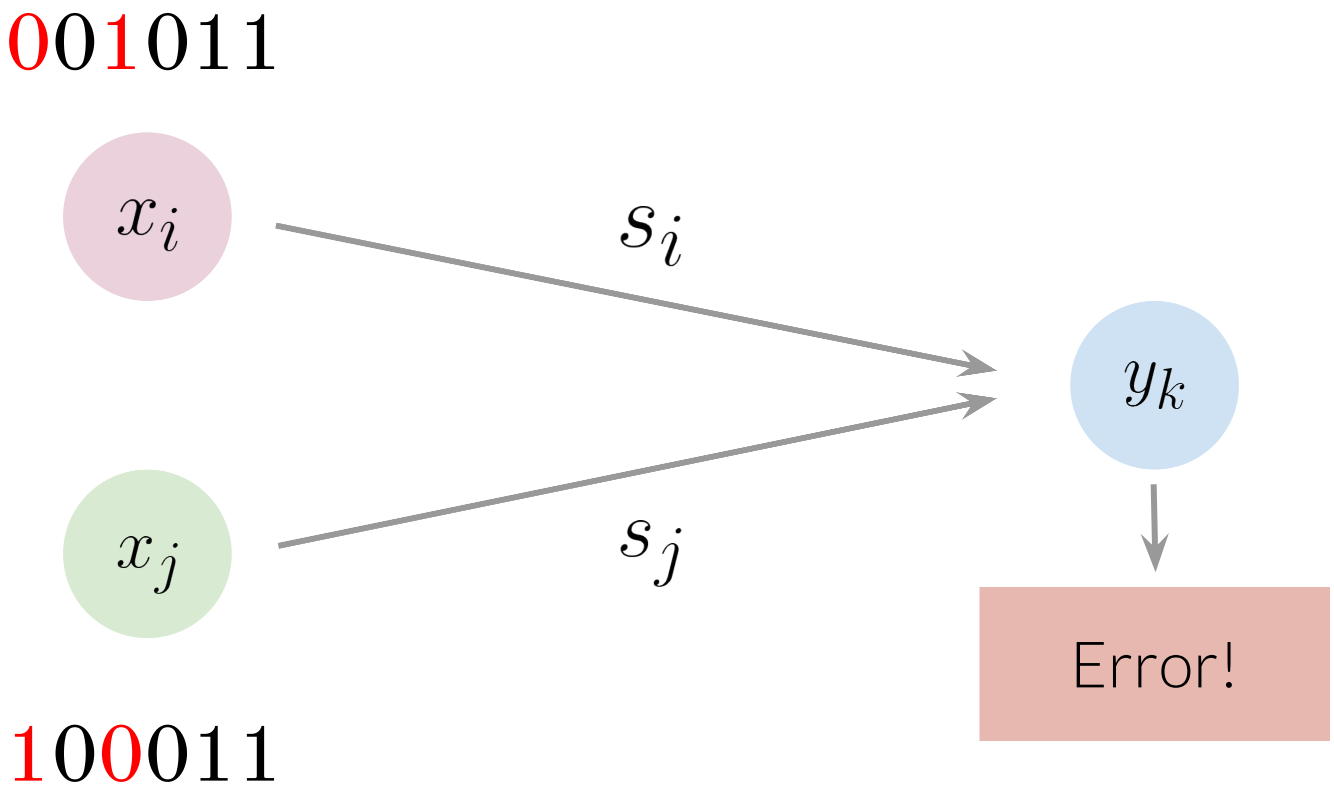
Available Randomness

$$\kappa := (1-\epsilon)\log(nR) - 1 < \log(n)$$

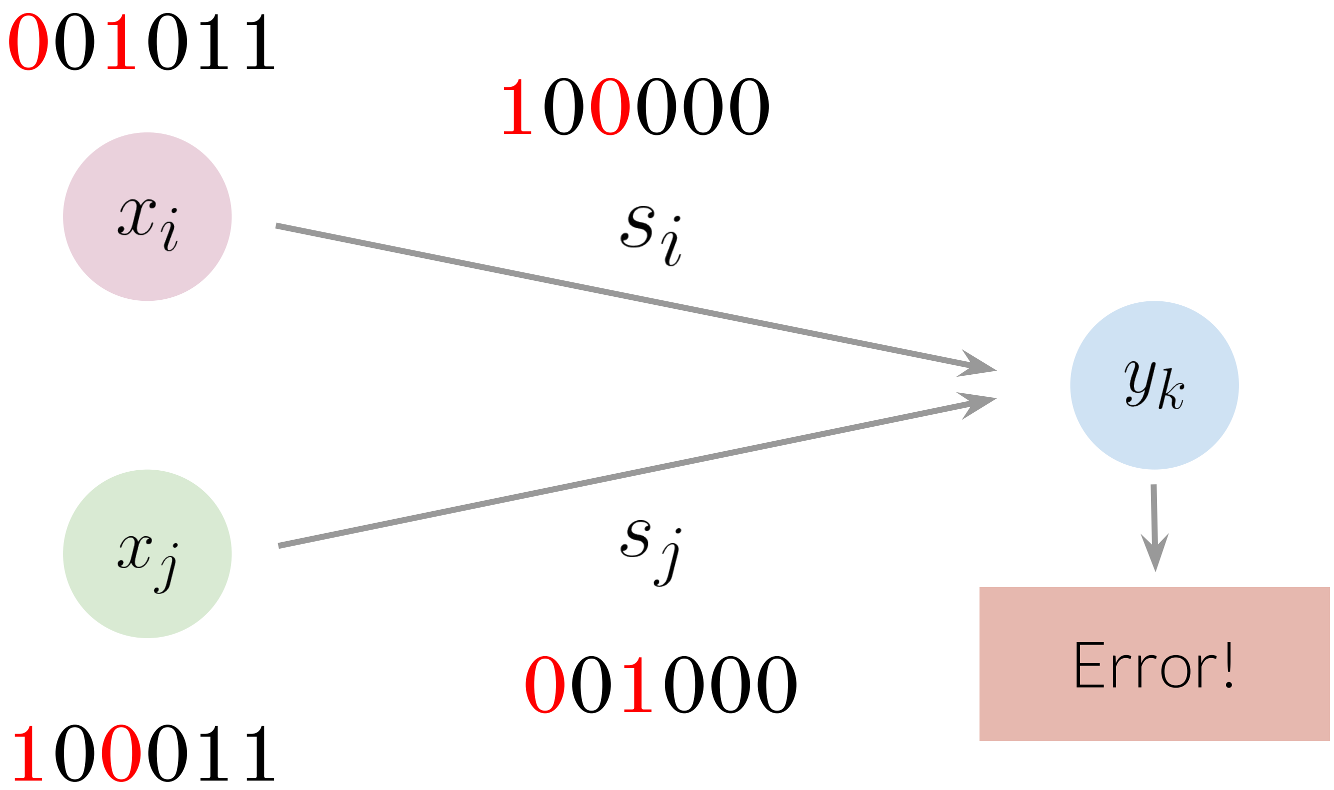
Confusability



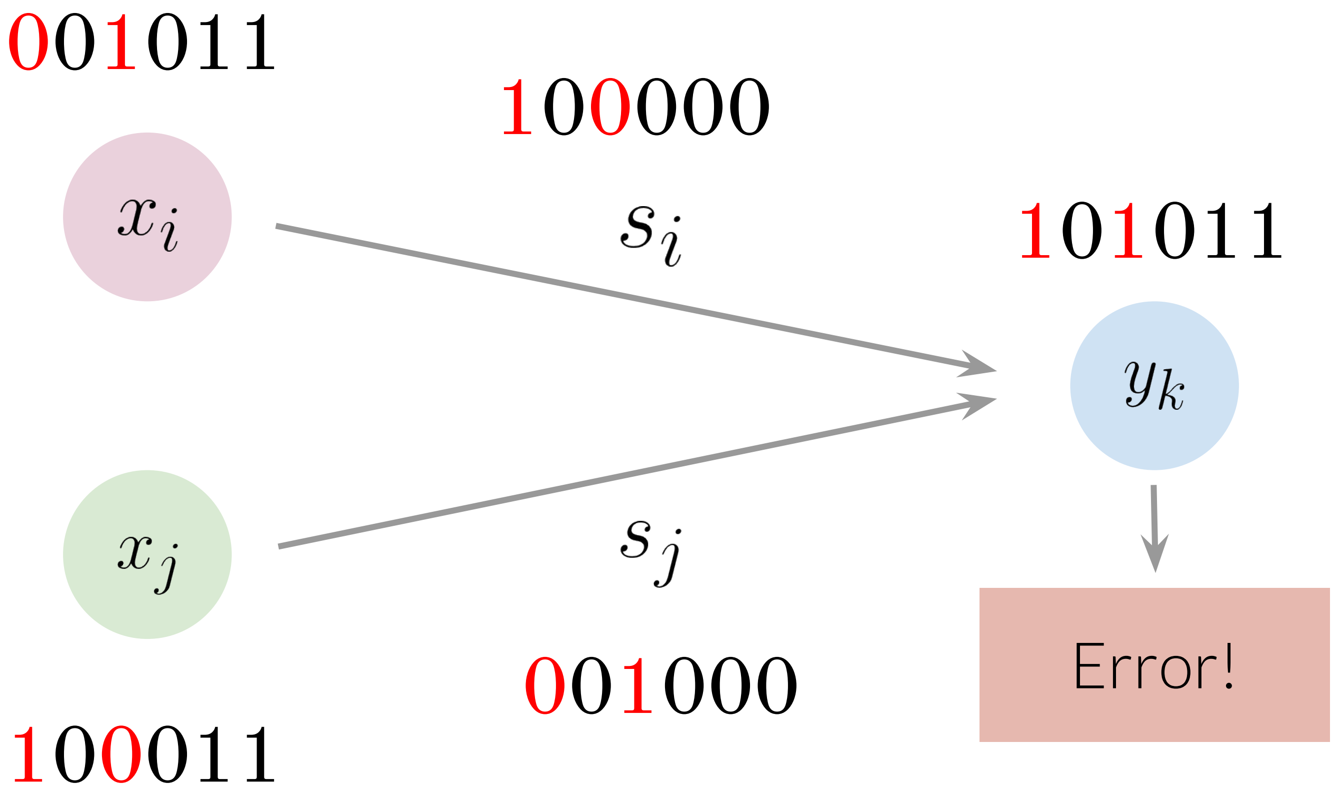
Confusability - example (Langberg)



Confusability - example (Langberg)



Confusability - example (Langberg)



Small and Large sets

$$\mathcal{U}_A := \{\mathbf{x} \in \mathcal{X}^n : \psi(m, k) = \mathbf{x}, k \in A\}$$

Small and Large sets

