k-SUM in the sparse regime

Shweta Agrawal, Sagnik Saha, Nikolaj Ignatieff Schwartzbach, Akhil Vanukuri, Prashant Nalini Vasudevan

k-SUM in the sparse regime

The (average-case) k-SUM problem

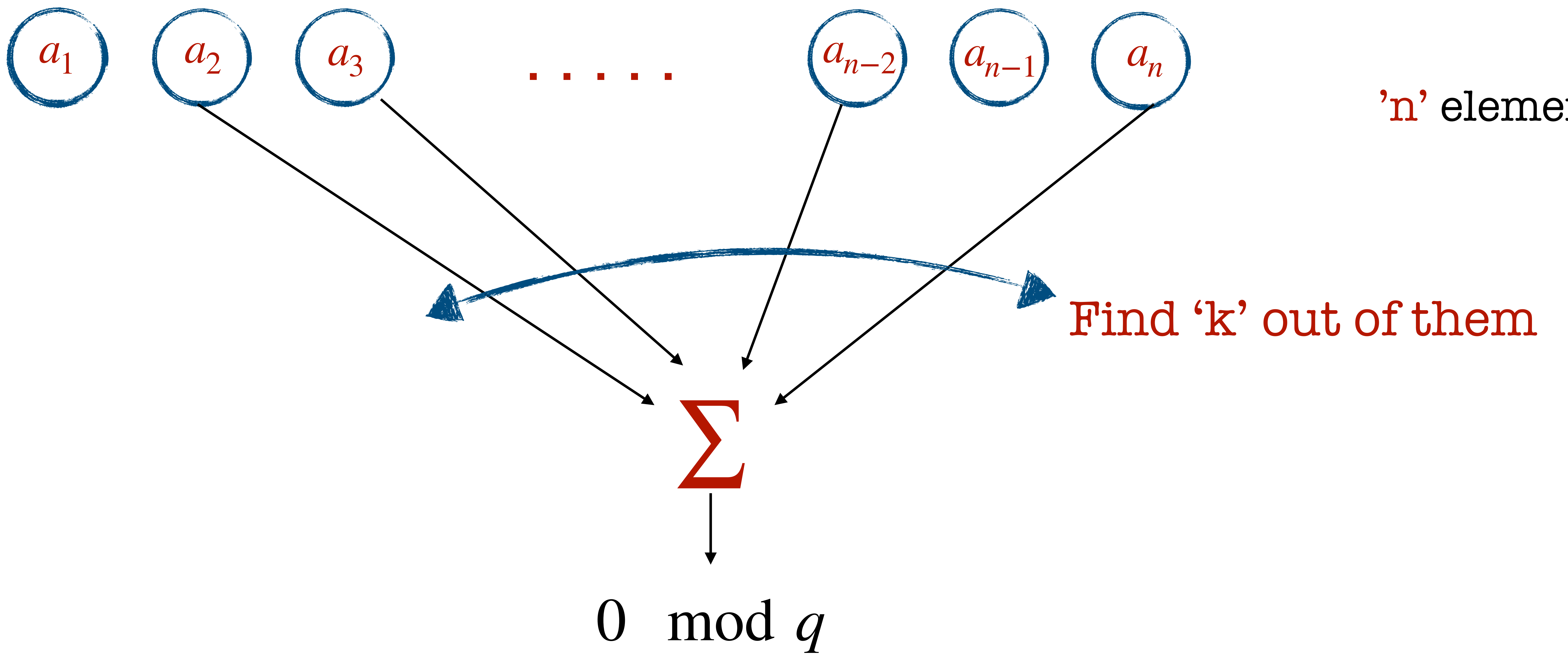
The (average-case) k-SUM problem

a_1 a_2 a_3 . . . . . a_{n-2} a_{n-1} a_n

$a_i \xleftarrow{\$} \mathbb{Z}_q$

'n' elements sampled uniformly from $\mathbb{Z}_q$

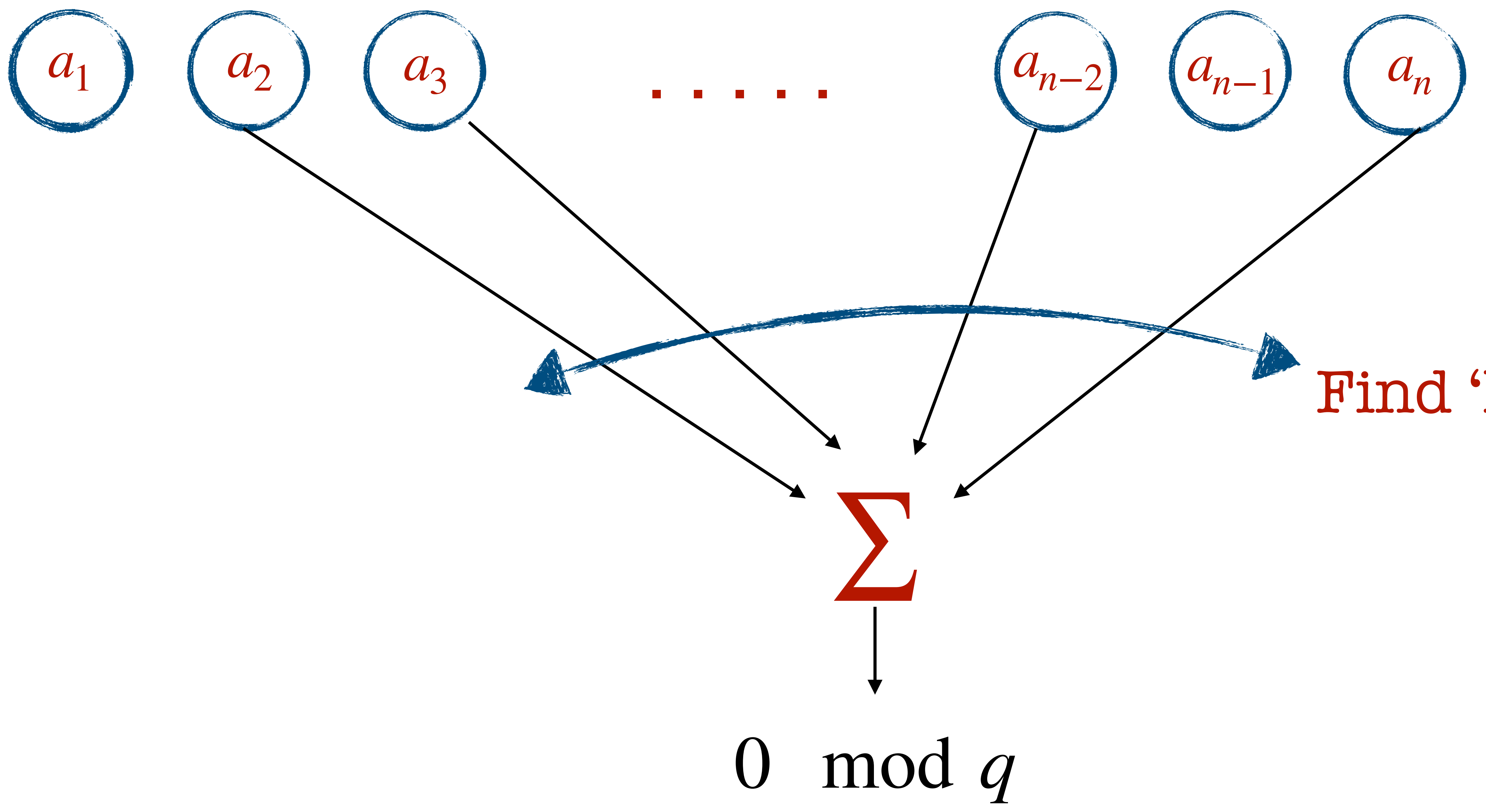
The (average-case) k-SUM problem



$a_i \xleftarrow{\$} \mathbb{Z}_q$

'n' elements sampled uniformly from $\mathbb{Z}_q$

The (average-case) k-SUM problem



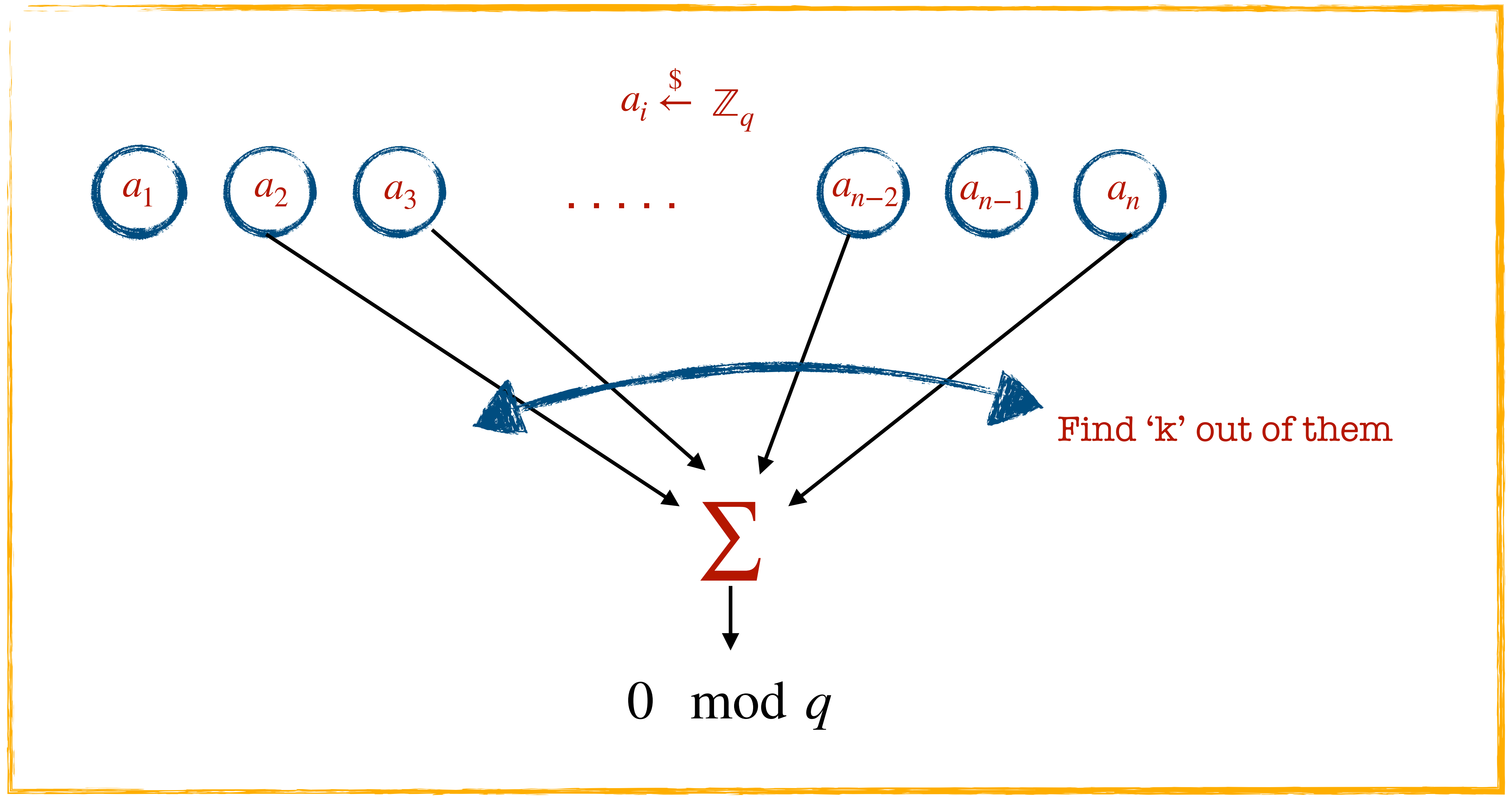
Find 'k' out of them

$a_i \xleftarrow{\$} \mathbb{Z}_q$

'n' elements sampled uniformly from $\mathbb{Z}_q$

k is 'small'

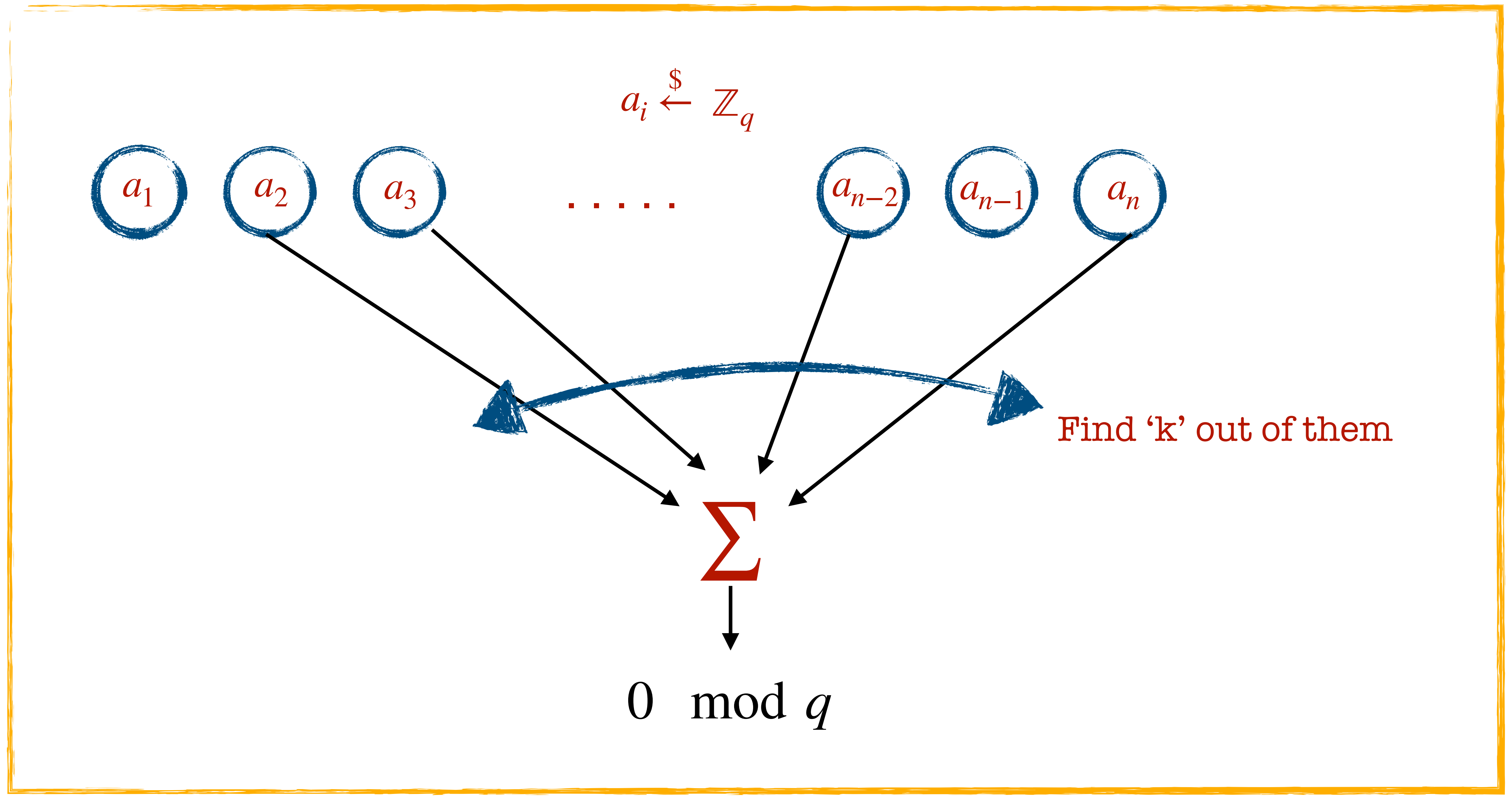
The (average-case) k-SUM problem



$$\sum_{i \in S} a_i = 0 \mod q \quad \text{where} \quad |S| = k$$

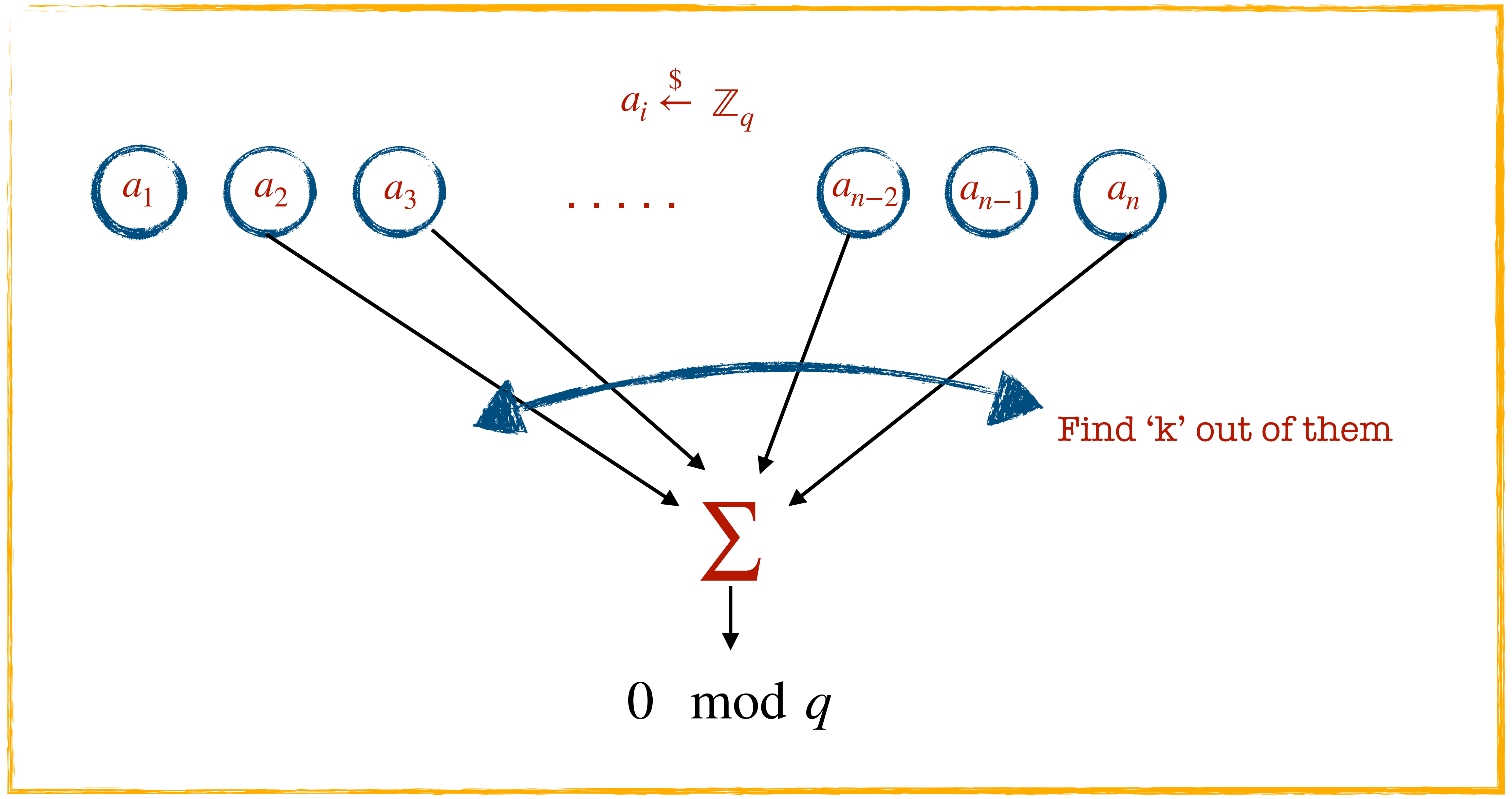
The (average-case) k-SUM problem

Expected no.of solutions = $\displaystyle\sum_{i=1}^{\binom{n}{k}} \frac{1}{q} = \frac{\binom{n}{k}}{q}$



$$\sum_{i \in S} a_i = 0 \mod q \quad \text{where} \quad |S| = k$$

The (average-case) k-SUM problem



$$\sum_{i \in S} a_i = 0 \mod q \quad \text{where} \quad |S| = k$$

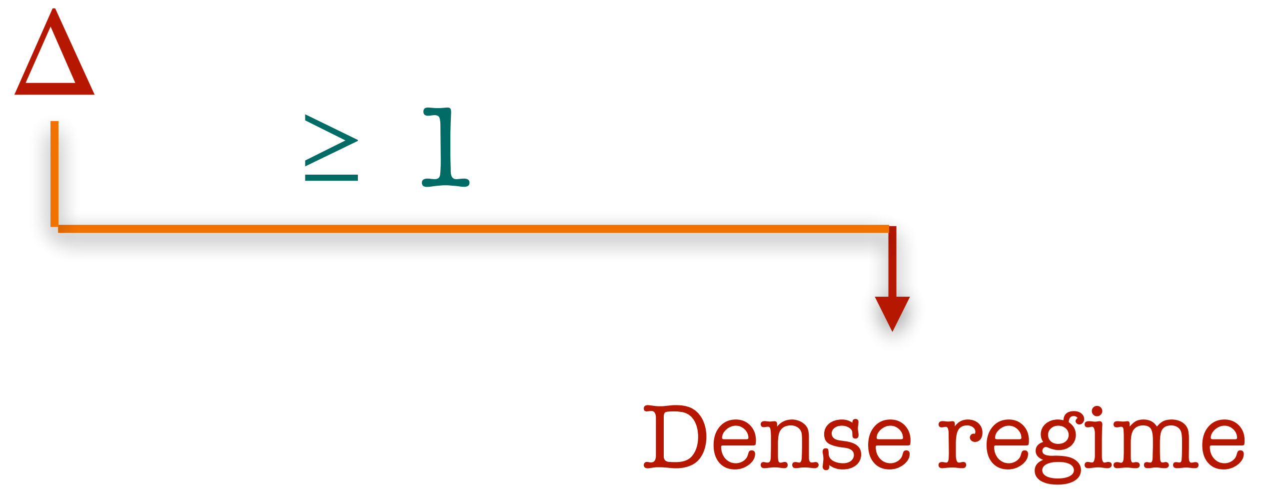
Expected no.of solutions = $\displaystyle\sum_{i=1}^{\binom{n}{k}} \frac{1}{q} = \frac{\binom{n}{k}}{q}$

Density, $\Delta = \dfrac{\log \binom{n}{k}}{logq} \approx \dfrac{k \log n}{\log q}$

The (average-case) k-SUM problem

Δ

The (average-case) k-SUM problem



The (average-case) k-SUM problem

Δ

≥ 1

Dense regime

(Well studied in literature)

[Wag02,Pet15, LLW19, BDV20, DKK21, etc..]

The (average-case) k-SUM problem

$\Delta \geq 1$

Dense regime

(Well studied in literature)

[Wag02,Pet15, LLW19, BDV20, DKK21, etc..]

- It has been central in studying the complexity of important problems in theoretical computer science [AW14, Pat10, GO95, BHP01, SEO03, KPP16].

The (average-case) k-SUM problem

Δ

< 1 | ≥ 1

Sparse regime

(Not much is known)

Dense regime

(Well studied in literature)

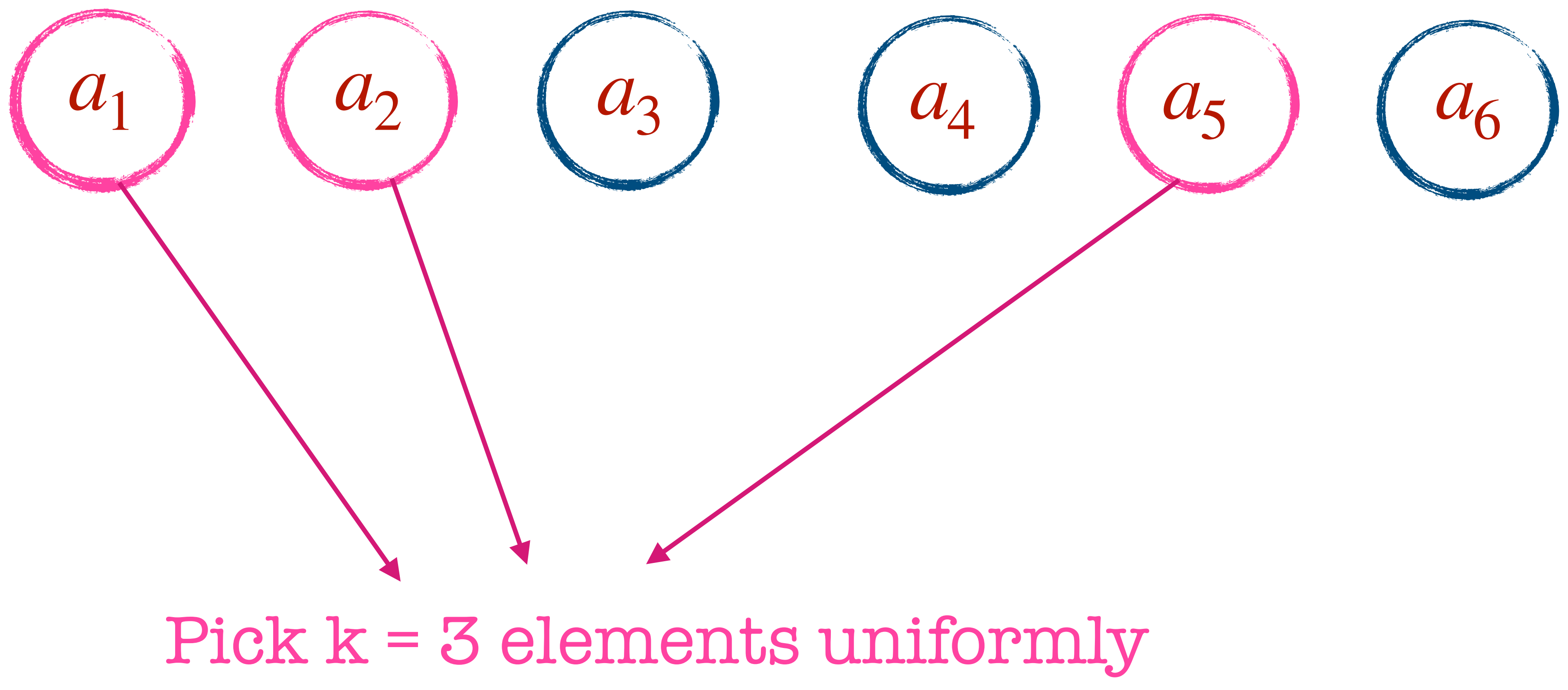
[Wag02,Pet15, LLW19, BDV20, DKK21, etc..]