$$\mathcal{U}_A := \{\mathbf{x} \in \mathcal{X}^n : \psi(m, k) = \mathbf{x}, k \in A\}$$

Given *A*, these are called large if

$$|\mathcal{U}_A| \geq 2^{nR-(nR)^{1-\epsilon}}$$

The union of small sets is small

The union of small sets is $\underbrace{\text{small}}_{\leq 2^{nR-\frac{(nR)^{1-\epsilon}}{2}}}$

The union of small sets is small

$$\leq 2^{nR - \frac{(nR)^{1-\epsilon}}{2}}$$

because...

$$\left|2^{\mathcal{V}(K)}\right| = 2^{2^{\kappa}} = 2^{(nR)^{1-\epsilon}/2}$$

The union of small sets is $\underbrace{\text{small}}_{\leq 2^{nR - \frac{(nR)^{1-\epsilon}}{2}}}$

because...

$$\left|2^{\mathcal{V}(K)}\right| = 2^{2^{\kappa}} = 2^{(nR)^{1-\epsilon}/2}$$

$$\mathcal{V}^{(K)} := \{k \in [2^{\kappa}] : \psi(m,k) \in \mathcal{U}\}$$

The union of small sets is small

$$\leq 2^{nR-\frac{(nR)^{1-\epsilon}}{2}}$$

because...

$$\left|2^{\mathcal{V}(K)}\right| = 2^{2^{\kappa}} = 2^{(nR)^{1-\epsilon}/2}$$

The union of small sets is small

because...

$$\left|2^{\mathcal{V}(K)}\right| = 2^{2^{\kappa}} = 2^{(nR)^{1-\epsilon}/2}$$

$$\leq 2^{nR - \frac{(nR)^{1-\epsilon}}{2}}$$

And small sets have size

$$|\mathcal{U}_A| \leq 2^{nR-(nR)^{1-\epsilon}}$$

The union of small sets is small

$$\leq 2^{nR - \frac{(nR)^{1-\epsilon}}{2}}$$

because...

$$\left|2^{\mathcal{V}(K)}\right| = 2^{2^{\kappa}} = 2^{(nR)^{1-\epsilon}/2}$$

And small sets have size

$$|\mathcal{U}_A| \leq 2^{nR-(nR)^{1-\epsilon}}$$

Which gives the result!

Number of *(m,k)* pairs that map to codewords in small sets is small

$$\underbrace{\leq 2^{nR - \frac{(nR)^{1-\epsilon}}{2}}}_{\text{small}} \quad \times \quad \underbrace{2^{\kappa}}_{\text{small}} \quad = \quad \text{SMALL}$$

Number of codewords in 'small' sets ✖ Total number of *k* vectors = SMALL

small

small

Pairs that map to codewords in **large** sets is large

Large codes with rates higher than C_d have at least $|C|/4$ confusable pairs

Otherwise, expurgation of the small number of confusable codewords gives a deterministic code with higher rate than C_d