- Planted k-SUM problem - example: $n = 6$, $k = 3$

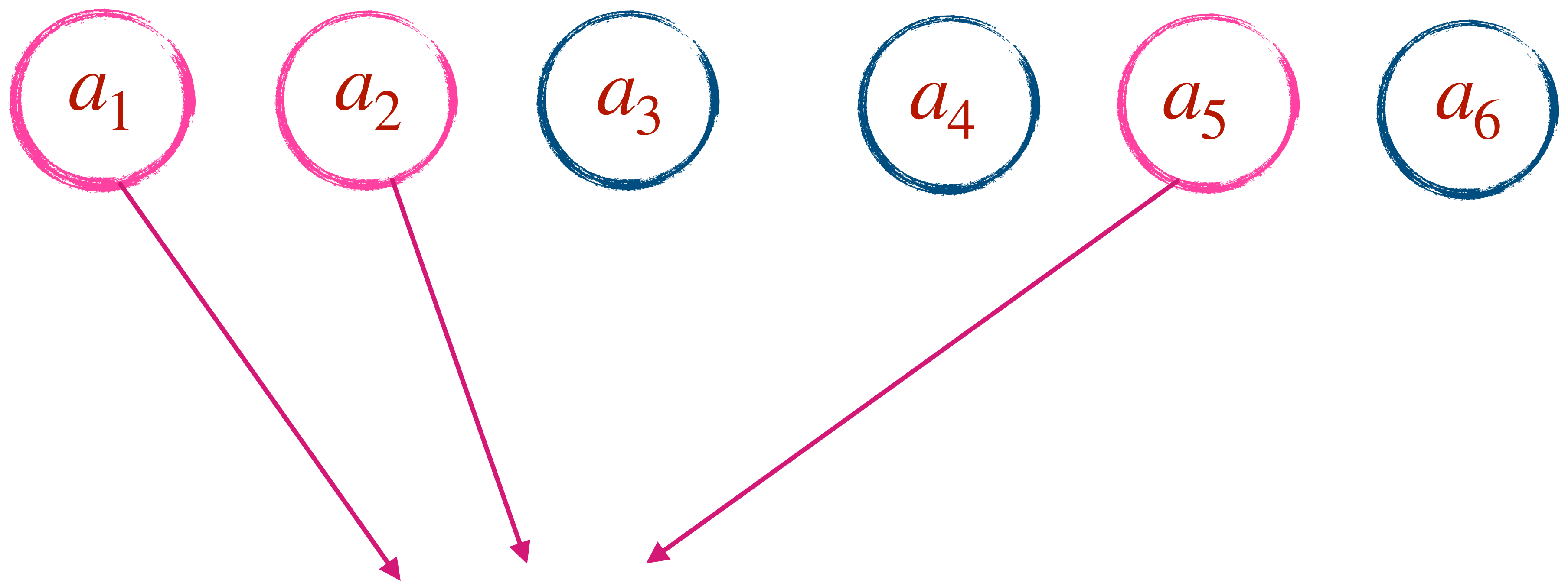
- Planted k-SUM problem - example: n = 6 , k = 3

a_1 a_2 a_3 a_4 a_5 a_6

- Planted k-SUM problem - example: n = 6 , k = 3



- Planted k-SUM problem - example: n = 6 , k = 3

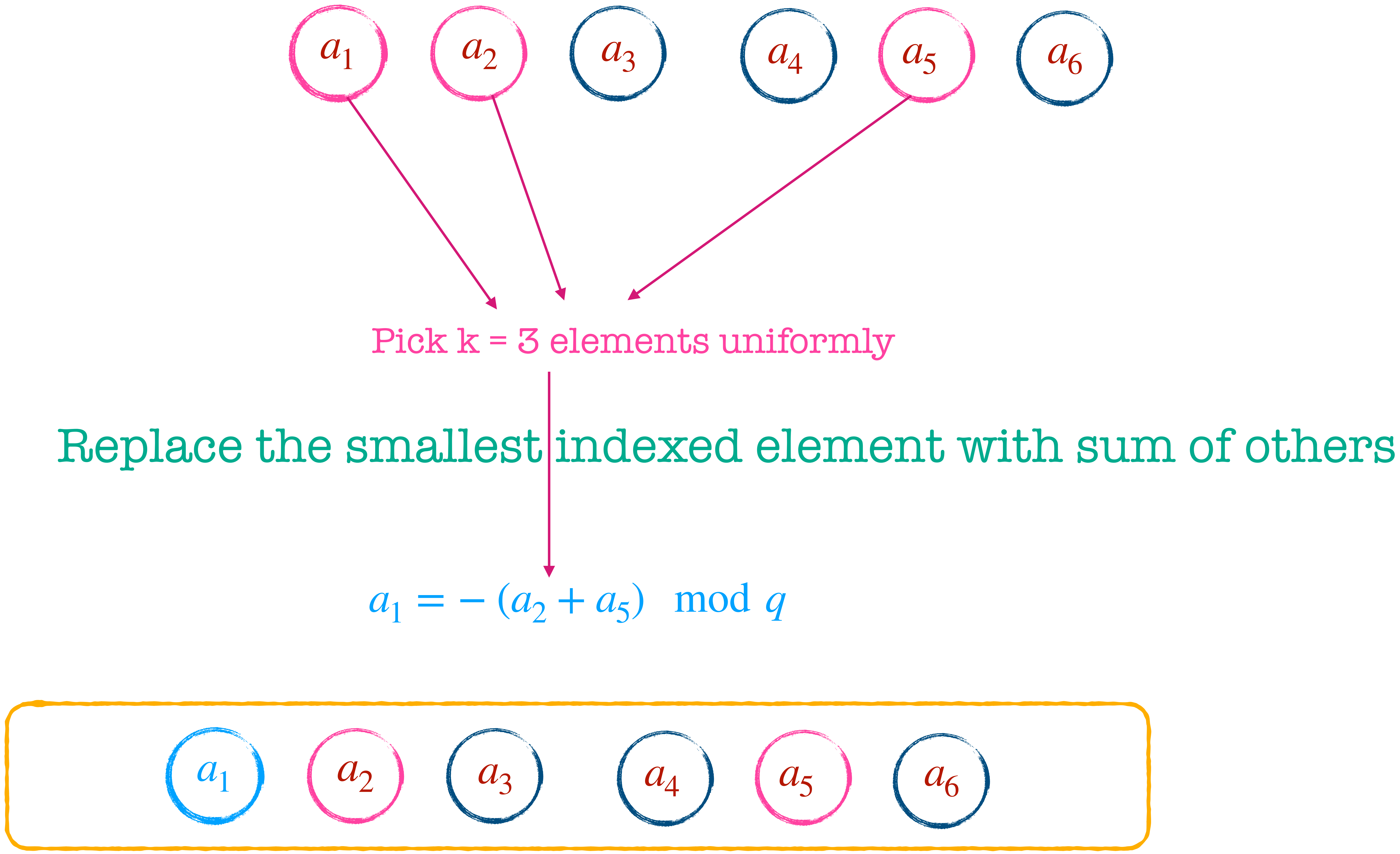


Pick k = 3 elements uniformly

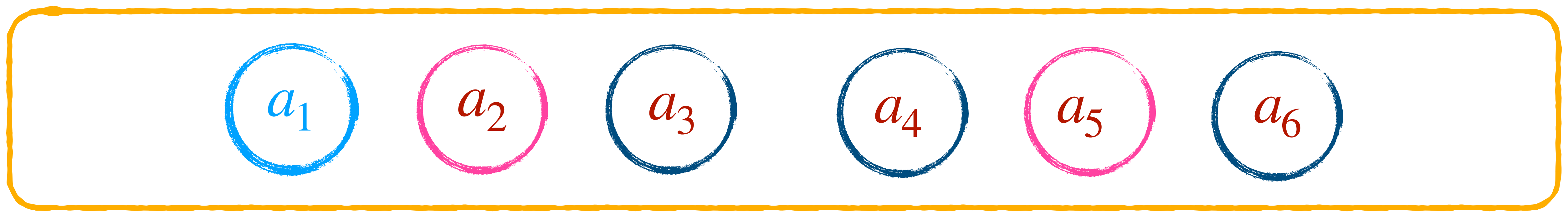
Replace the smallest indexed element with sum of others

$$a_1 = -(a_2 + a_5) \mod q$$

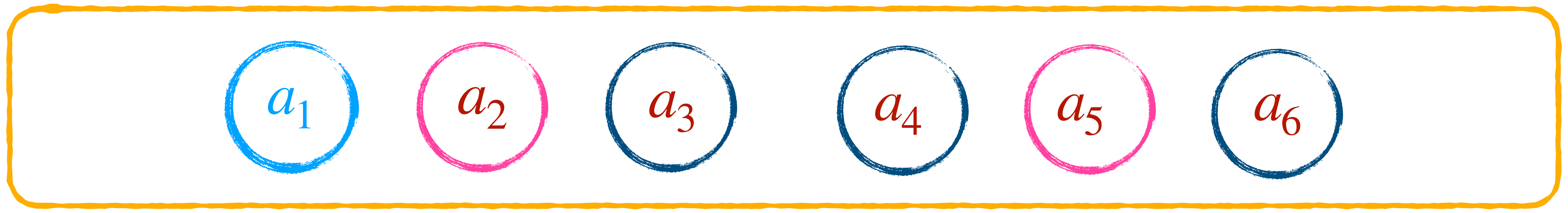
- Planted k-SUM problem - example: n = 6 , k = 3



Planted k-SUM instance



Planted k-SUM instance



Planted k-SUM instance

Search problem

Find the planted solution

a_1 a_2 a_3 a_4 a_5 a_6

Planted k-SUM instance

Search problem

Find the planted solution

Decision problem

Does the instance have a planting (or) not

Planted k-SUM

Complexity

Planted k-SUM

Complexity

Cryptography

Planted k-SUM

Complexity

Cryptography

- It is good to diversify the hardness assumptions used in cryptography

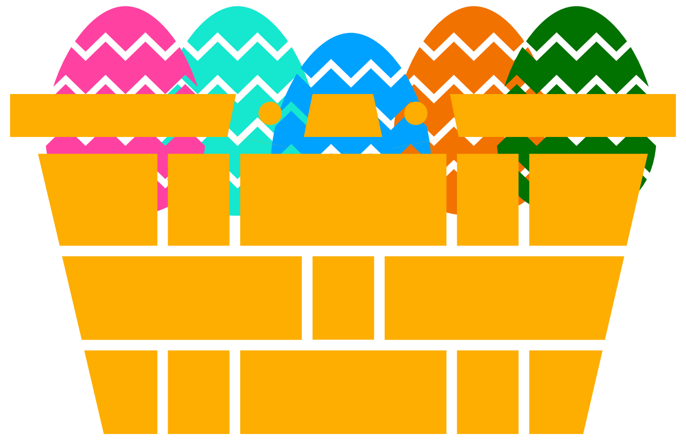
Planted k-SUM

Complexity

Cryptography

- It is good to diversify the hardness assumptions used in cryptography

You don't put all your eggs in the same basket!



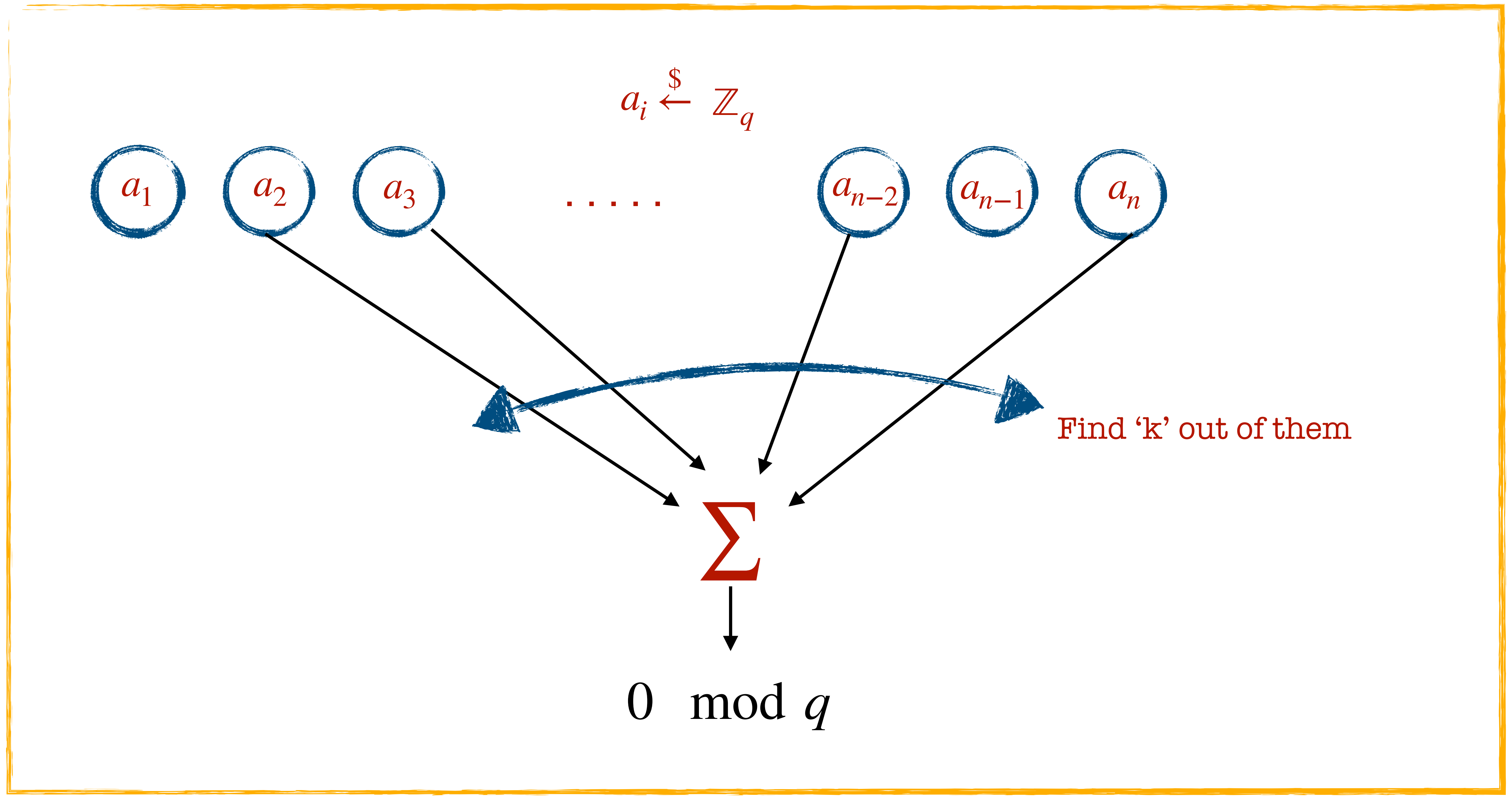
Planted k-SUM

Complexity

Cryptography

Algorithms

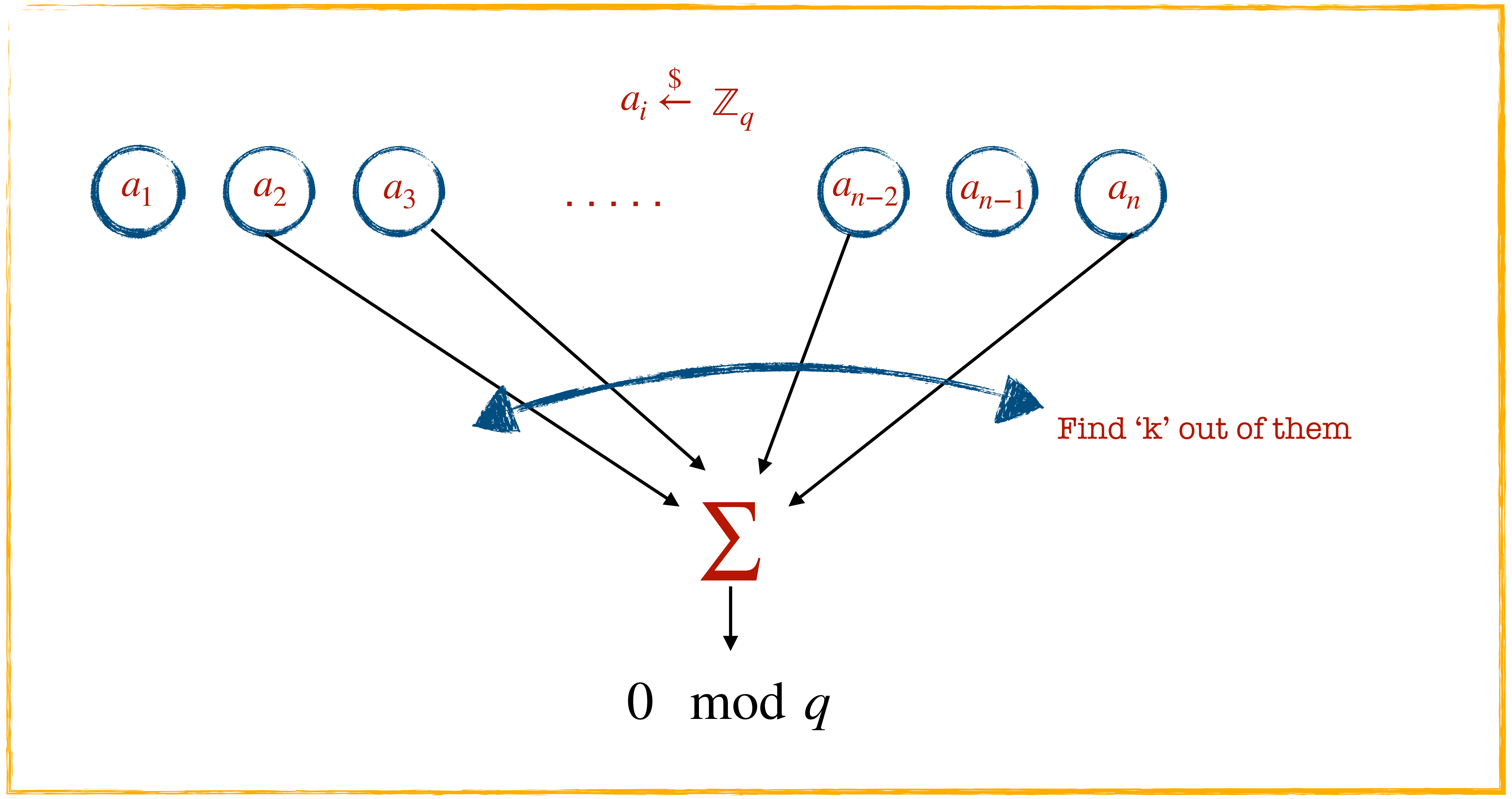
Planted k-SUM - conjecture



Brute force : $\binom{n}{k}$

Meet-in-the-middle : $n^{\lceil \frac{k}{2} \rceil}$

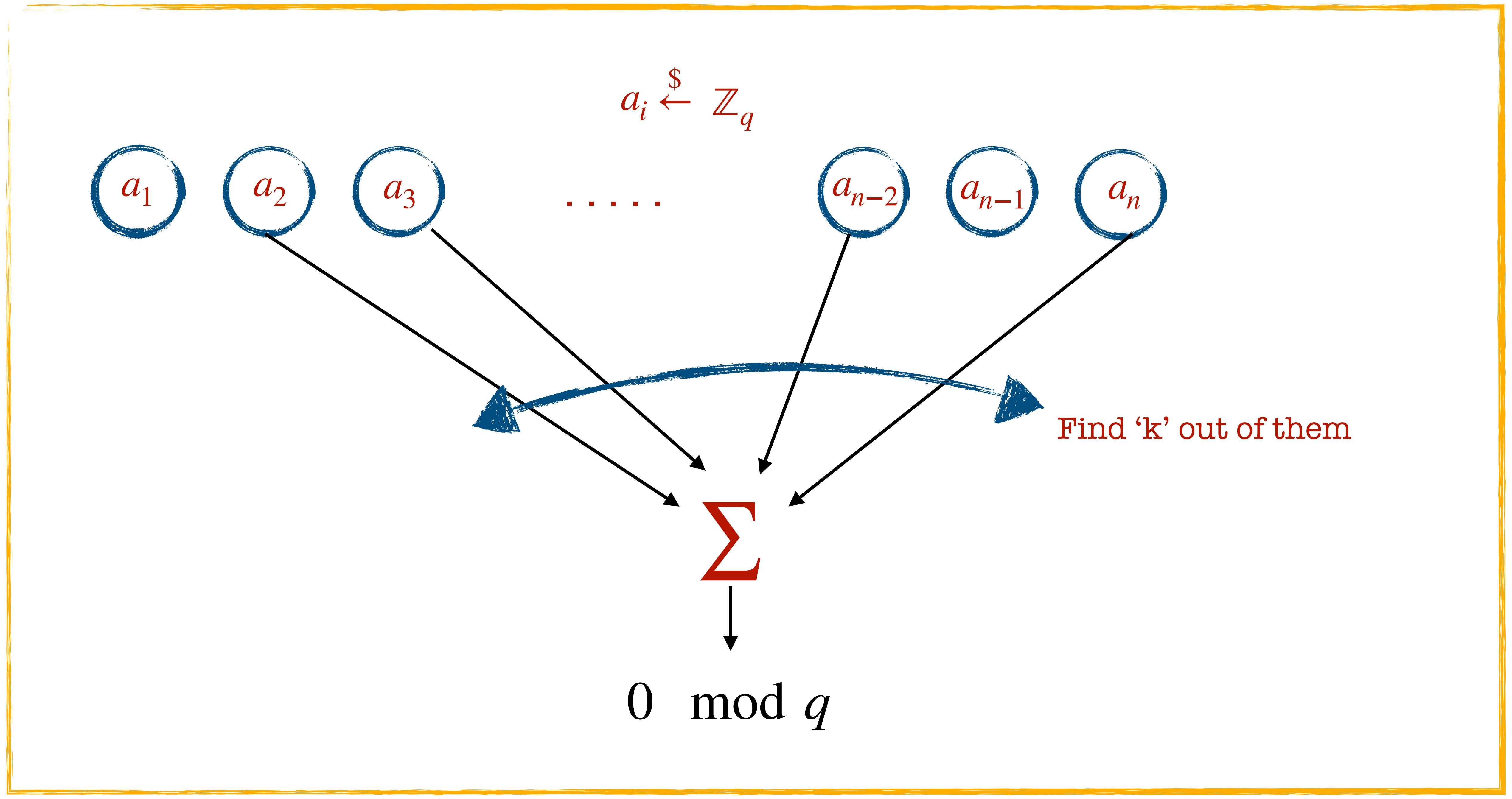
Planted k-SUM - conjecture



Brute force : $\binom{n}{k}$ Meet-in-the-middle : $n^{\lceil \frac{k}{2} \rceil}$