Quick aside - a non-adversary weakened AVC

$$\mathcal{X} = \mathcal{S} = \{0, 1\}$$
$$\mathcal{Y} = \{0, 1, 2\}$$

y = x if s = 0

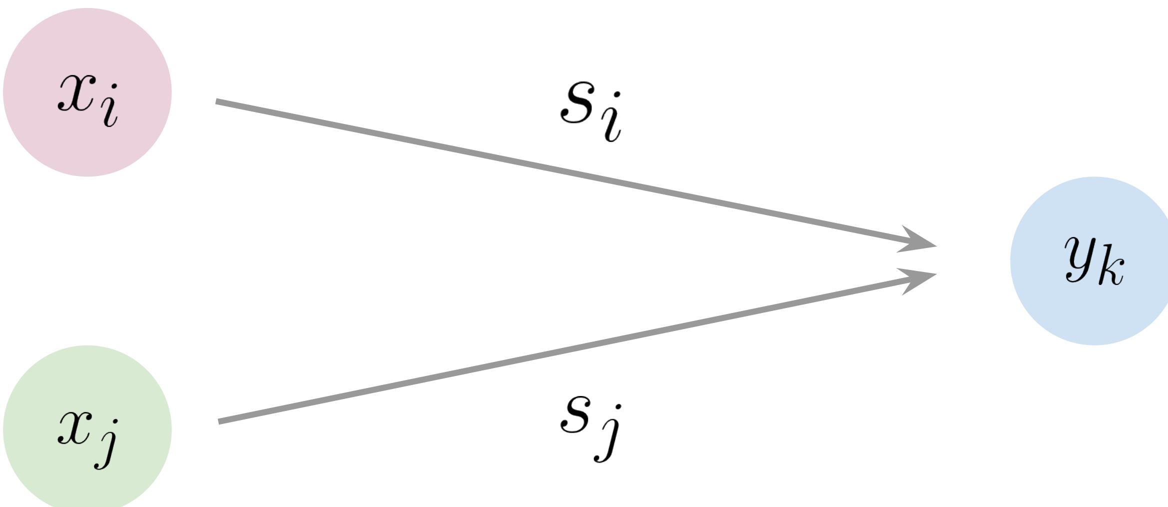
y = x + s if s = 1

The $\mathcal{U}_A$ have many confusable pairs

By the previous result

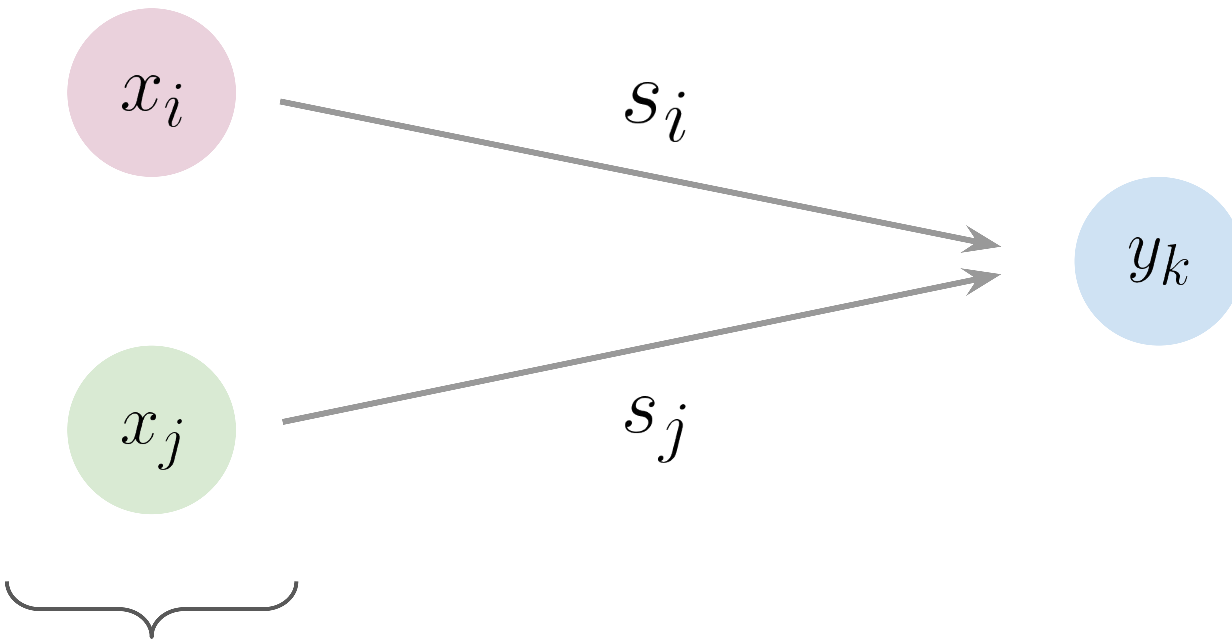
Jamming strategy

1. Identify the large $\mathcal{U}_A$'s



Jamming strategy

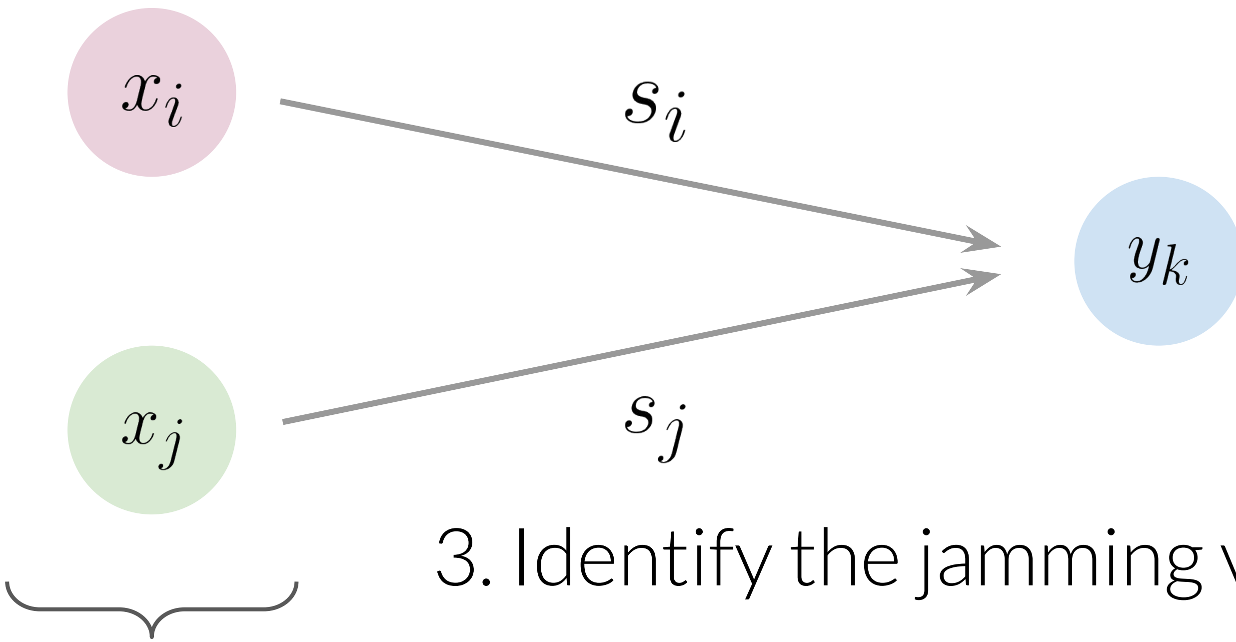
1. Identify the large $\mathcal{U}_A$'s



2. Identify the confusable pairs

Jamming strategy

1. Identify the large $\mathcal{U}_A$'s

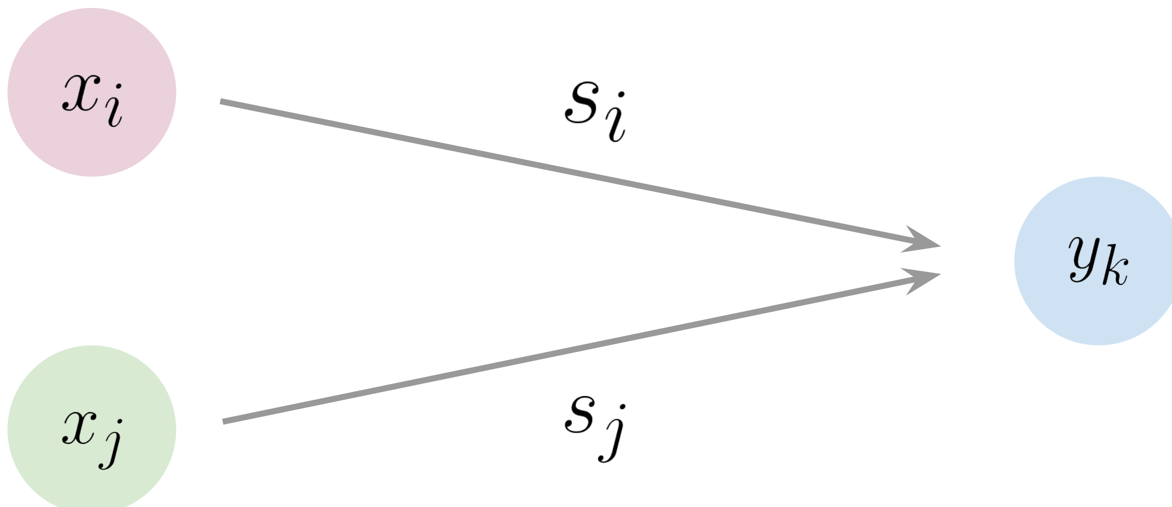


3. Identify the jamming vectors

2. Identify the confusable pairs

Jamming strategy

1. Identify the large $\mathcal{U}_A$'s



3. Identify the jamming vectors

2. Identify the confusable pairs

Error!

Many Errors

Since there are many codewords in large sets, and many of them are confusable, probability of error is **large**

QED

Take-away

$$(1 \pm \epsilon) \log(n)$$

bits are necessary and sufficient to achieve randomized coding capacity.

Precise threshold!

Thank you!

References

1. A. Lapidoth and P. Narayan, “Reliable communication under channel uncertainty,” October 1998.
2. R. Ahlswede, “Elimination of correlation in random codes for arbitrarily varying channels,” 1978.
3. D. Blackwell, L. Breiman, and A. J. Thomasian, “The capacity of a class of channels,” 1959.
4. R. Ahlswede, “Arbitrarily varying channels with states sequence known to the sender,”1986.
5. M. Langberg, “Private codes or succinct random codes that are (almost) perfect,” 2004.
6. A. Sarwate, “Robust and adaptive communication under uncertain interference,” Ph.D. dissertation, University of California, Berkeley, 2008.
7. J. T. Schwartz, “Fast probabilistic algorithms for verification of polynomial identities,” 1980.
8. R. Zippel, “Probabilistic algorithms for sparse polynomials,” 1979.