If k is a super constant i.e, $\omega(1)$ then this is super-poly time

Planted k-SUM - conjecture



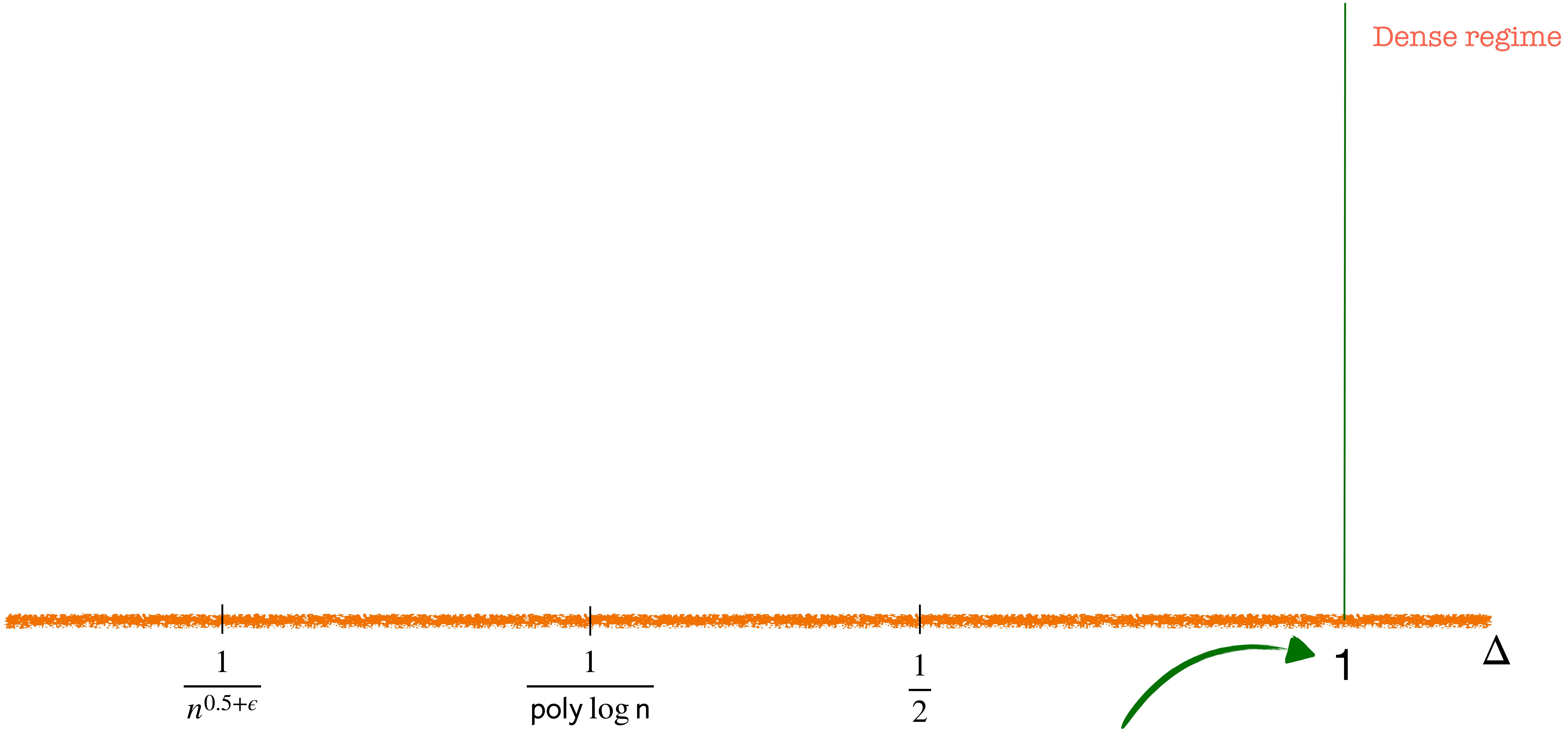
Brute force : $\binom{n}{k}$

Meet-in-the-middle : $n^{\lceil \frac{k}{2} \rceil}$

If k is a super constant i.e, $\omega(1)$ then this is super-poly time

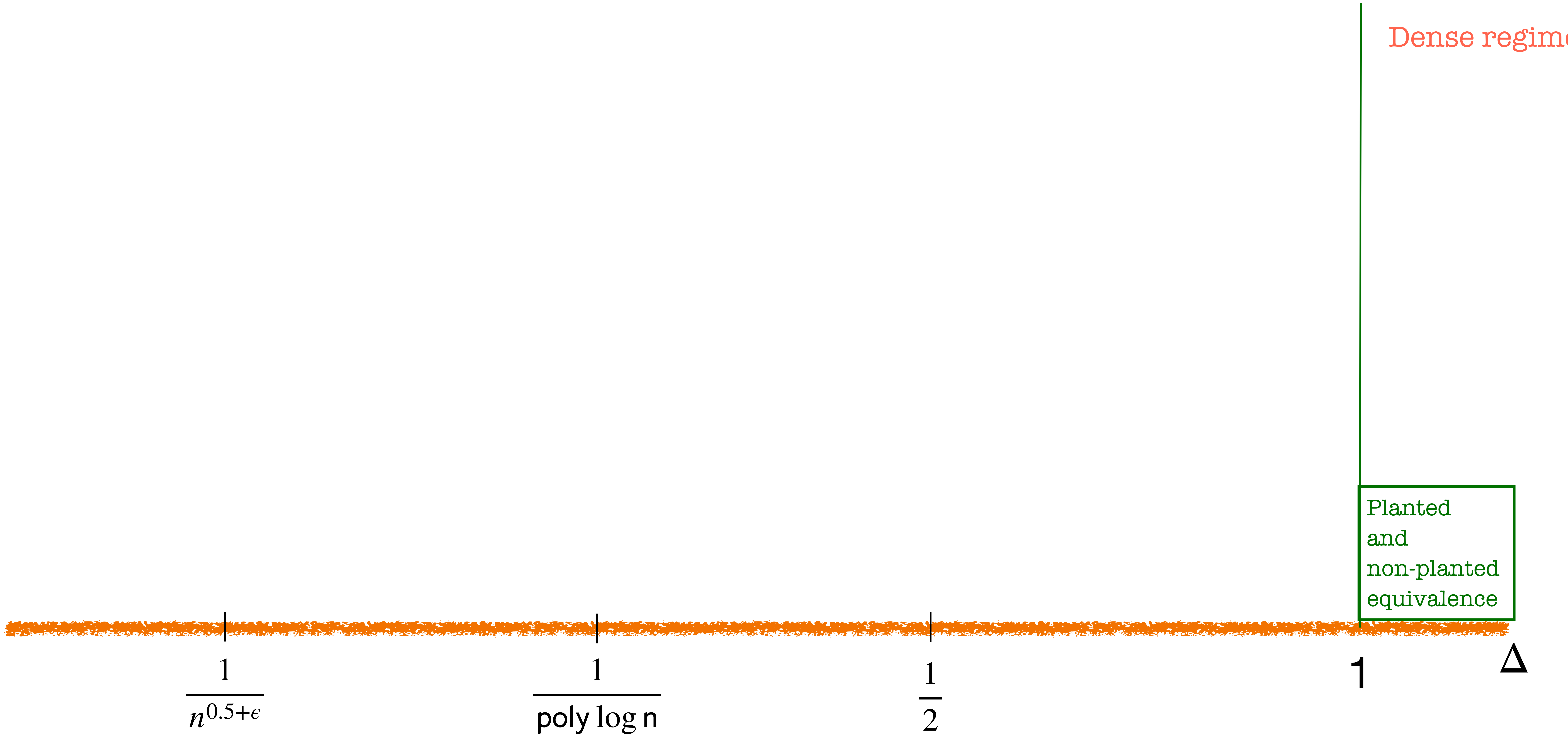
Conjecture : At $\Delta = 1$ best runtime of k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$ [Pet15, LLW19, DKK21].

Our Results



Conjecture : At $\Delta = 1$ best runtime of k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$ [Pet15, LLW19, DKK21].

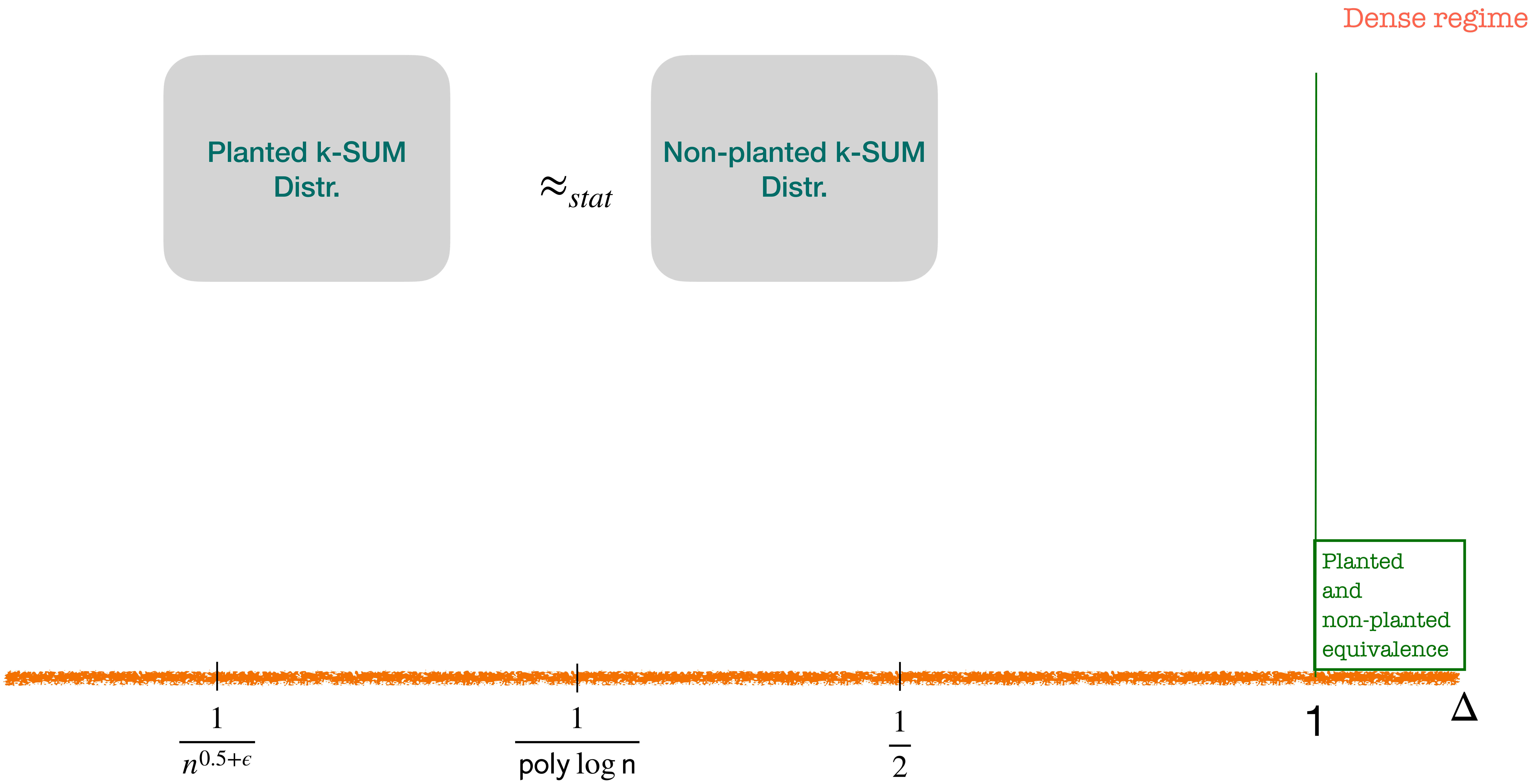
Our Results



Conjecture : At $\Delta = 1$ best runtime of k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$ [Pet15, LLW19, DKK21].

Our Results

Planted k-SUM Distr. $\approx_{stat}$ Non-planted k-SUM Distr.



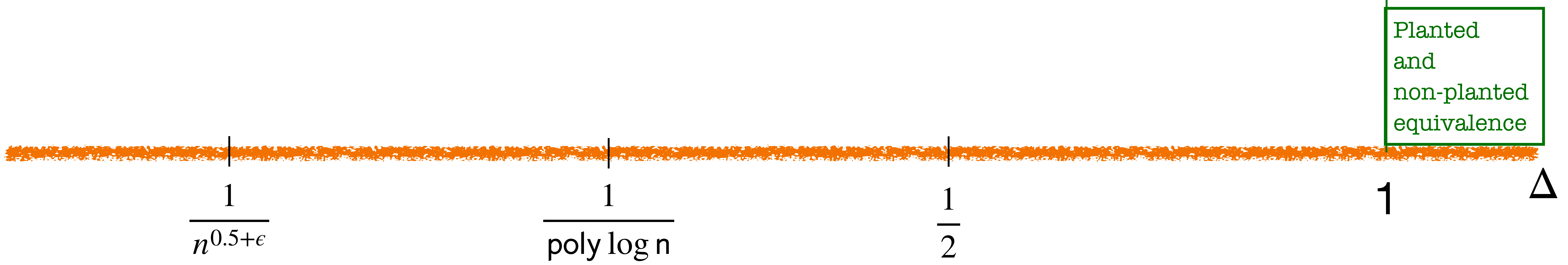
Conjecture : At $\Delta = 1$ best runtime of k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$ [Pet15, LLW19, DKK21].

Our Results

Dense regime

Planted k-SUM Distr. $\approx_{stat}$ Non-planted k-SUM Distr.

$\Longrightarrow$ Algorithm for planted-kSUM solves non-planted k-SUM



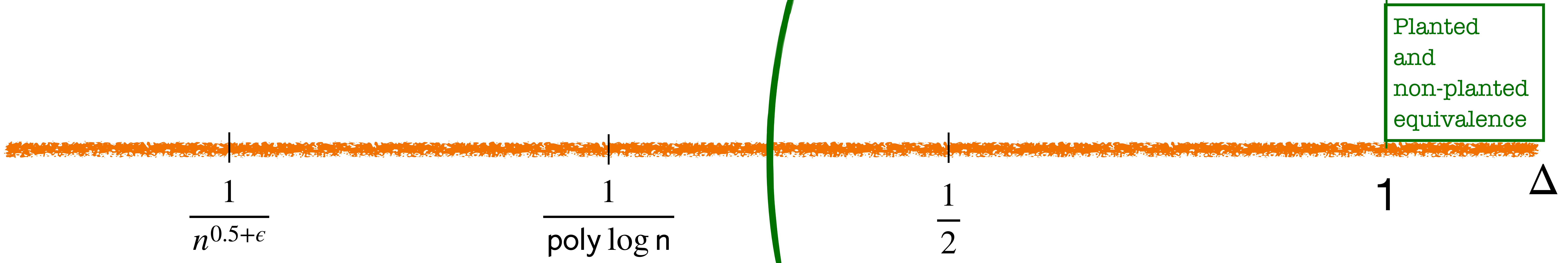
Conjecture : At $\Delta = 1$ best runtime of k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$ [Pet15, LLW19, DKK21].

Our Results

Dense regime

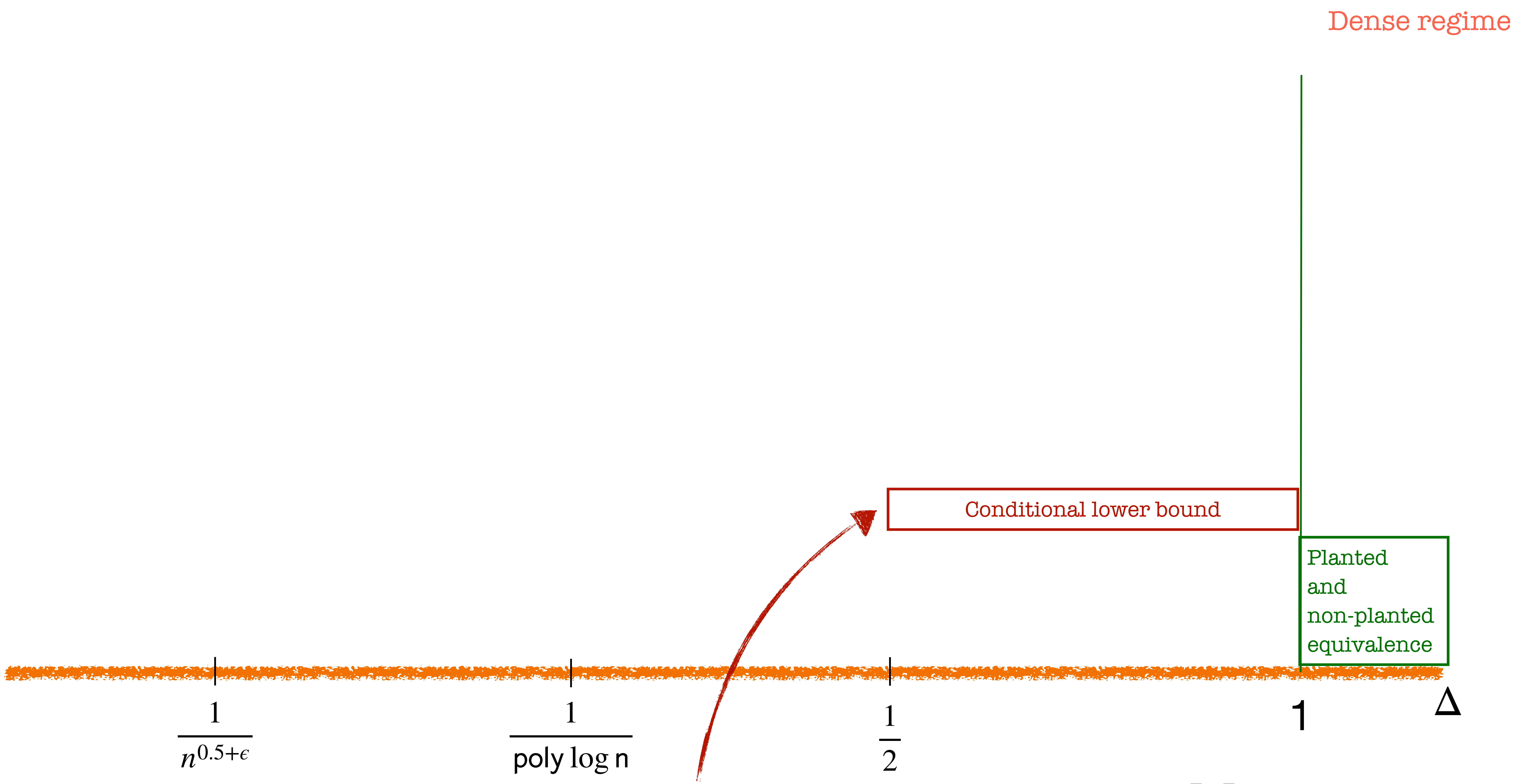
Planted k-SUM Distr. $\approx_{stat}$ Non-planted k-SUM Distr.

$\implies$ Algorithm for planted-kSUM solves non-planted k-SUM



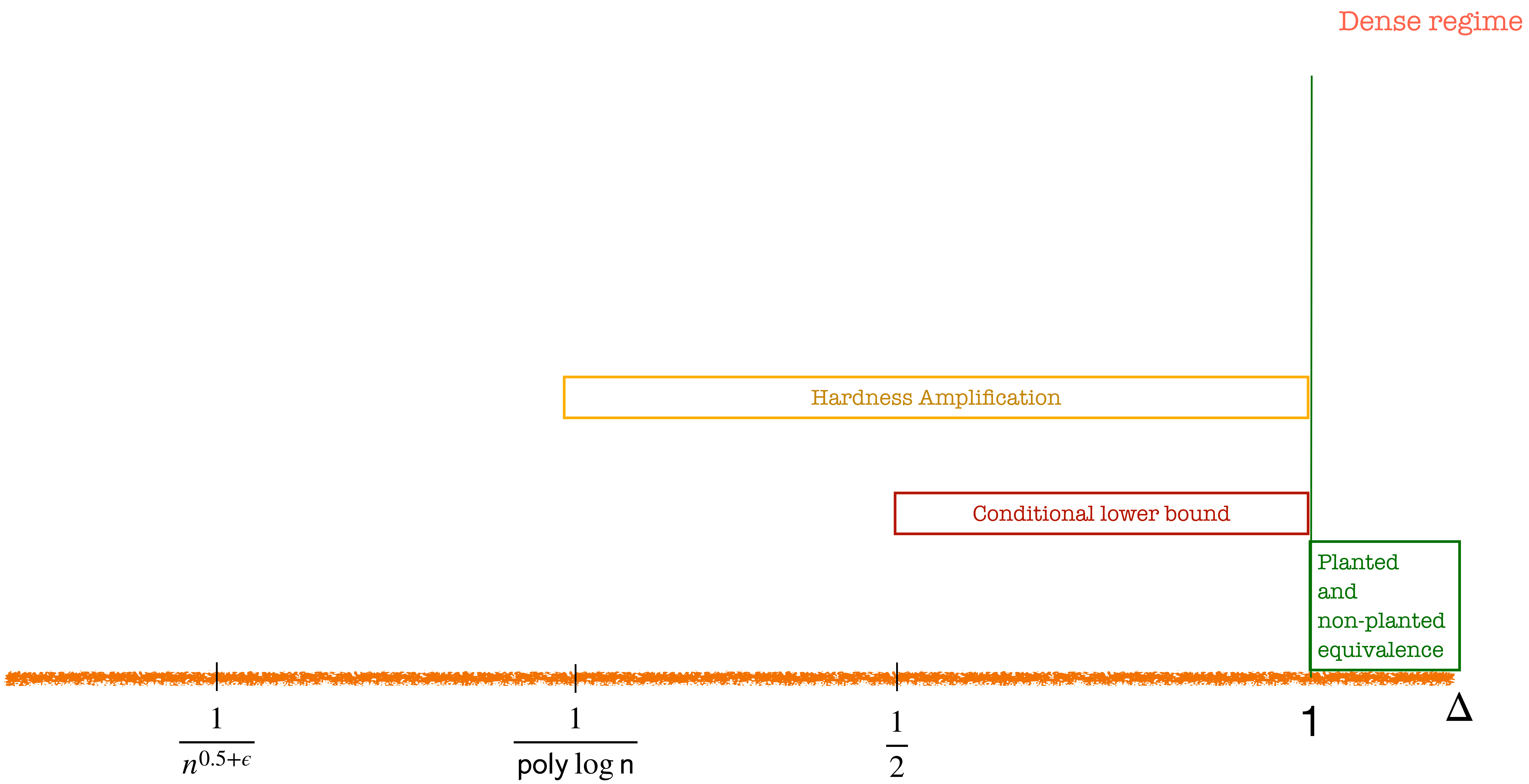
Conjecture : At $\Delta = 1$ best runtime of (planted) k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$

Our Results



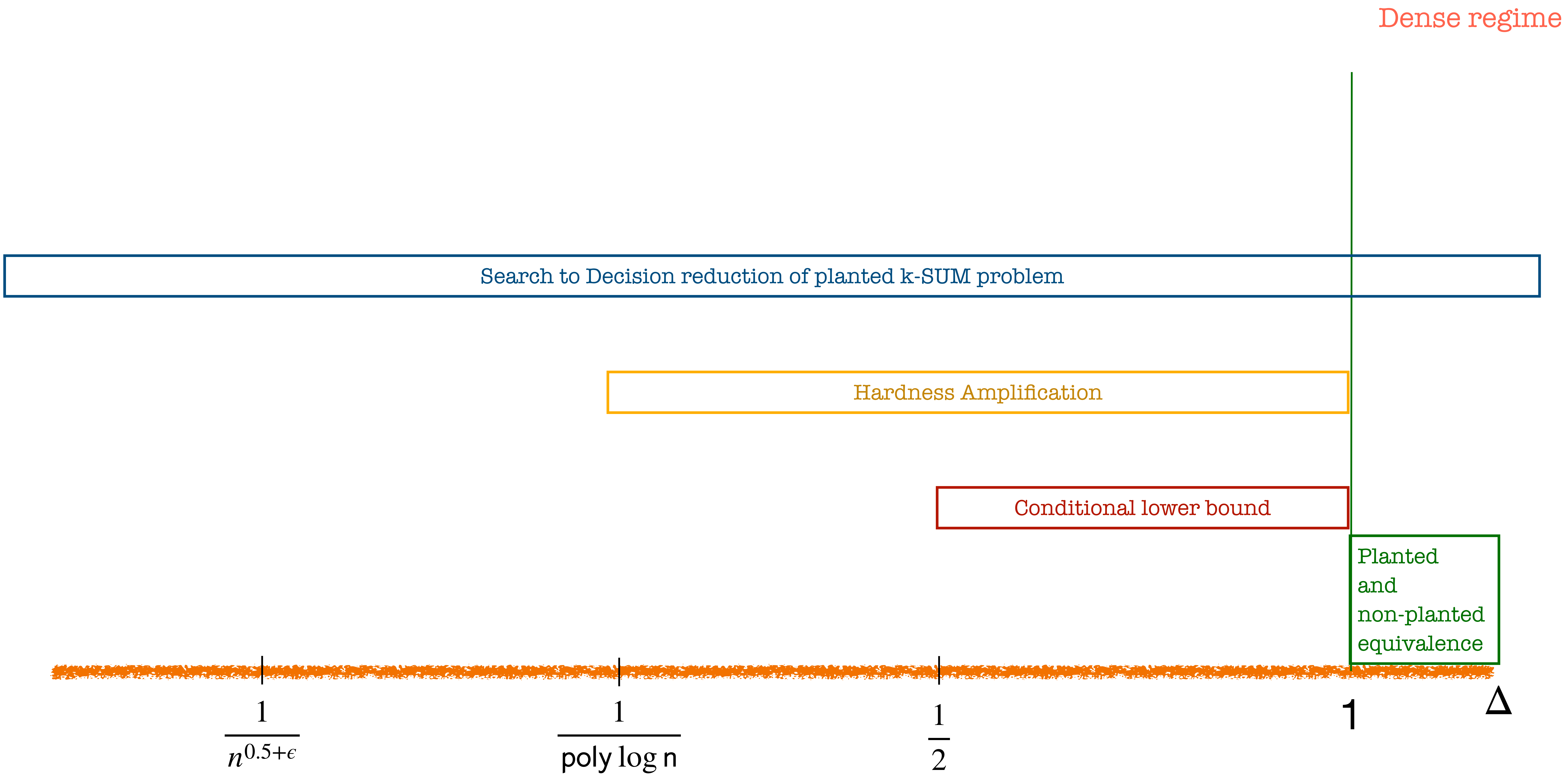
Conjecture : At $\Delta = 1$ best runtime of (planted) k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$

Our Results



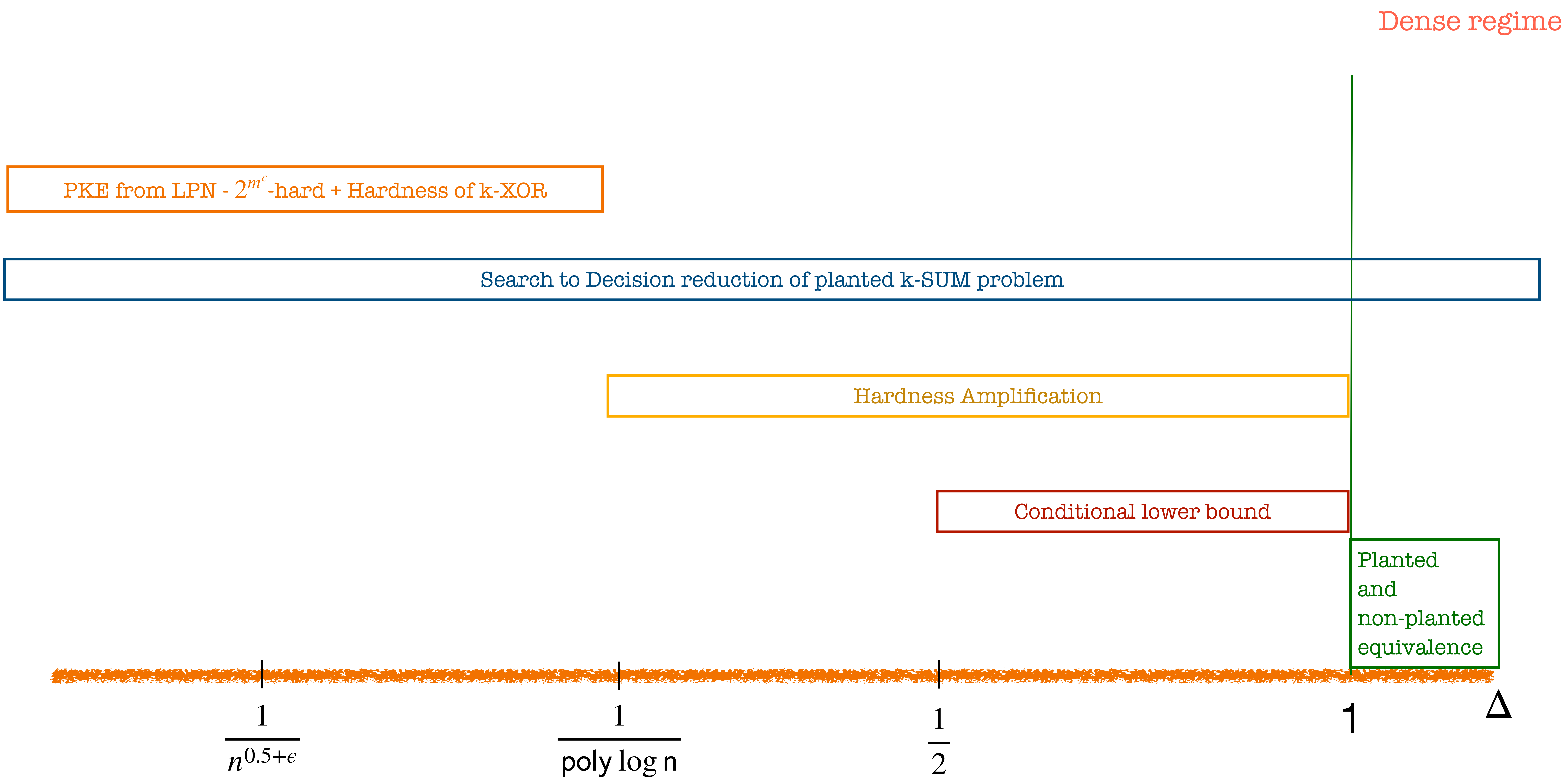
Conjecture : At $\Delta = 1$ best runtime of (planted) k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$

Our Results



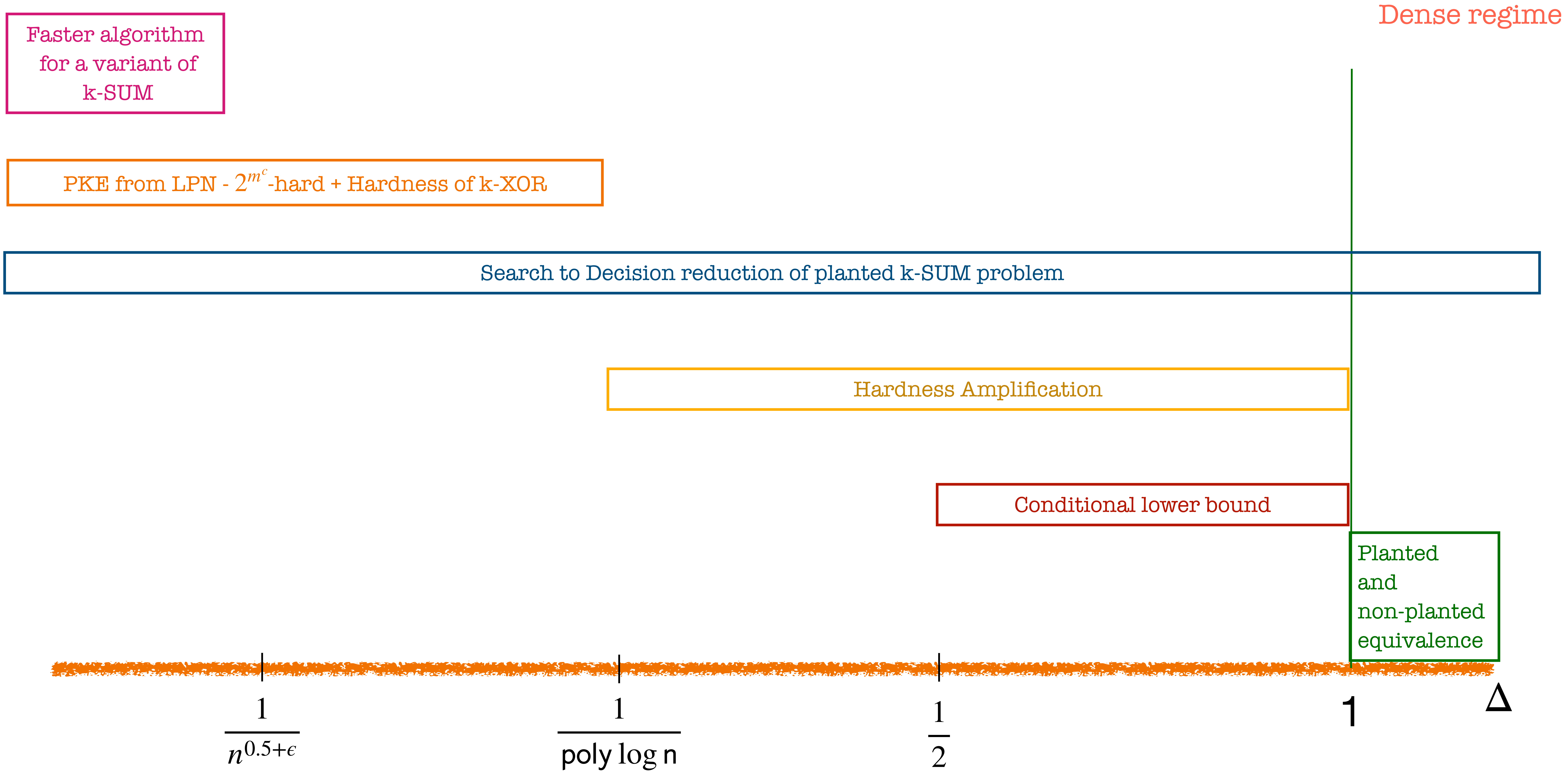
Conjecture : At $\Delta = 1$ best runtime of (planted) k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$

Our Results



Conjecture : At $\Delta = 1$ best runtime of (planted) k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$

Our Results



Conjecture : At $\Delta = 1$ best runtime of (planted) k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$

Planted k-SUM - cryptography - overview

Planted k-SUM - cryptography - overview

PKE

Not known to

A certain hardness of LPN

PKE : Public Key Encryption

Planted k-SUM - cryptography - overview

PKE

Not known to

A certain hardness of LPN

PKE

Not known to

Variant of k-SUM at certain density

PKE : Public Key Encryption

Planted k-SUM - cryptography - overview

PKE

Not known to

A certain hardness of LPN

+

PKE

PKE

Not known to

Variant of k-SUM at certain density

PKE : Public Key Encryption

Planted k-SUM - cryptography - preliminaries

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$$(A \quad s \quad \oplus \qquad e\ ) \qquad \approx_{comp} \qquad u$$

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$$(A \quad s \quad \oplus \qquad e\ ) \quad \approx_{comp} \quad u$$

$$A \xleftarrow{\$} \{0,1\}^{n\times m}$$

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$$s \xleftarrow{\$} \{0,1\}^{m \times 1}$$

$$(\,A \quad s \quad \oplus \qquad e\,) \quad \approx_{comp} \quad u$$

$$A \xleftarrow{\$} \{0,1\}^{n \times m}$$

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$s \xleftarrow{\$} \{0,1\}^{m\times 1}$

$$(A \quad s \quad \oplus \qquad e\ ) \quad \approx_{comp} \quad u$$

$A \xleftarrow{\$} \{0,1\}^{n\times m} \qquad e \xleftarrow{\$} (\mathrm{Ber}_\eta)^{n\times 1}$

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$$s \xleftarrow{\$} \{0,1\}^{m\times 1}$$

$$(A \; s \oplus e) \approx_{comp} u$$

$$A \xleftarrow{\$} \{0,1\}^{n\times m} \qquad e \xleftarrow{\$} (\mathrm{Ber}_\eta)^{n\times 1} \qquad u \xleftarrow{\$} \{0,1\}^{n\times 1}$$

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$$s \overset{\$}{\leftarrow} \{0,1\}^{m\times 1}$$

$$(A \; s \; \oplus \; e) \approx_{comp} u$$

$$A \overset{\$}{\leftarrow} \{0,1\}^{n\times m} \qquad e \overset{\$}{\leftarrow} (\mathrm{Ber}_{\eta})^{n\times 1} \qquad u \overset{\$}{\leftarrow} \{0,1\}^{n\times 1}$$

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$s \xleftarrow{\$} \{0,1\}^{m\times 1}$

$$(A \quad s \quad \oplus \quad e) \quad \approx_{comp} \quad u$$

$A \xleftarrow{\$} \{0,1\}^{n\times m}$ $\quad e \xleftarrow{\$} (\mathrm{Ber}_\eta)^{n\times 1}$ $\quad u \xleftarrow{\$} \{0,1\}^{n\times 1}$

Error Rate η

Hardness

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$$s \xleftarrow{\$} \{0,1\}^{m\times 1}$$

$$(\,A \quad s \quad \oplus \quad e\,) \quad \approx_{comp} \quad u$$

$$A \xleftarrow{\$} \{0,1\}^{n\times m} \qquad e \xleftarrow{\$} (\mathrm{Ber}_\eta)^{n\times 1} \qquad u \xleftarrow{\$} \{0,1\}^{n\times 1}$$

Error Rate η	Constant
Hardness	PPT One-way Functions [Blu94]

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$$s \xleftarrow{\$} \{0,1\}^{m\times 1}$$

$$(\,A \quad s \;\oplus\; e\,) \approx_{comp} u$$

$A \xleftarrow{\$} \{0,1\}^{n\times m}$ $\quad e \xleftarrow{\$} (\mathrm{Ber}_\eta)^{n\times 1}$ $\quad u \xleftarrow{\$} \{0,1\}^{n\times 1}$

Error Rate η	Constant	Constant
Hardness	PPT	$2^{m^{0.51}}$
	One-way Functions [Blu94]	PKE [YZ16] CRHF[YZWGL19]

Planted k-SUM - cryptography - preliminaries

LPN (Learning Parity with Noise)

$$s \xleftarrow{\$} \{0,1\}^{m\times 1}$$

$$(\, A \; s \oplus e \,) \approx_{comp} u$$

$$A \xleftarrow{\$} \{0,1\}^{n\times m} \qquad e \xleftarrow{\$} (\mathrm{Ber}_\eta)^{n\times 1} \qquad u \xleftarrow{\$} \{0,1\}^{n\times 1}$$

Error Rate η	Constant	Constant	$1/\sqrt{m}$
Hardness	PPT	$2^{m^{0.51}}$	PPT
	One-way Functions [Blu94]	PKE [YZ16] CRHF[YZWGL19]	PKE [Ale03, DMN12, KMP14]

Planted k-SUM - cryptography - preliminaries

k-XOR - Variant of k-SUM

Planted k-XOR instance

Planted k-SUM - cryptography - preliminaries

k-XOR - Variant of k-SUM

Planted k-XOR instance

But each element is a m-dimensional binary vector

$$a_i \in \{0,1\}^m$$

Planted k-SUM - cryptography - preliminaries

A certain hardness of LPN

$\eta = \text{constant}$

$2^{m^c} : c \in (0,0.5)$

$+$

k-XOR at certain density

$$\Delta = \frac{1}{poly\log(n)}$$

$\Rightarrow$ PKE

PKE

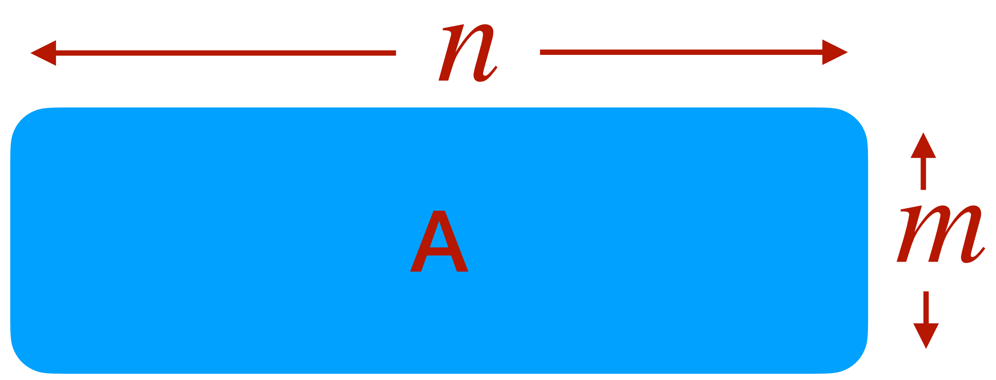
PKE

(All matrix and vector elements are in {0,1})

PKE

(All matrix and vector elements are in {0,1})

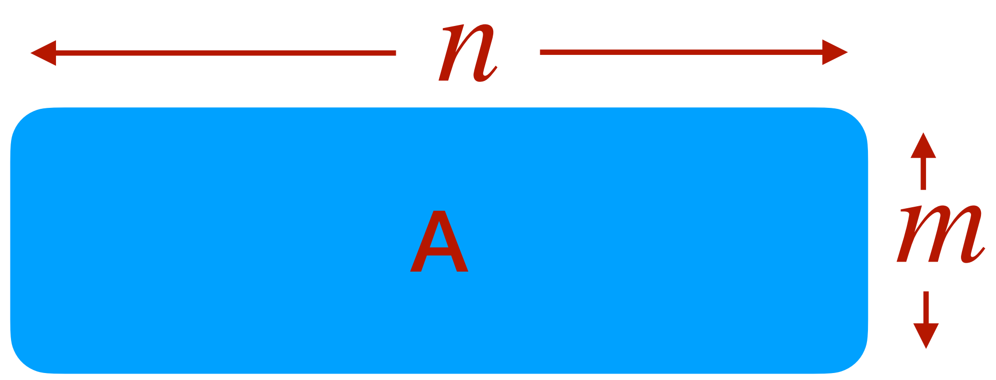
Public key :



PKE

(All matrix and vector elements are in {0,1})

Public key :



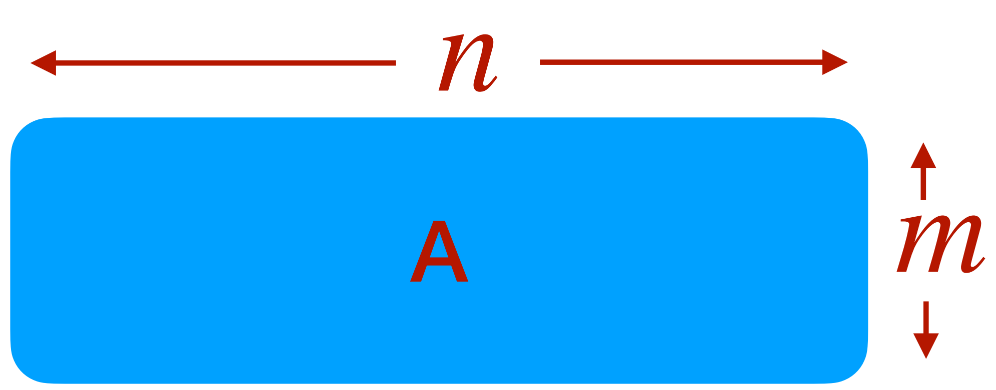
Secret key :

#1's = k

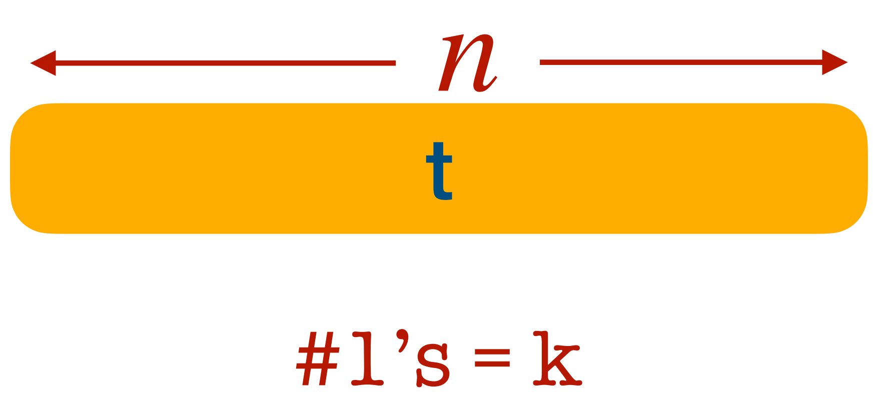
PKE

(All matrix and vector elements are in {0,1})

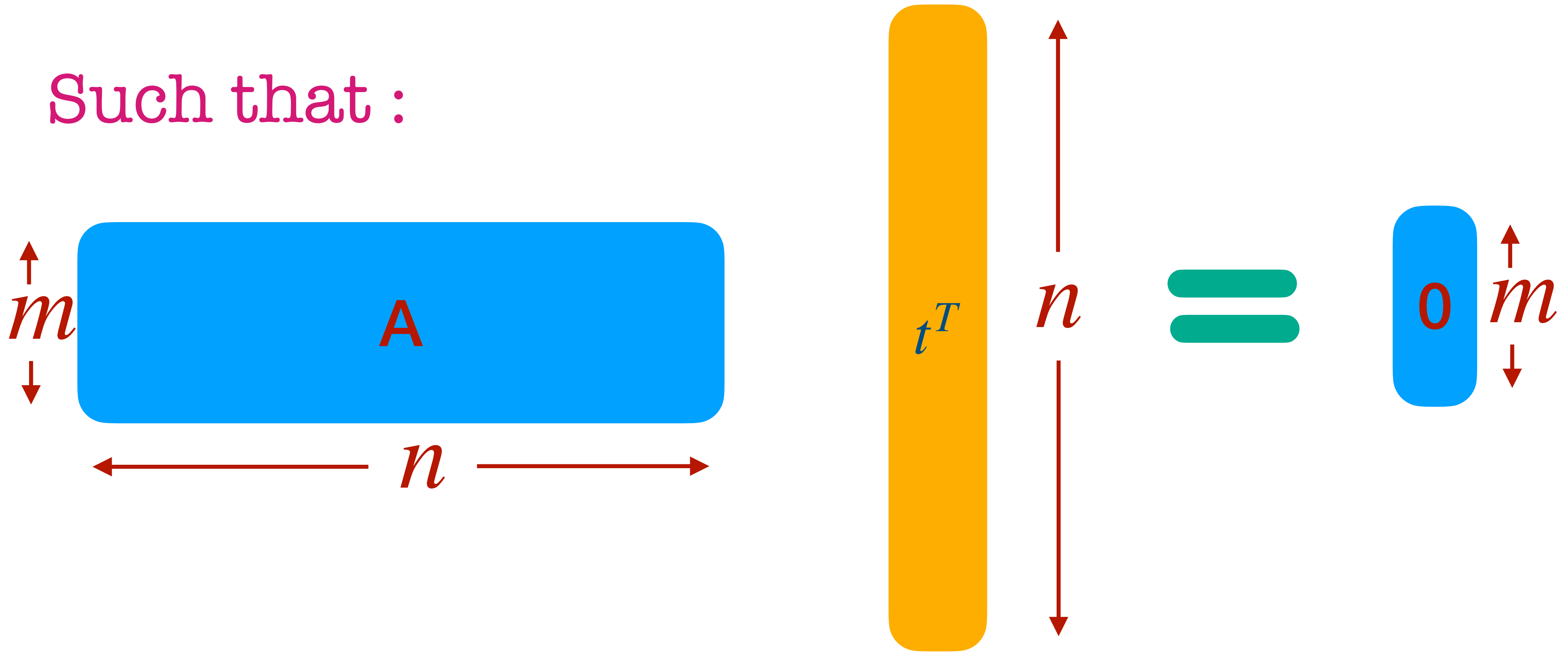
Public key :



Secret key :



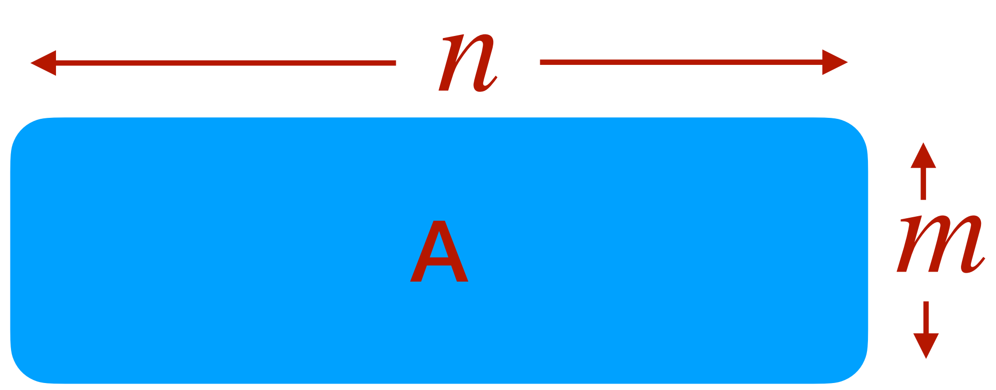
Such that :



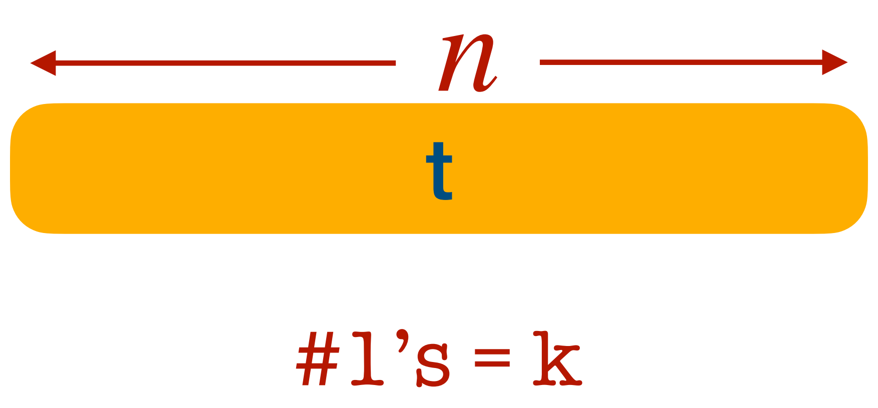
PKE

(All matrix and vector elements are in {0,1})

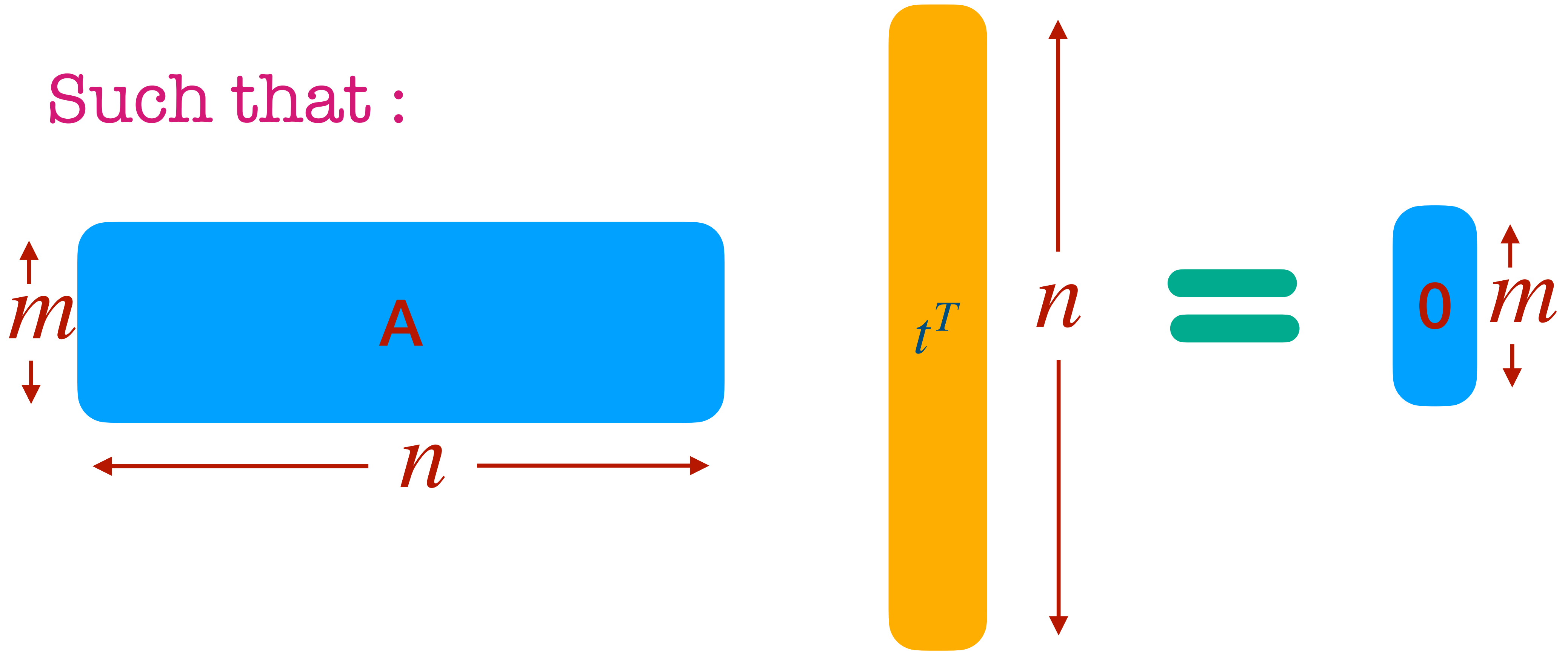
Public key :



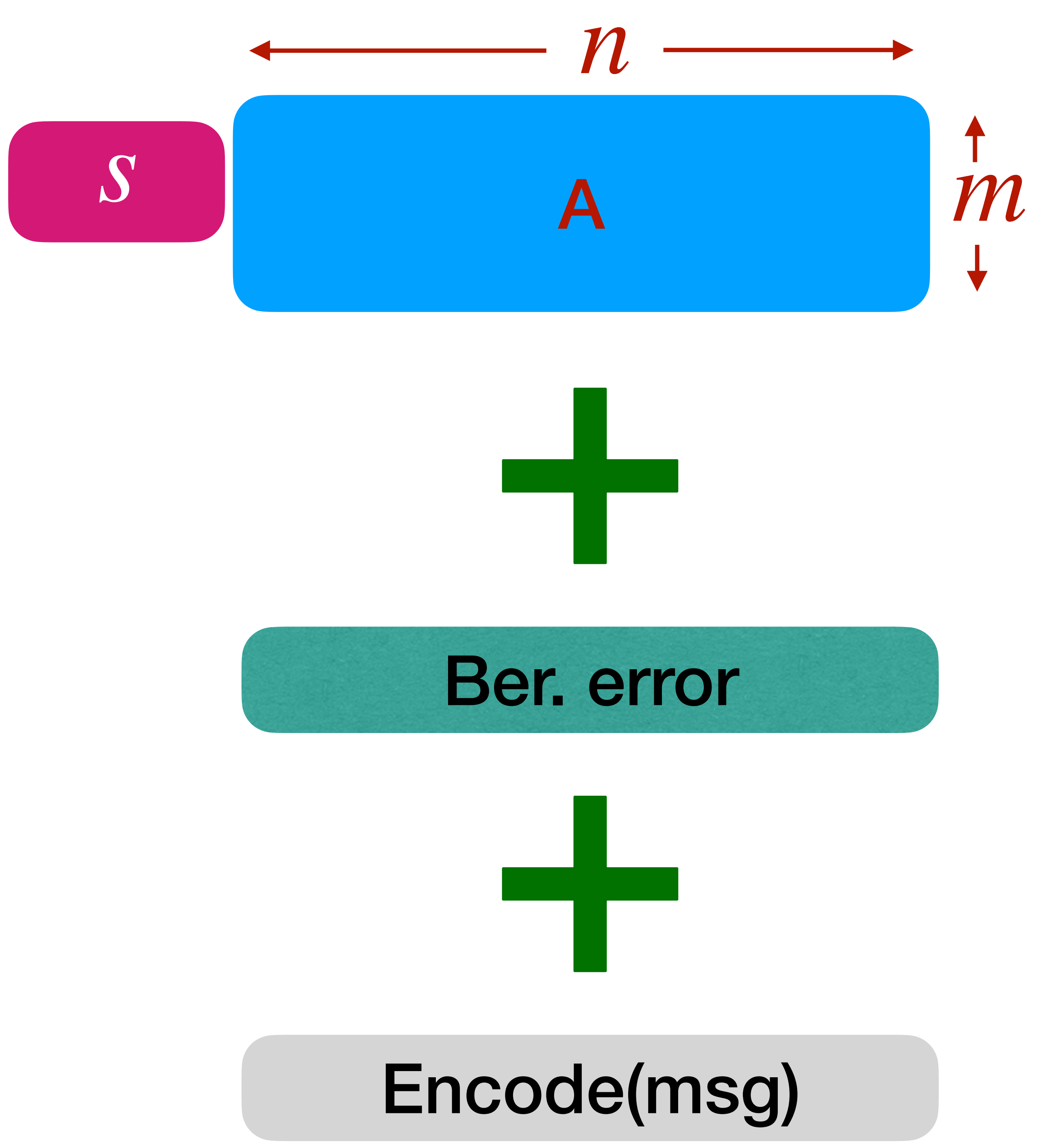
Secret key :



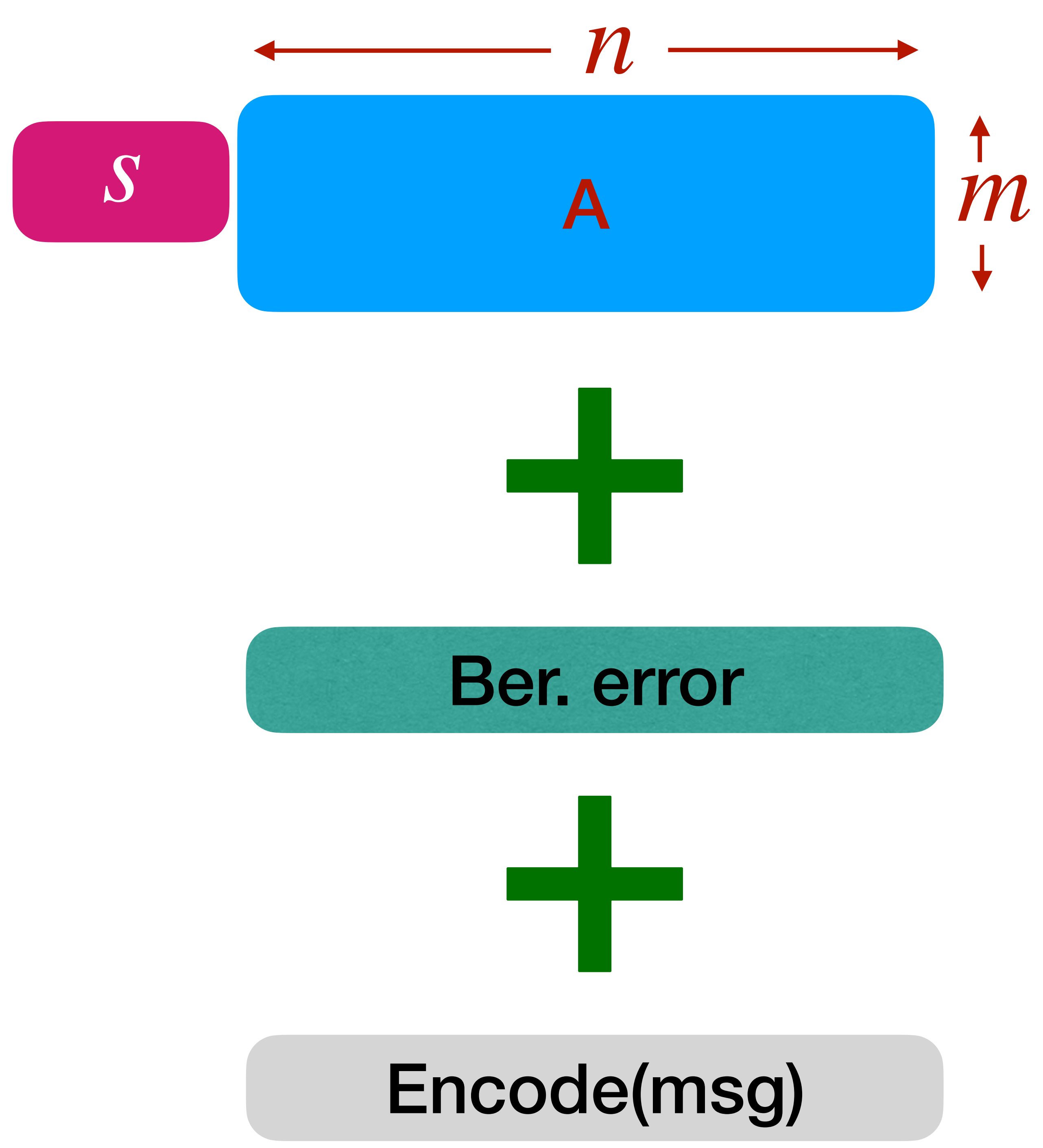
Such that :



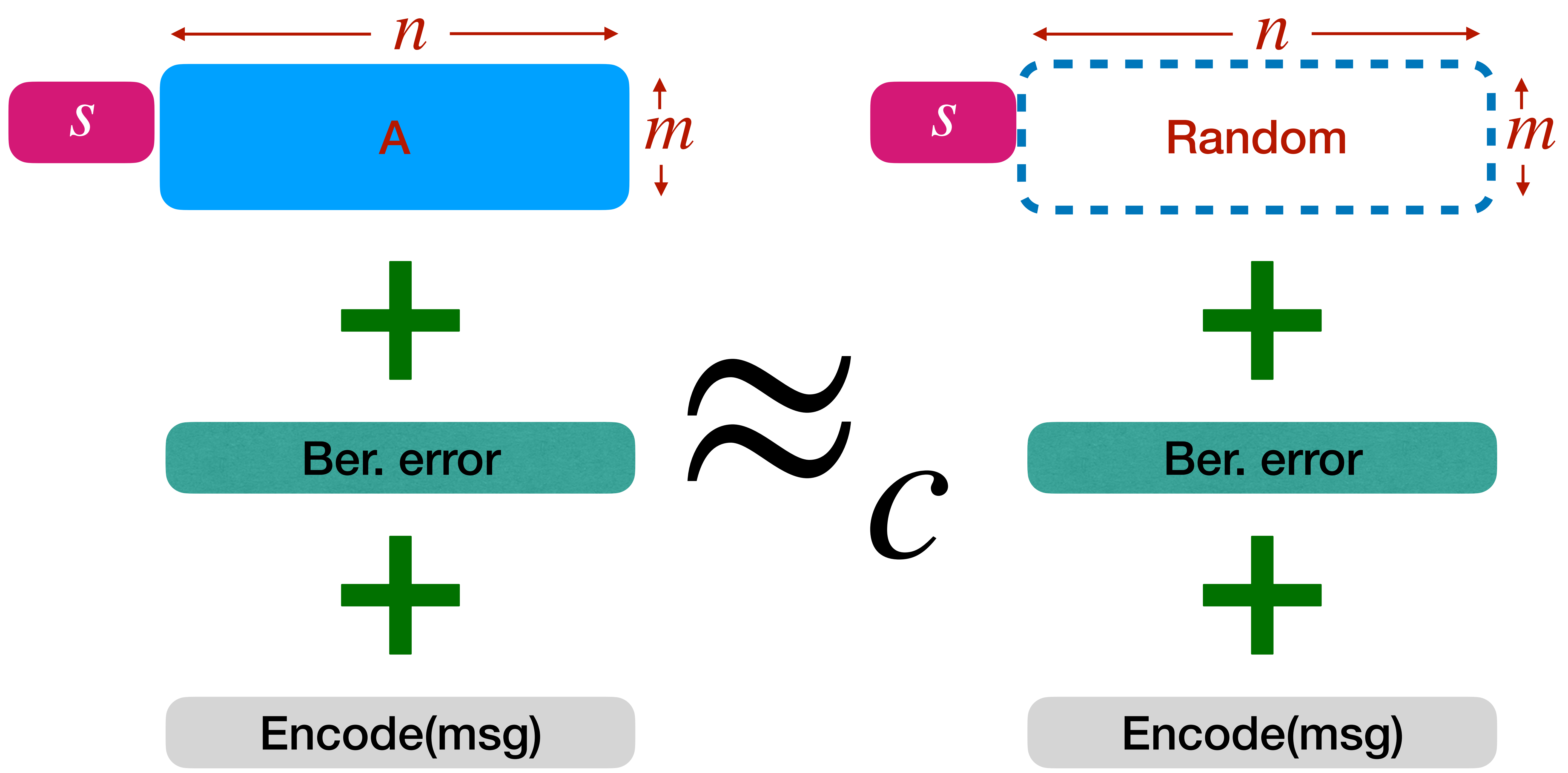
Encryption :



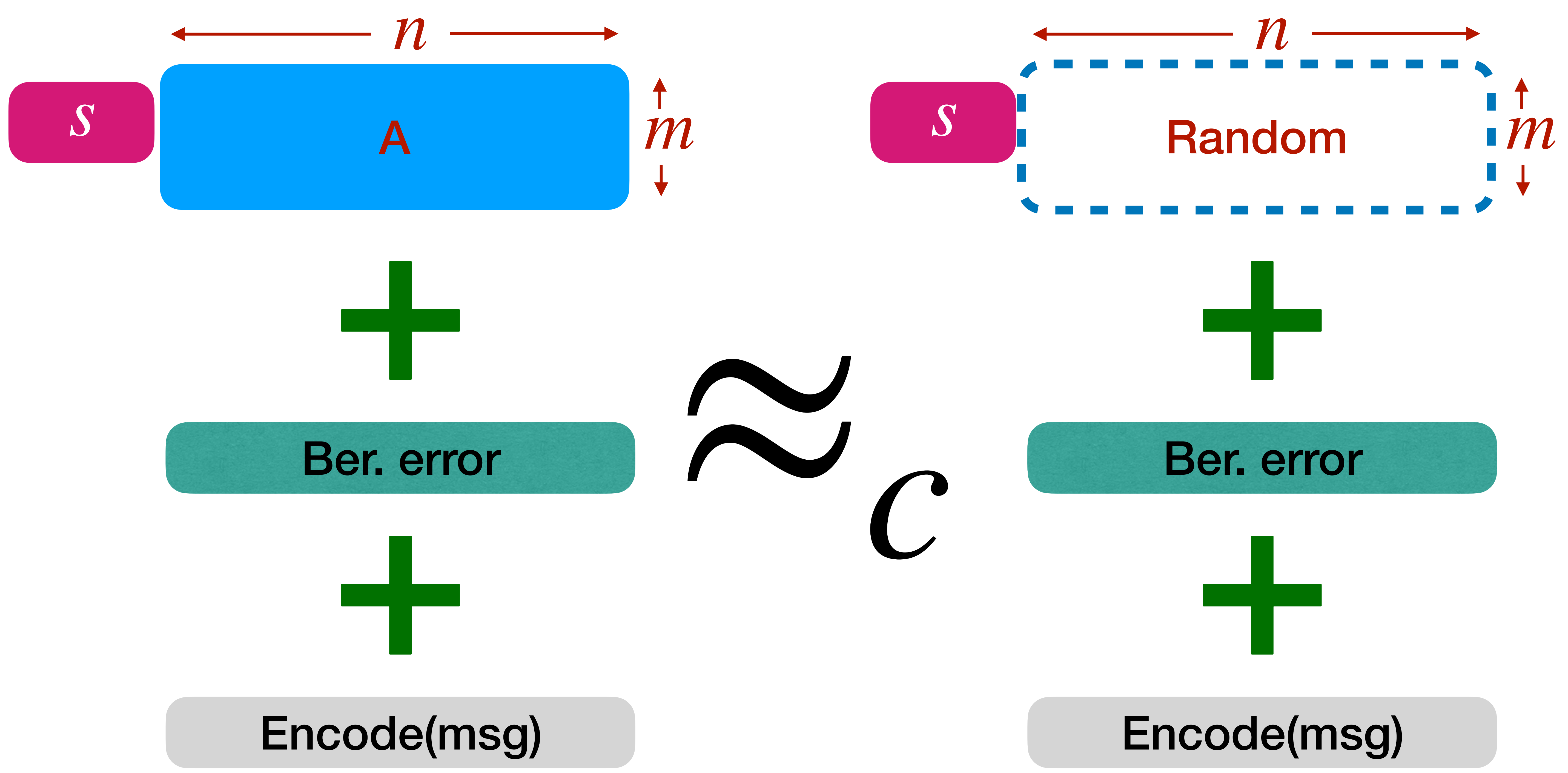
PKE Security



PKE Security



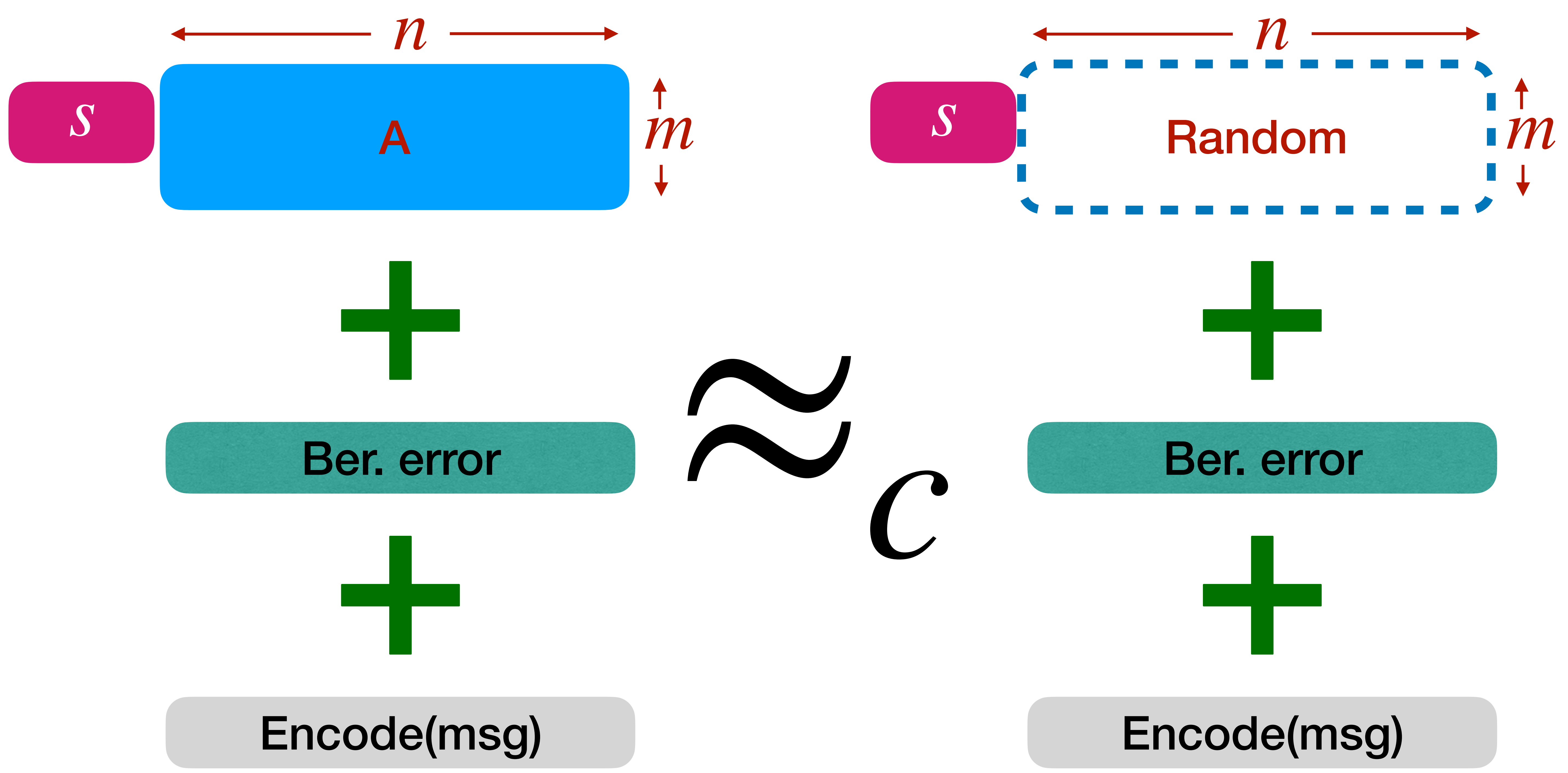
Hardness of k-XOR



PKE Security

$$\Delta = \frac{k \log n}{m}$$

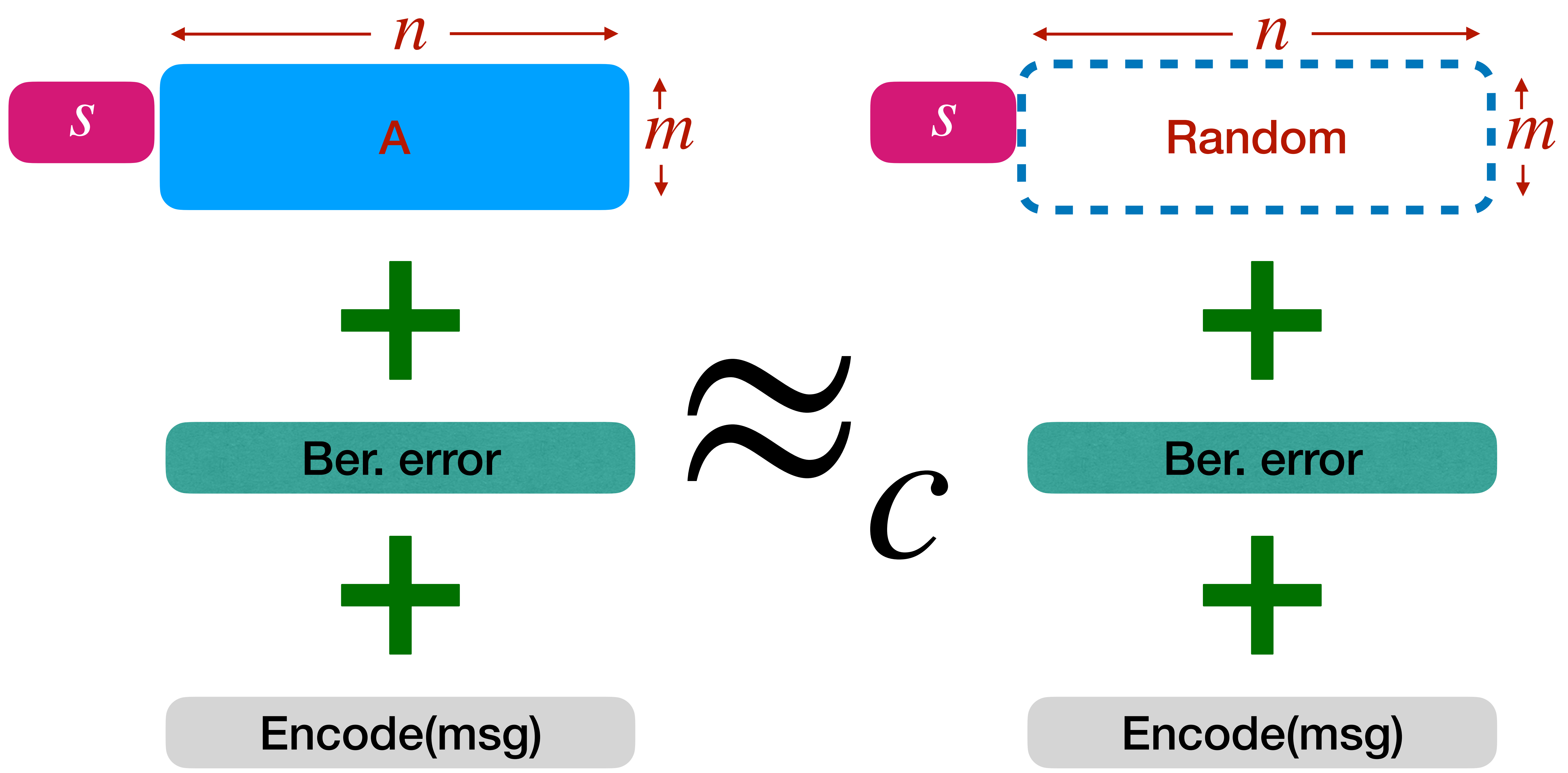
Hardness of k-XOR



PKE Security

$$\Delta = \frac{k \log n}{m} = \frac{1}{(\log n)^{\alpha}}$$

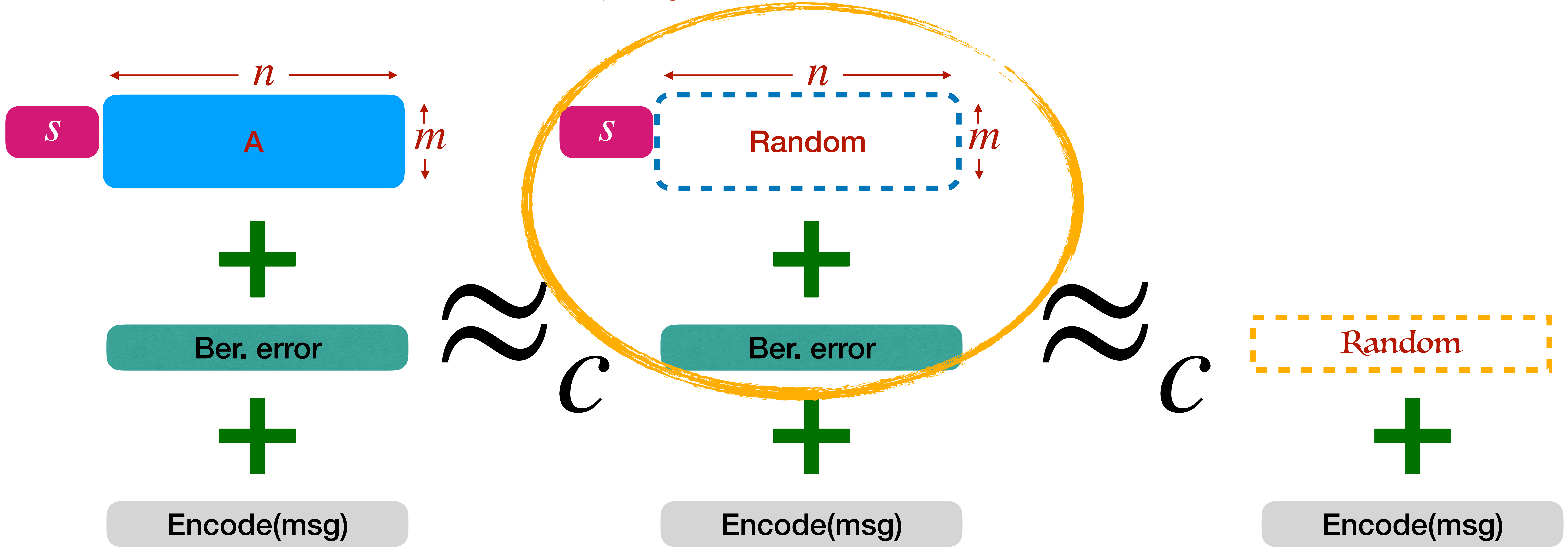
Hardness of k-XOR



PKE Security

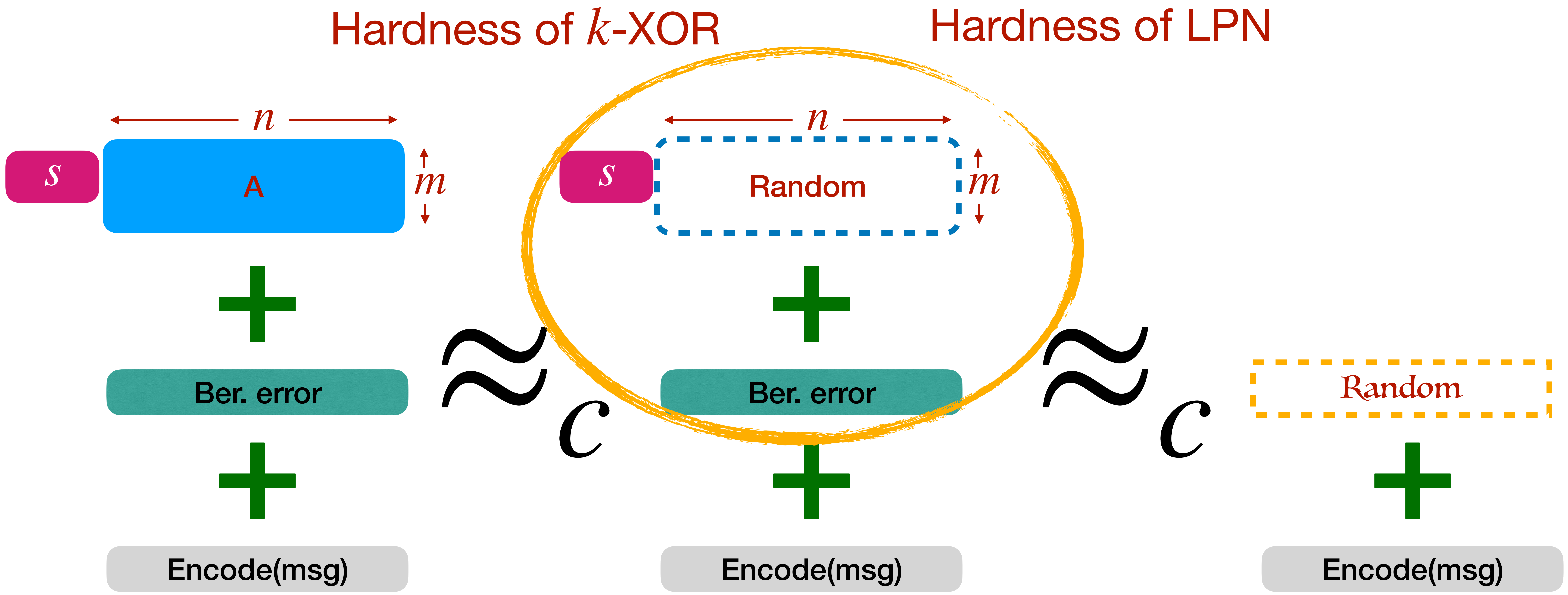
$$\Delta = \frac{k \log n}{m} = \frac{1}{(\log n)^{\alpha}}$$

Hardness of k-XOR



PKE Security

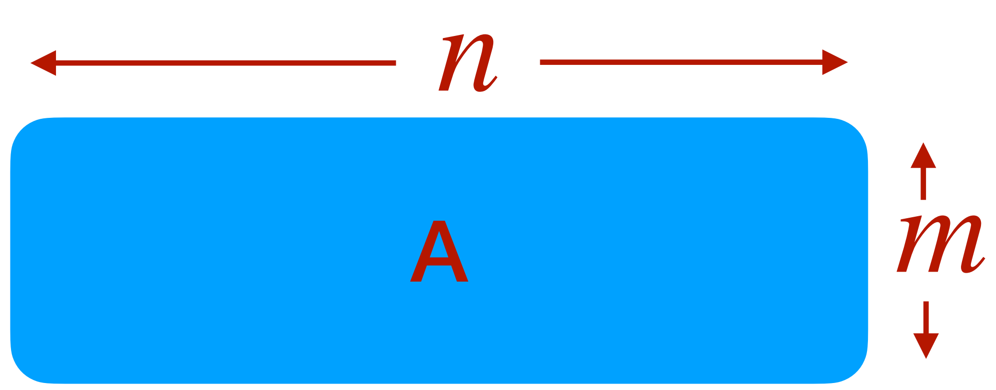
$$\Delta = \frac{k \log n}{m} = \frac{1}{(\log n)^{\alpha}}$$



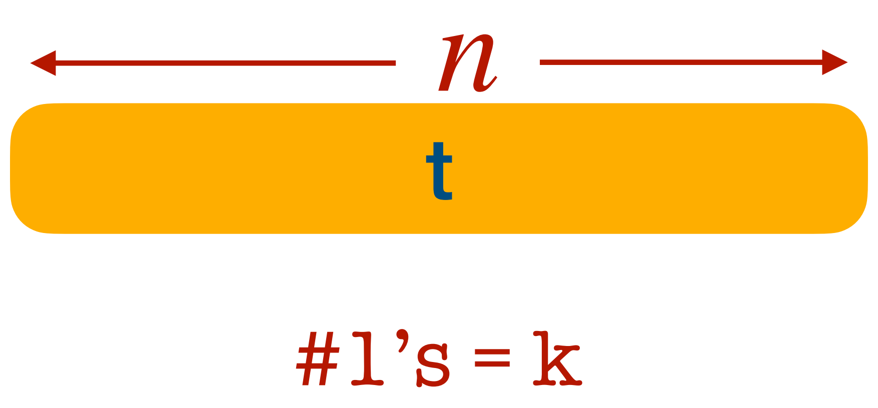
PKE

(All matrix and vector elements are in {0,1})

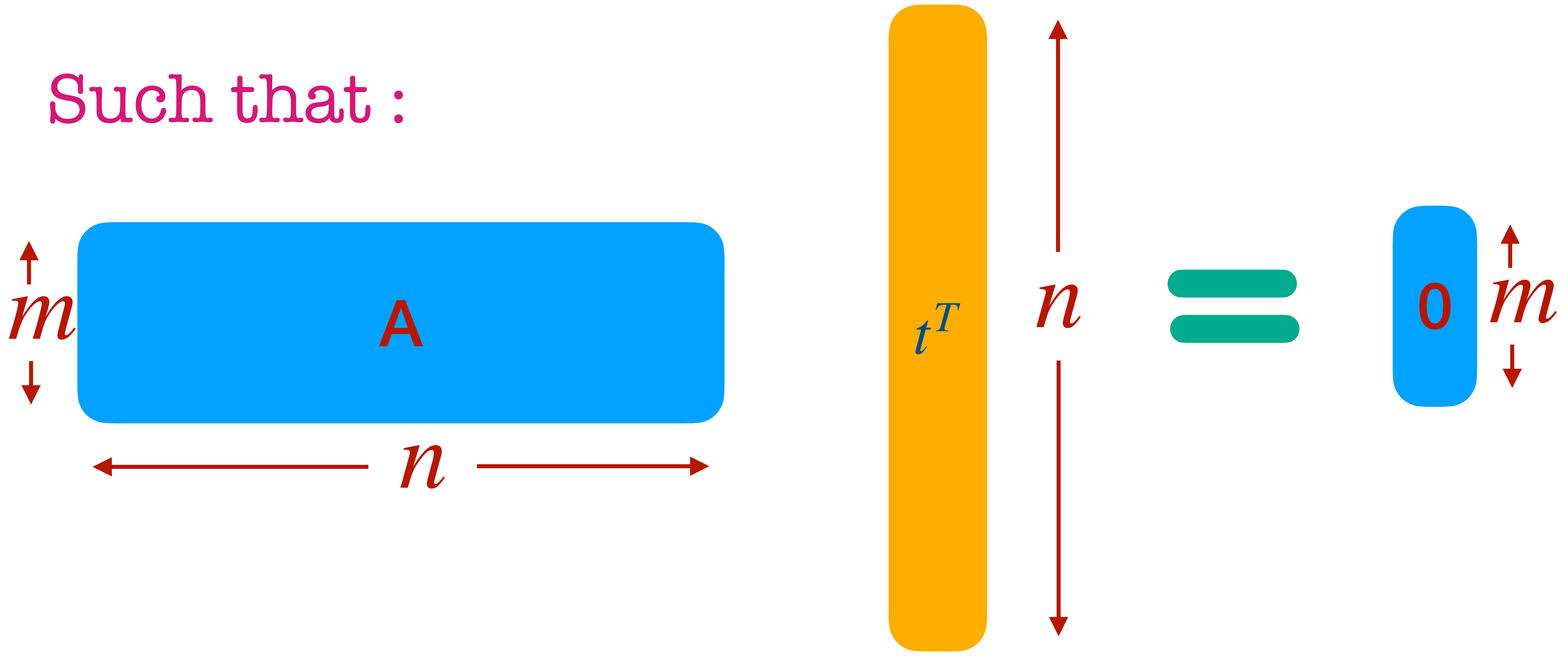
Public key :



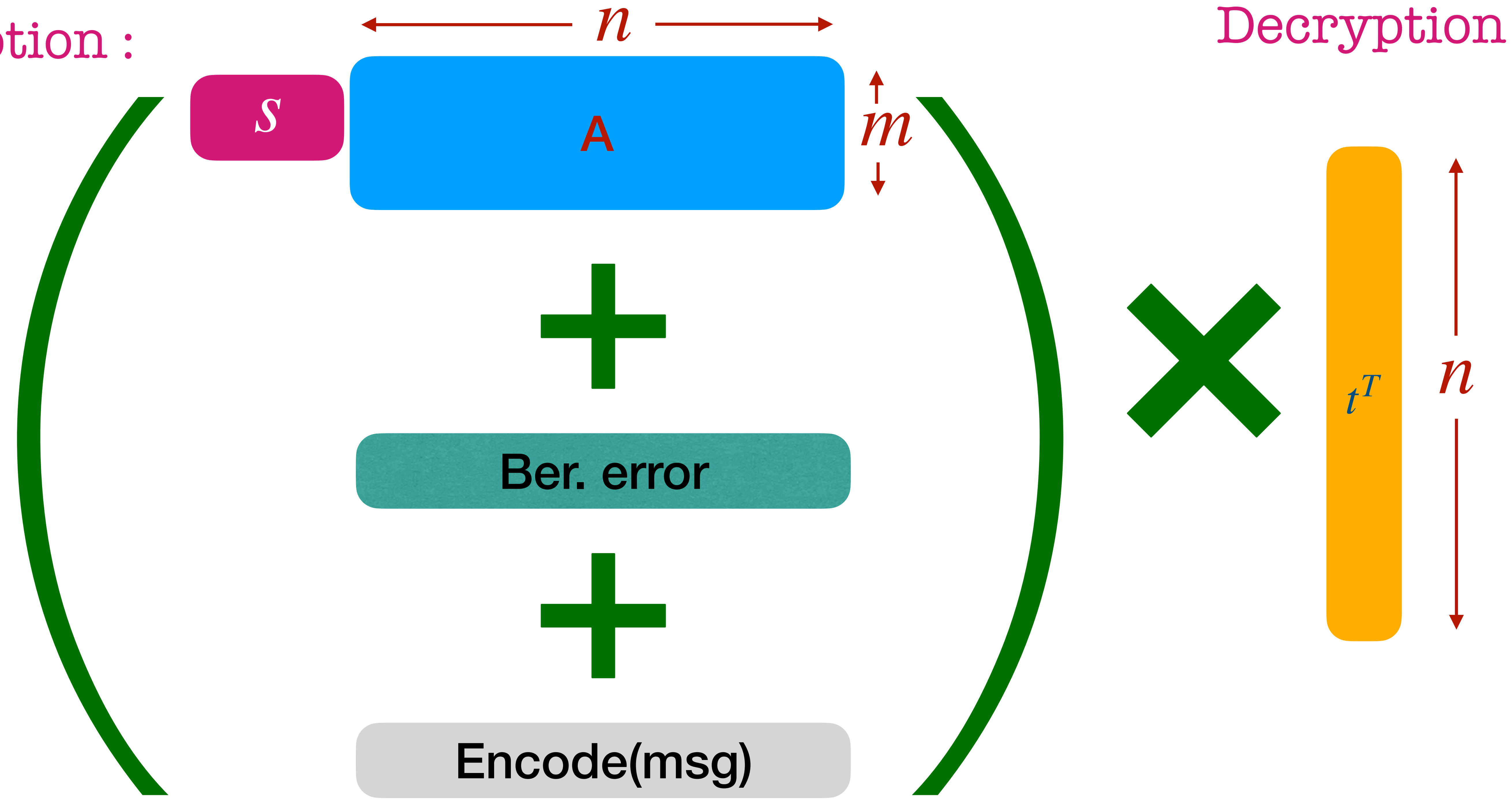
Secret key :



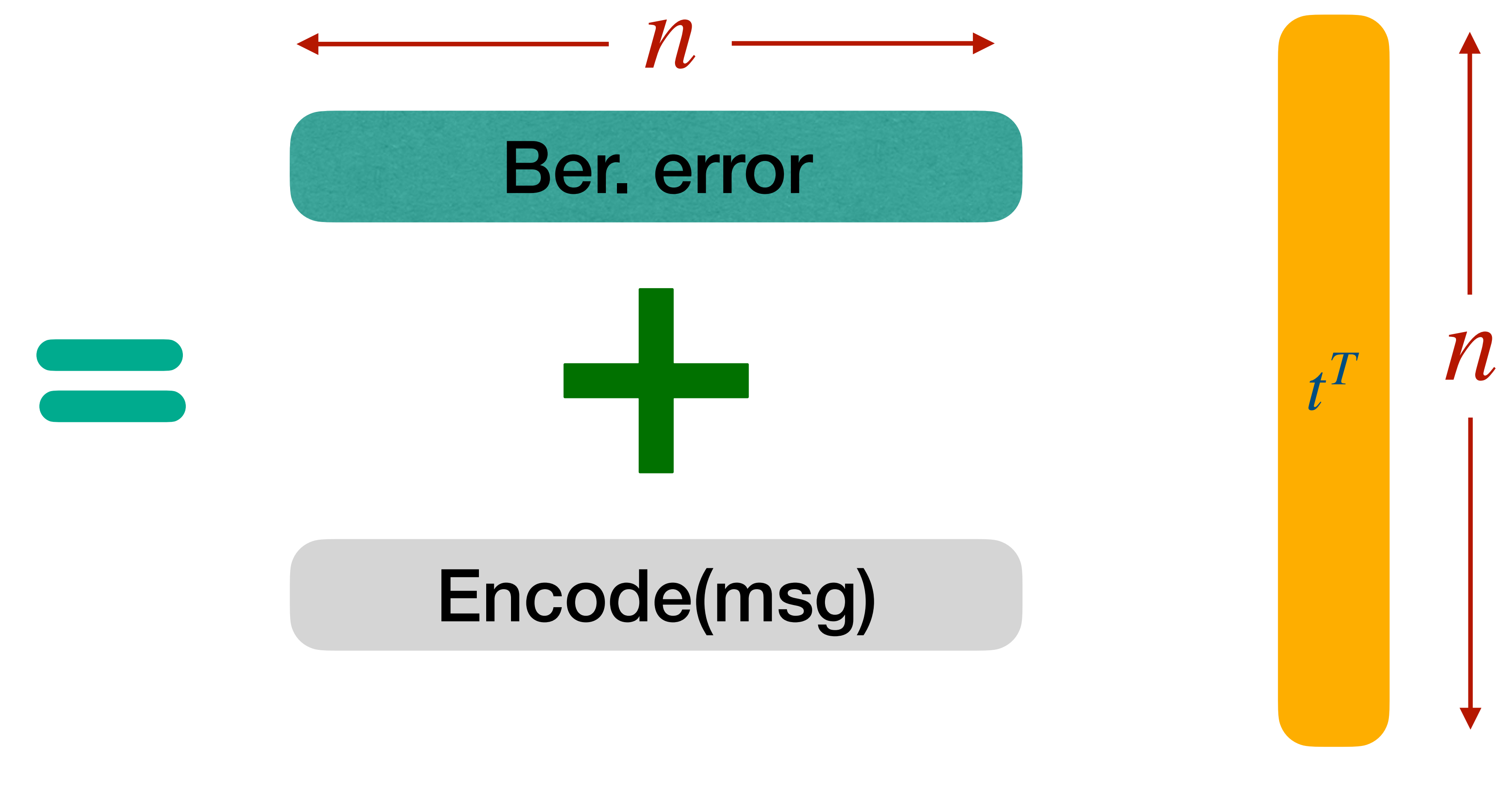
Such that :



Encryption :



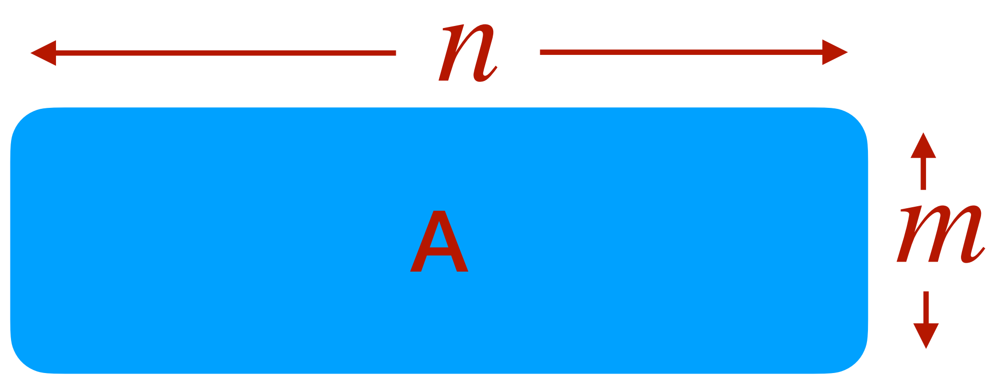
Decryption :



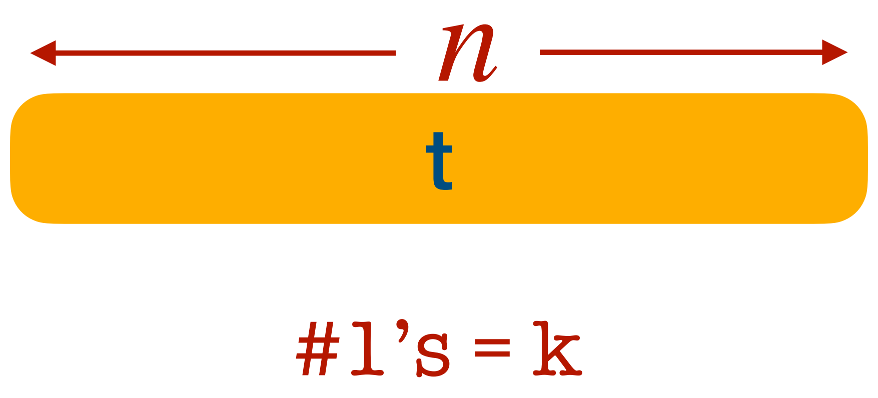
PKE

(All matrix and vector elements are in {0,1})

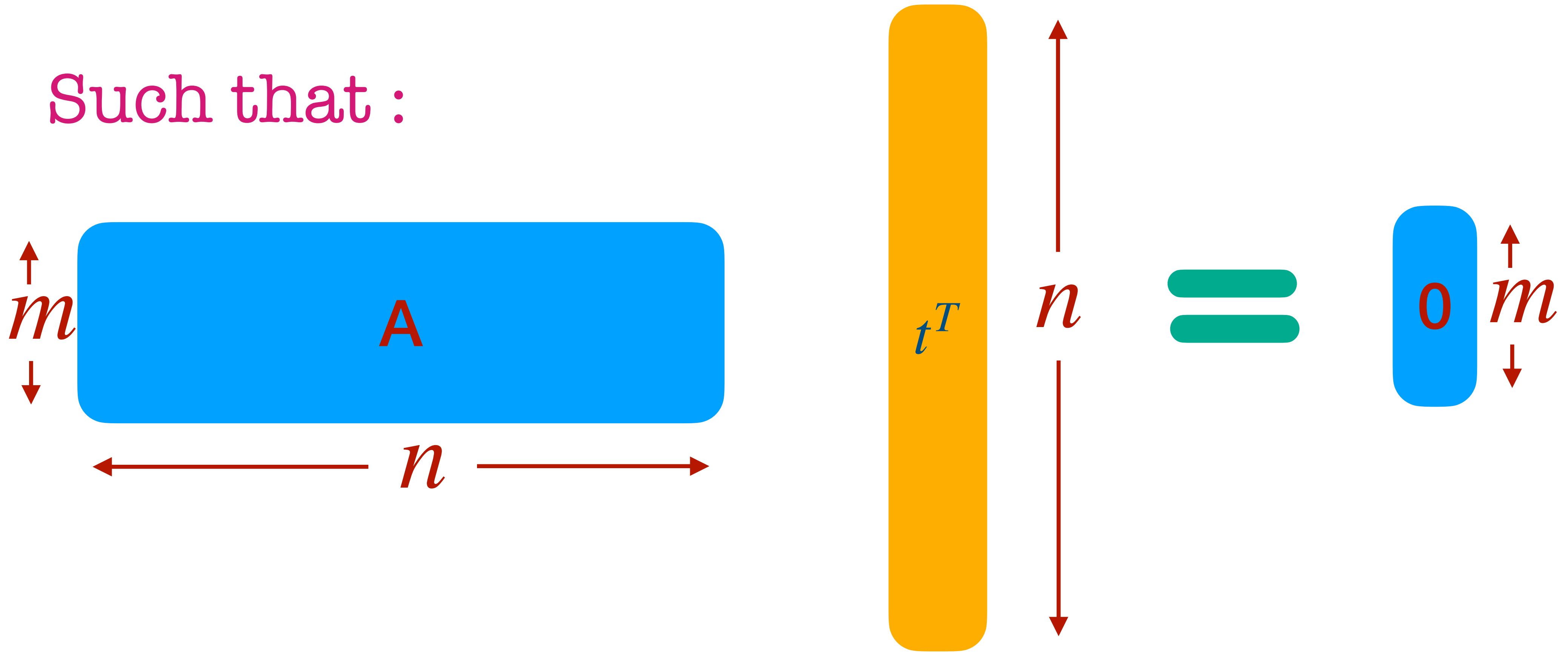
Public key :



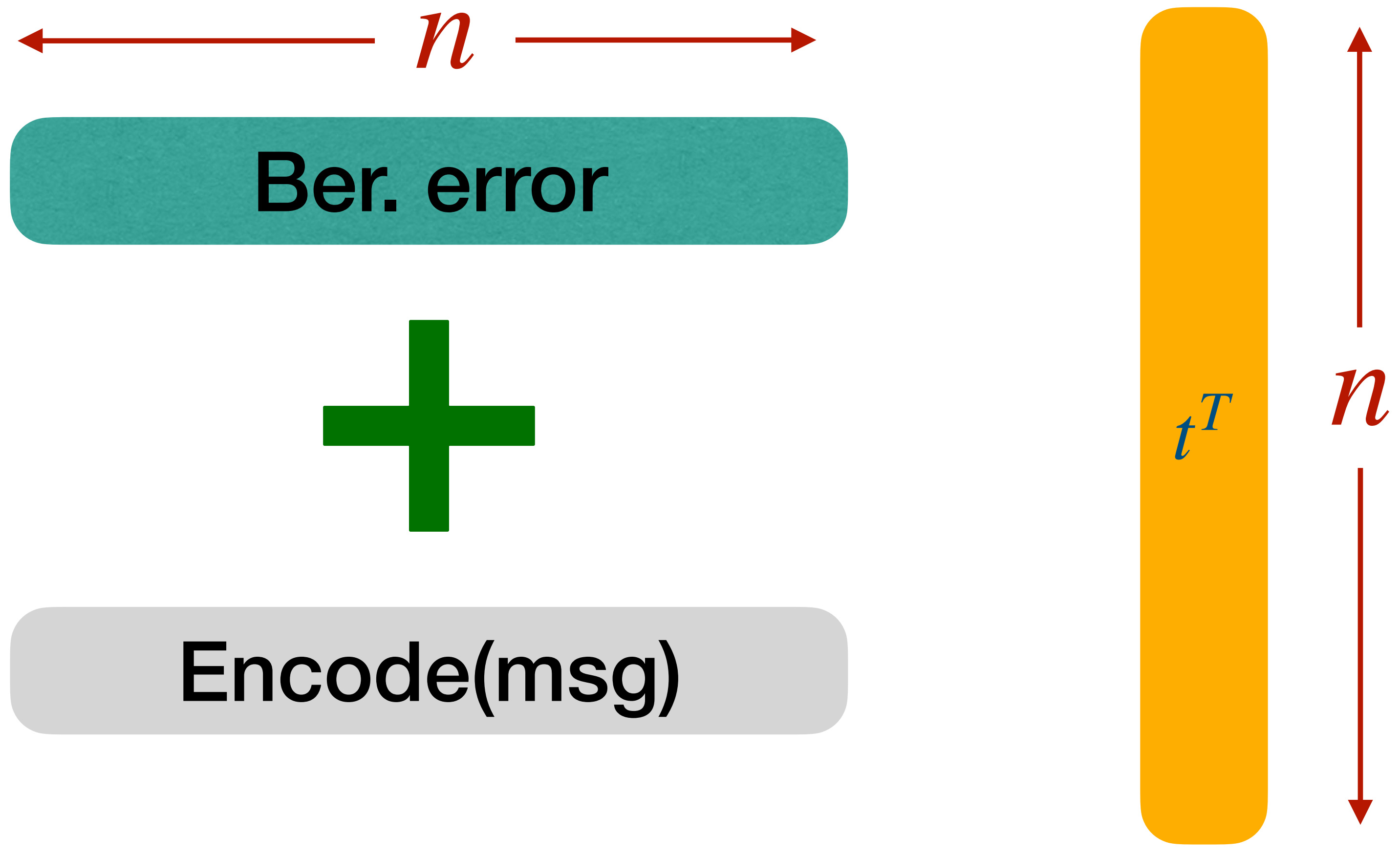
Secret key :



Such that :



Decryption :

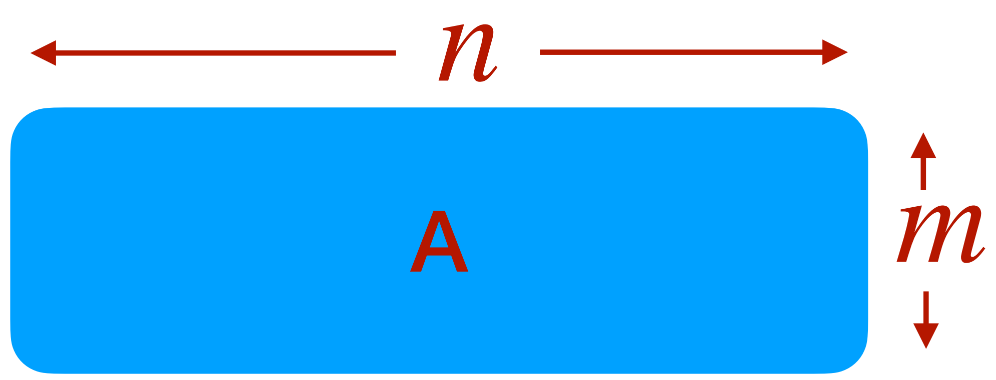


Decryption is possible only if k <= log(n) given Ber. error is constant- more #1's, more error accumulation

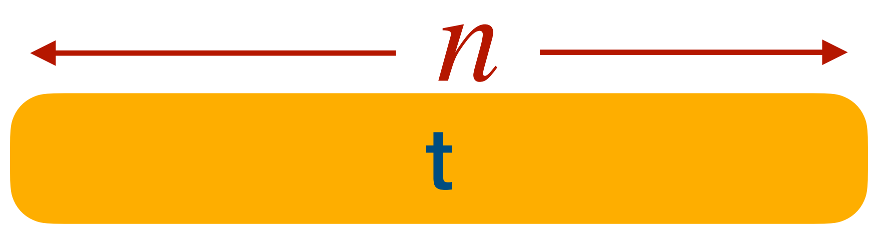
PKE

(All matrix and vector elements are in {0,1})

Public key :

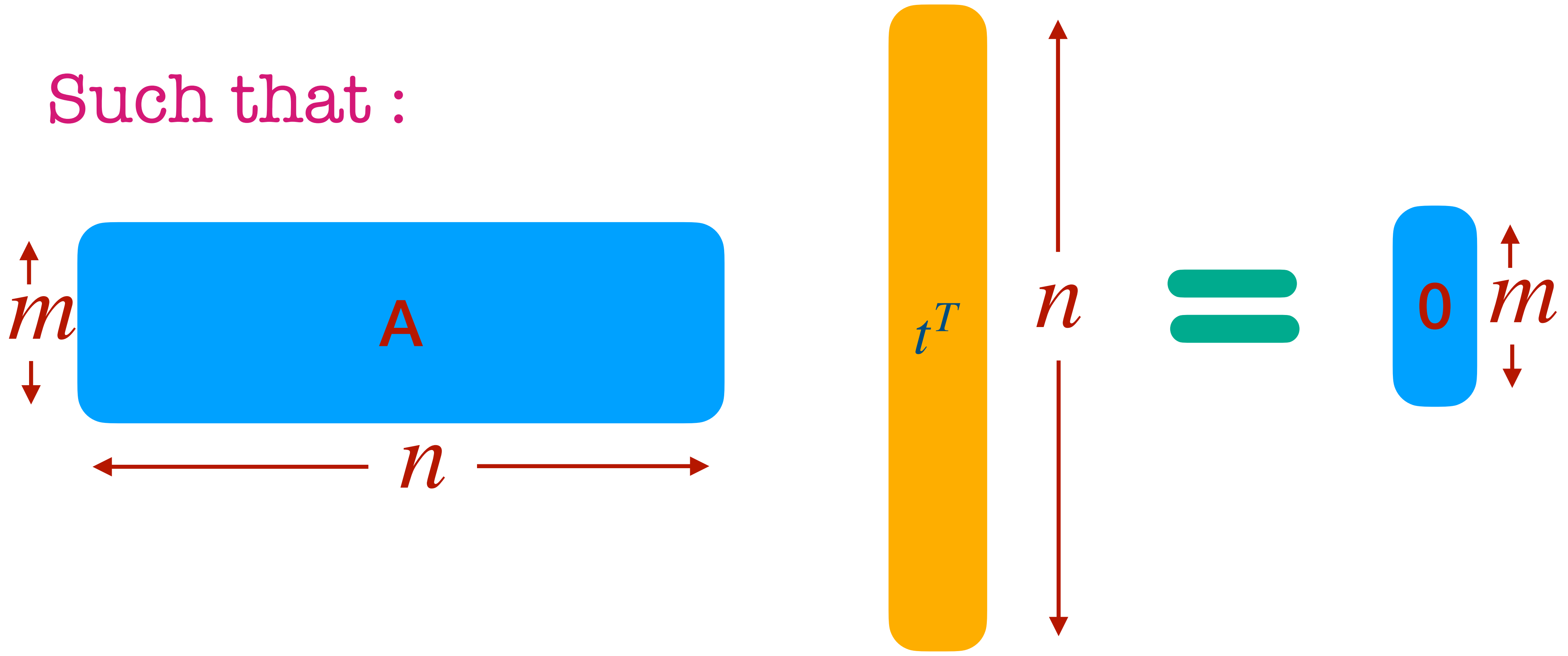


Secret key :

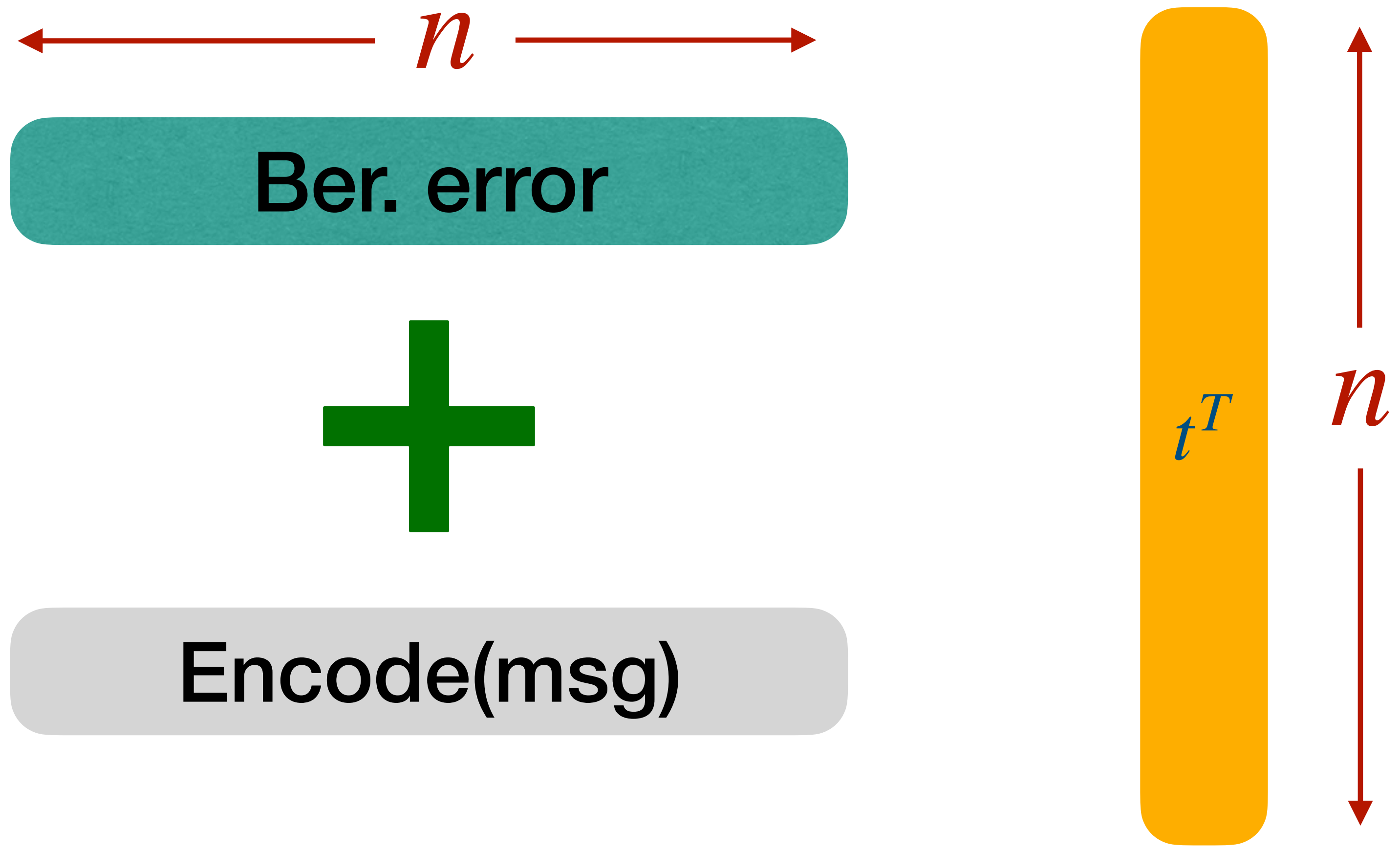


#1's = k

Such that :



Decryption :



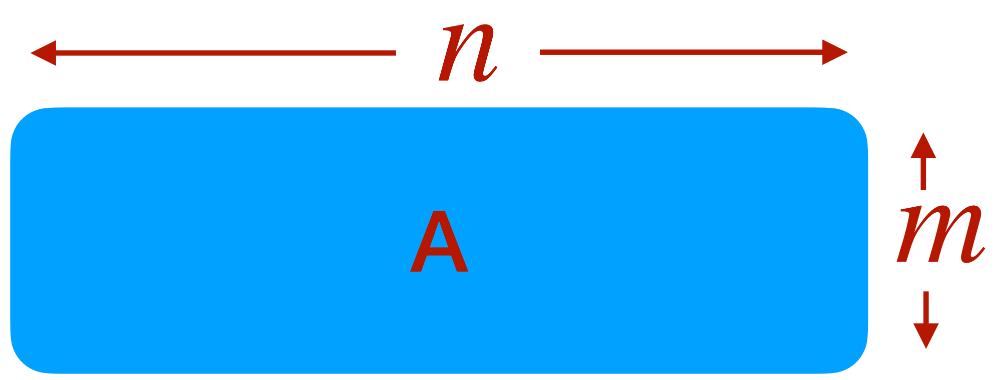
$$m = \frac{k \log n}{\Delta}$$

Decryption is possible only if k <= log(n) given Ber. error is constant- more #1's, more error accumulation

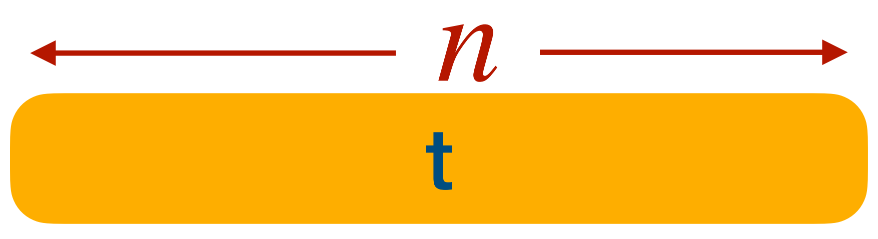
PKE

(All matrix and vector elements are in {0,1})

Public key :

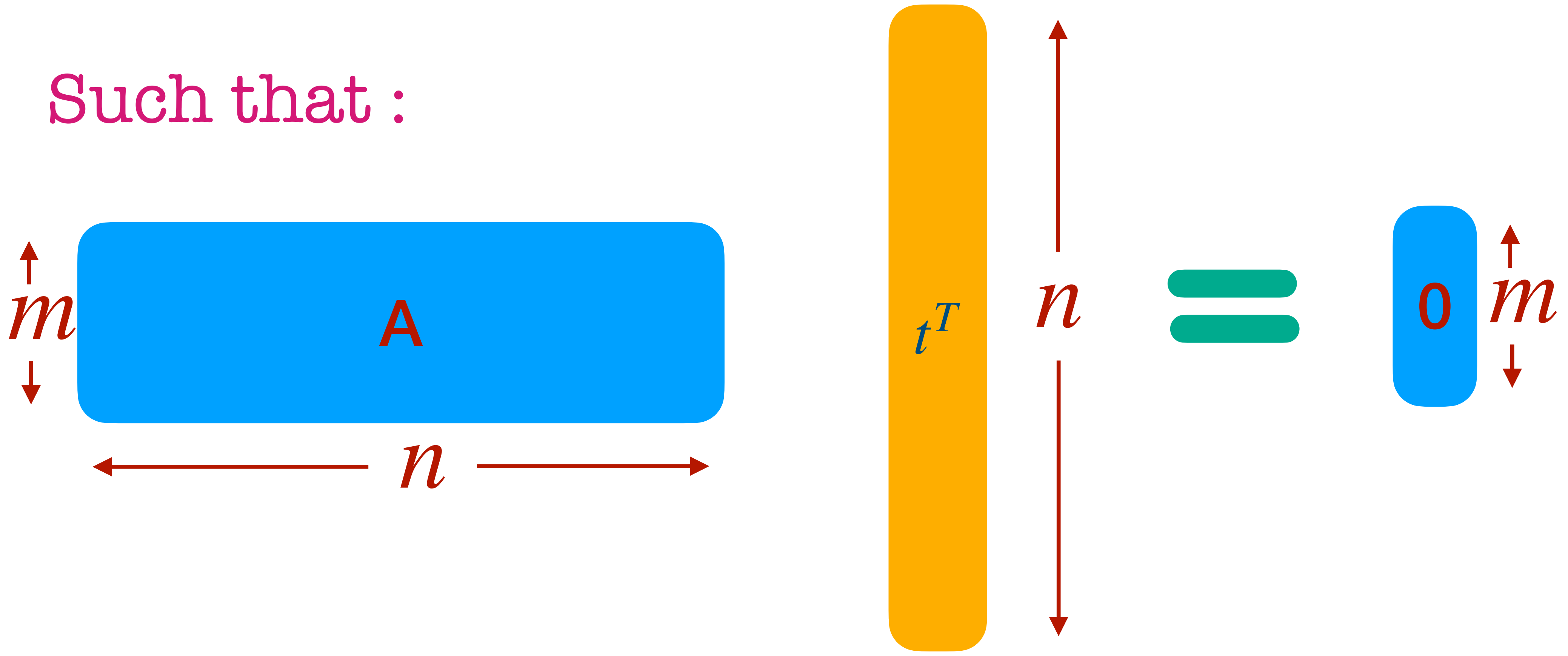


Secret key :

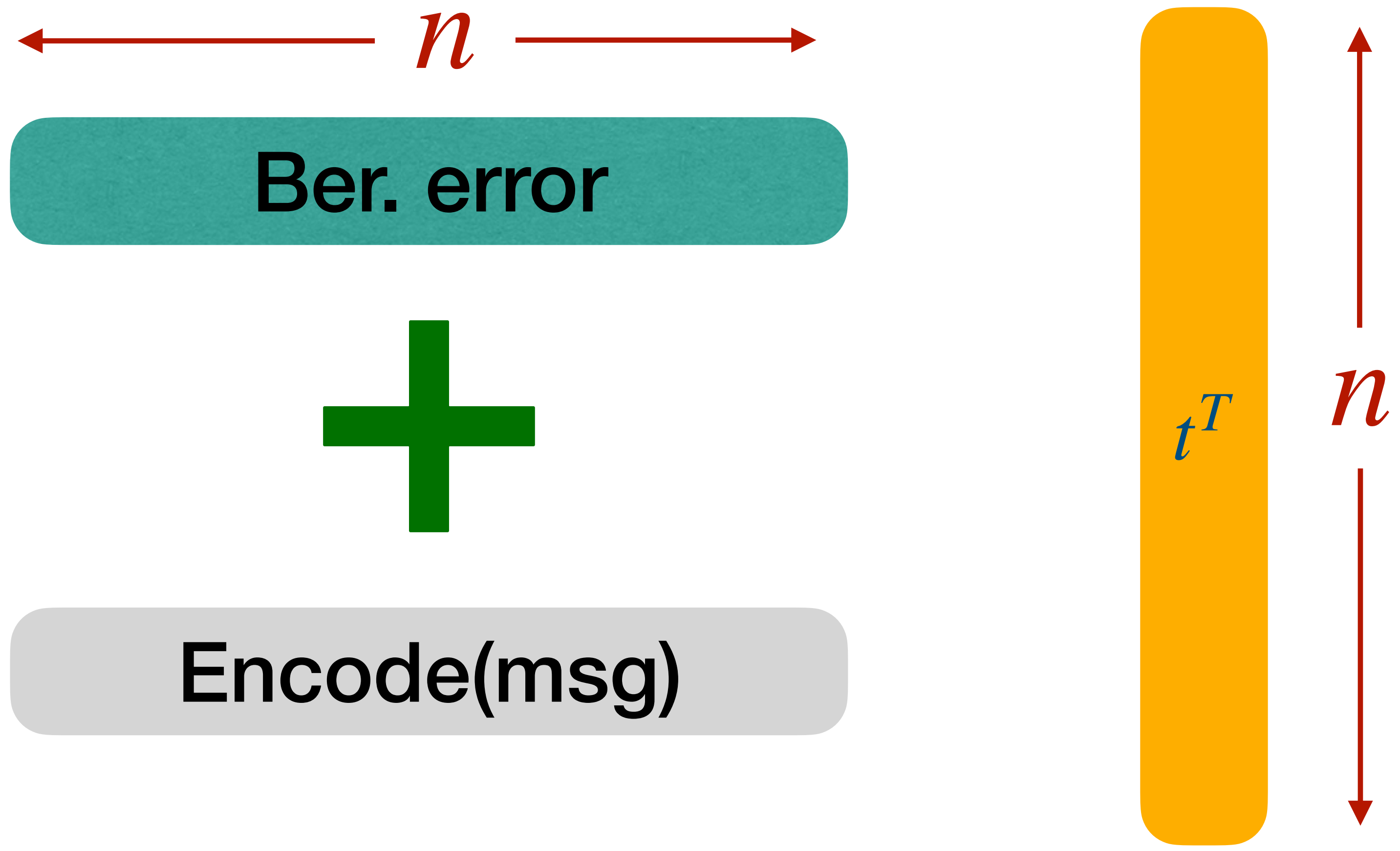


#1's = k

Such that :



Decryption :



$$m = \frac{k \log n}{\Delta}$$

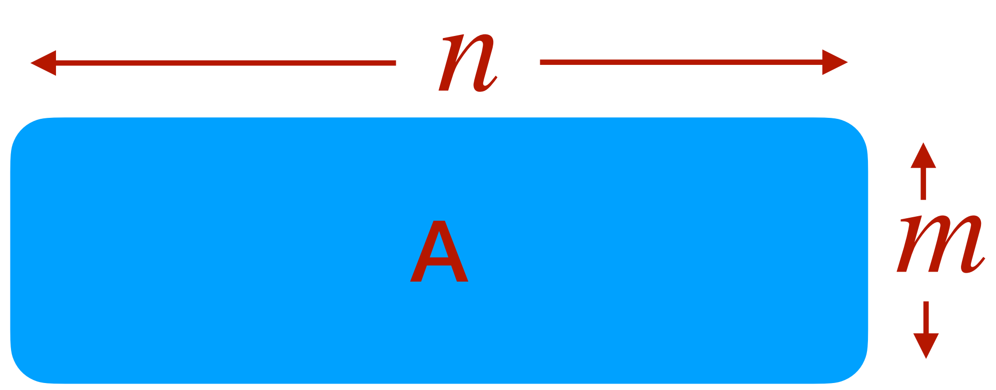
$$m \leq \frac{(\log n)^2}{\Delta}$$

Decryption is possible only if k <= log(n) given Ber. error is constant- more #1's, more error accumulation

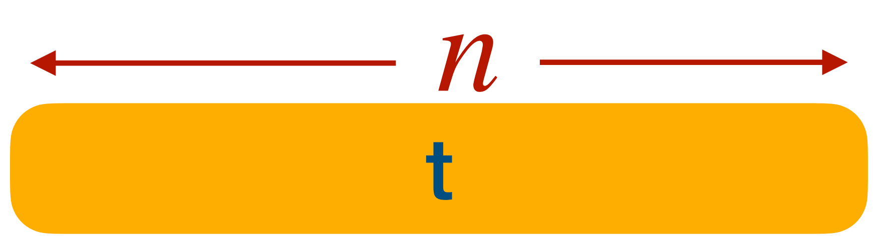
PKE

(All matrix and vector elements are in {0,1})

Public key :

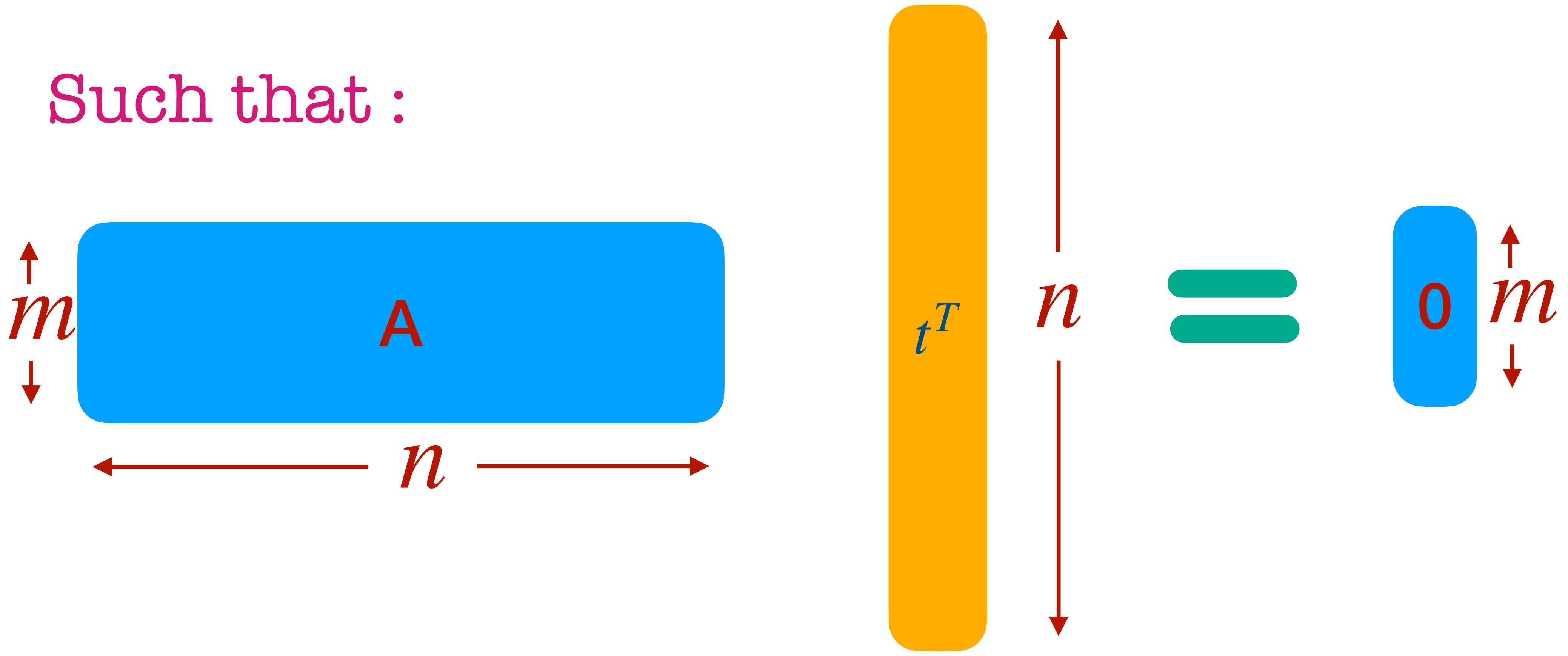


Secret key :

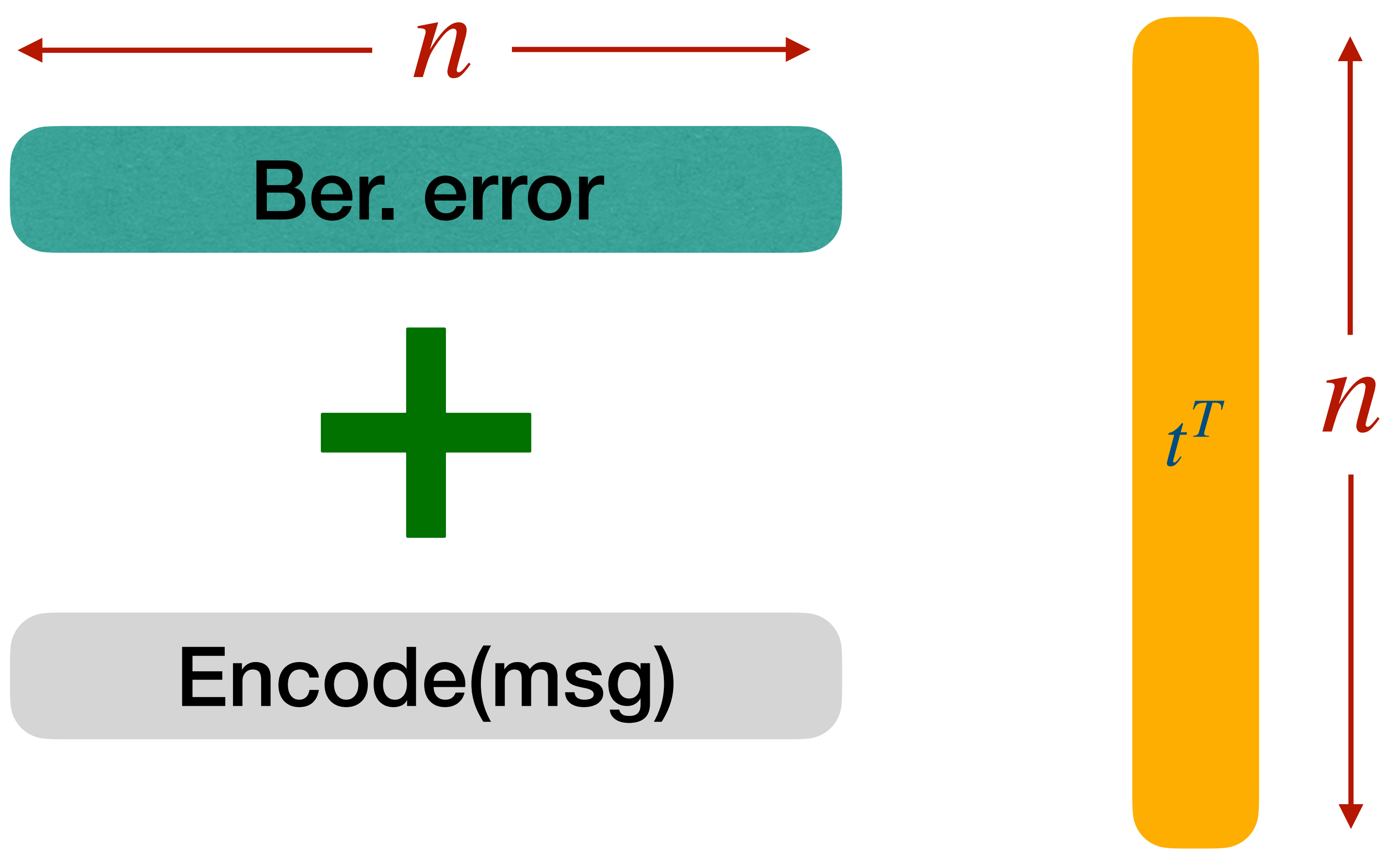


#1's = k

Such that :



Decryption :



$$m = \frac{k \log n}{\Delta}$$

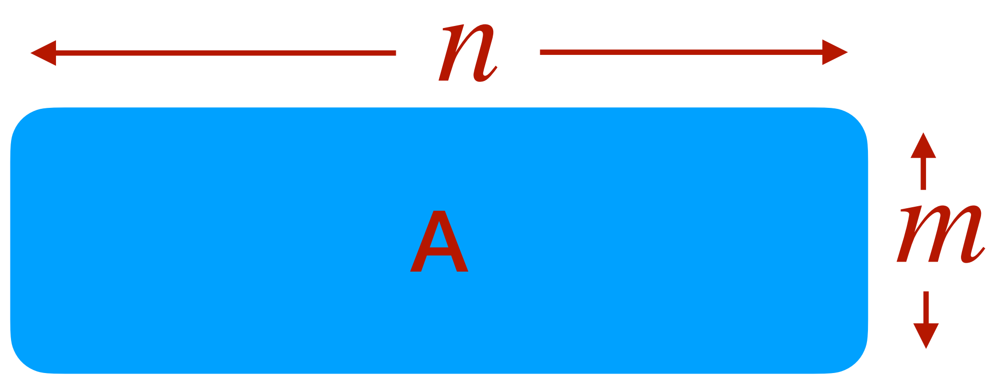
$$m \leq \frac{(\log n)^2}{\Delta}$$

Decryption is possible only if k <= log(n) given Ber. error is constant- more #1's, more error accumulation

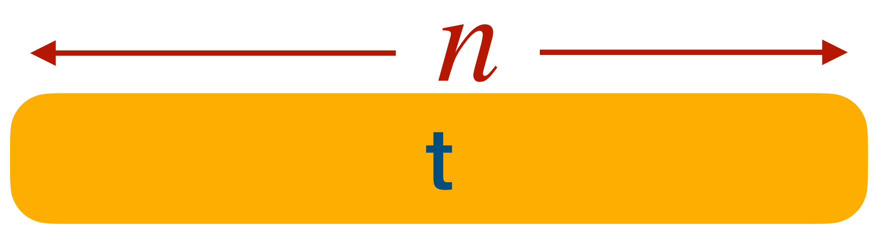
PKE

(All matrix and vector elements are in {0,1})

Public key :

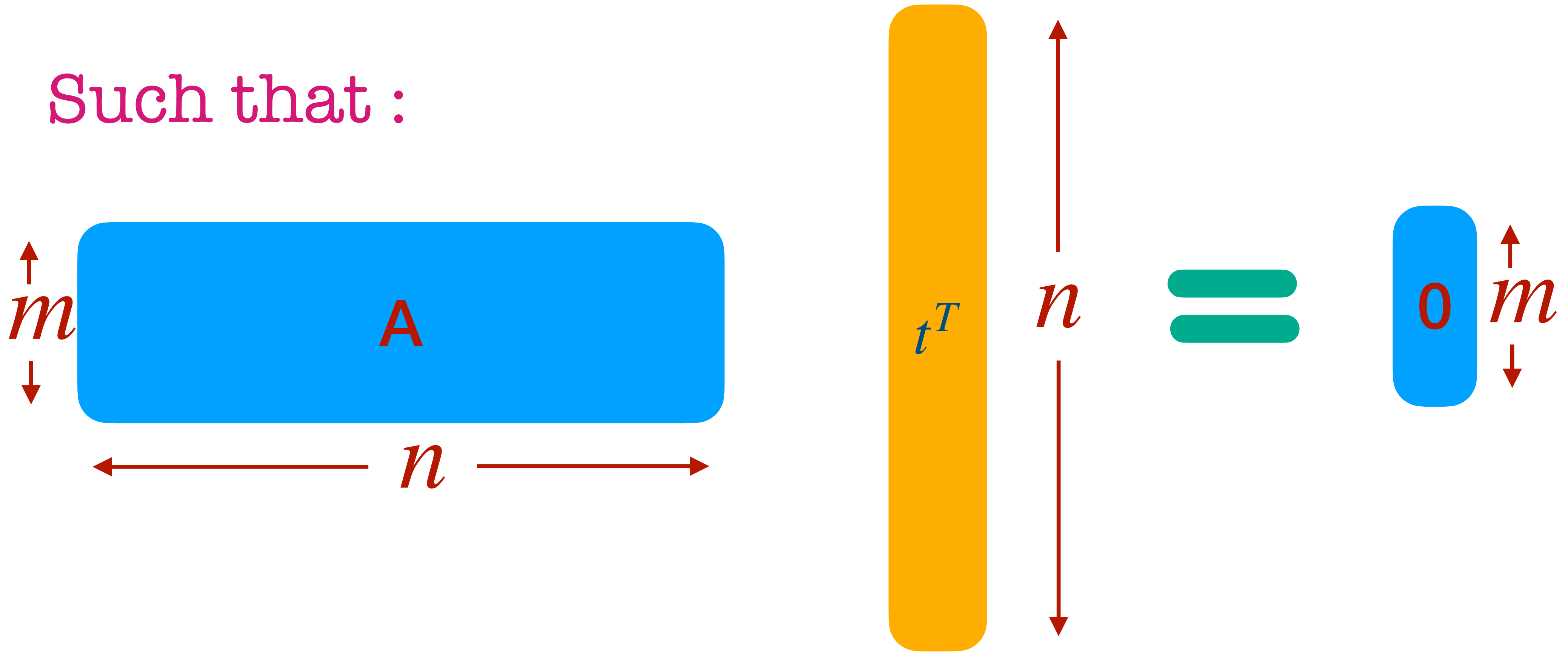


Secret key :

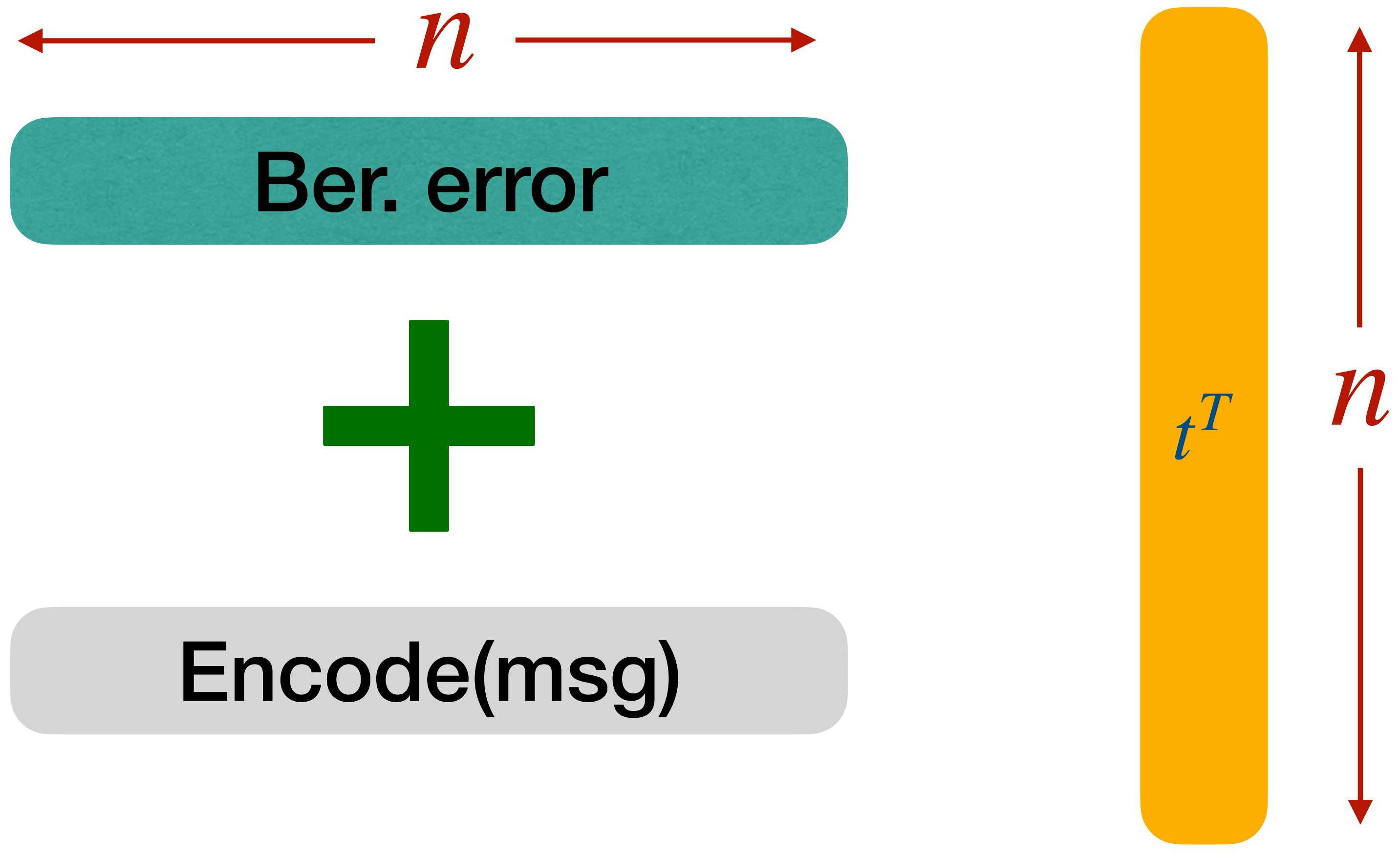


#1's = k

Such that :



Decryption :



Without k-XOR :

$\Delta \geq 1$

Planted $\approx_{stat}$ non-Planted

$m \leq (\log n)^2$

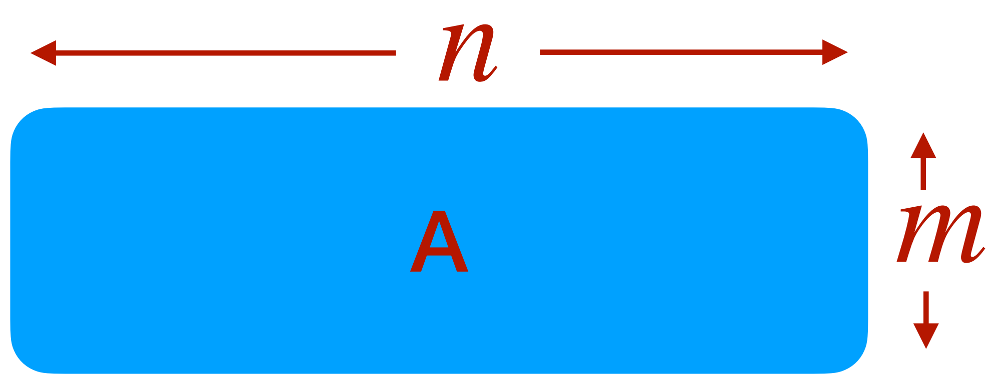
$$\boxed{m = \frac{k \log n}{\Delta} \qquad m \leq \frac{(\log n)^2}{\Delta}}$$

Decryption is possible only if k <= log(n) given Ber. error is constant- more #1's, more error accumulation

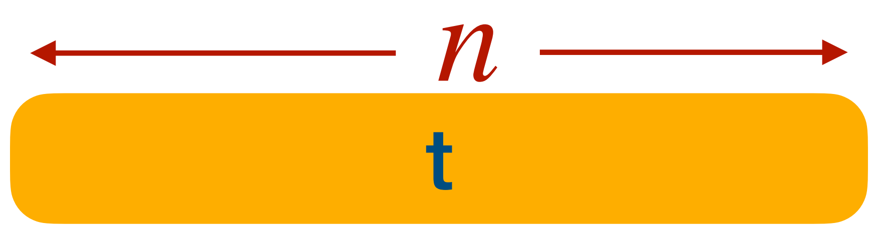
PKE

(All matrix and vector elements are in {0,1})

Public key :

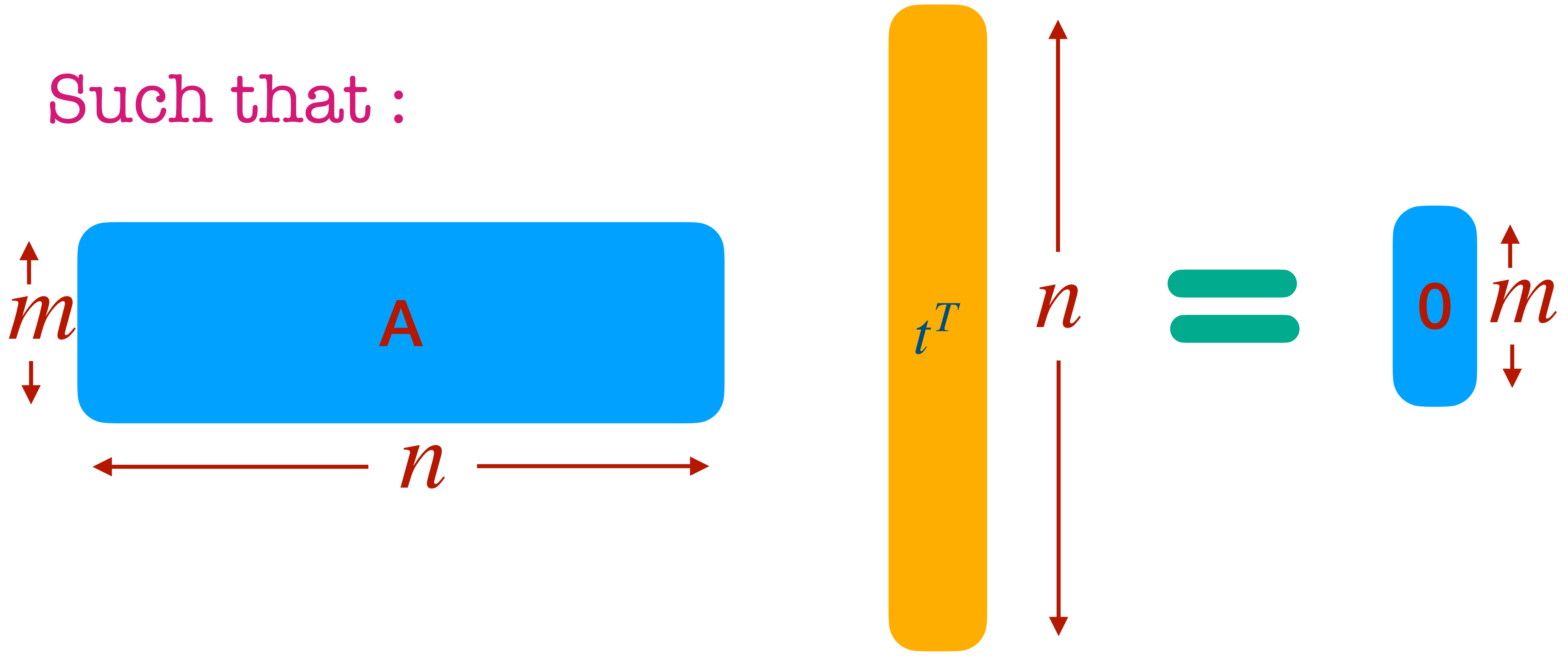


Secret key :

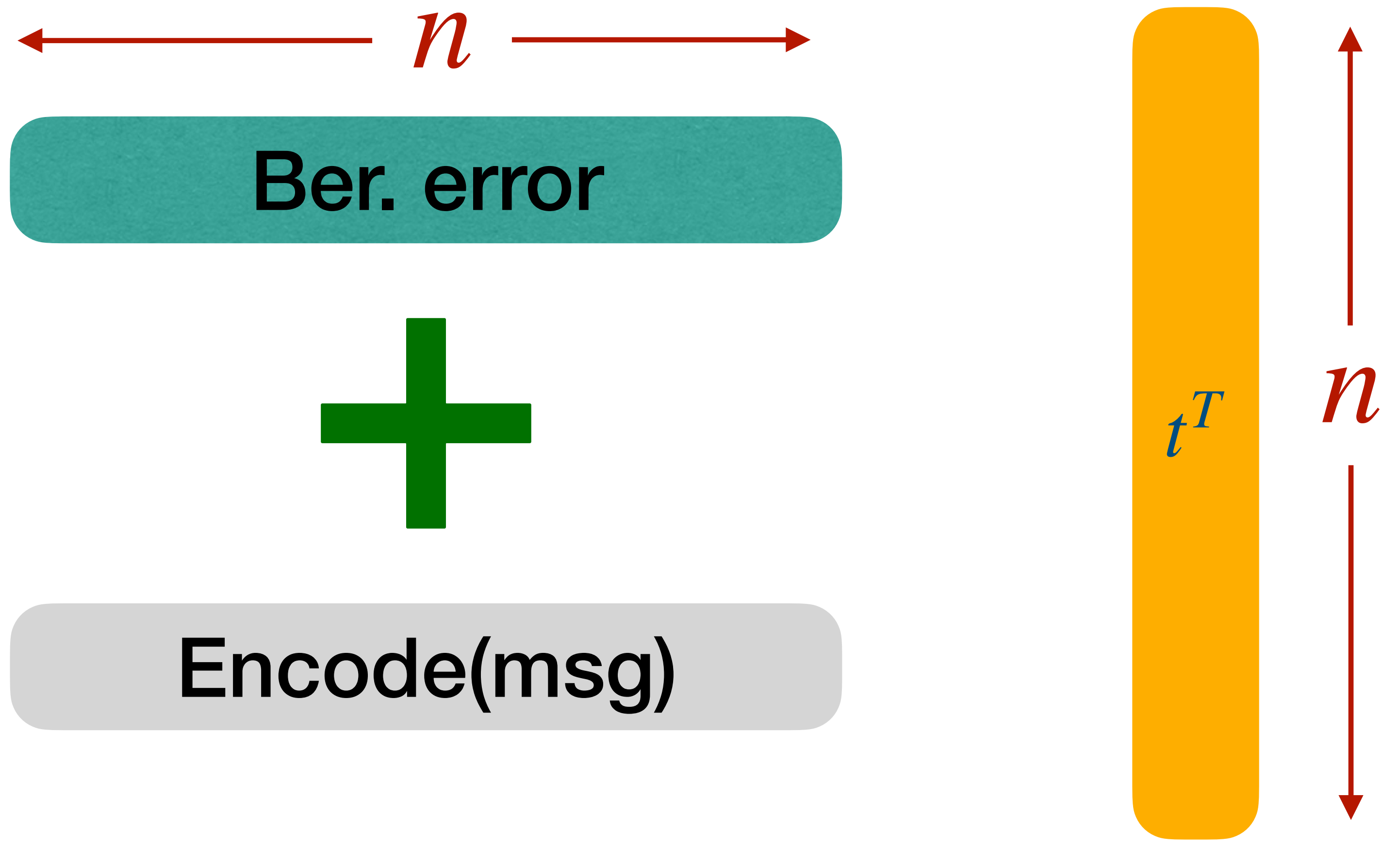


#1's = k

Such that :



Decryption :



Without k-XOR :

$$\Delta \geq 1$$

Planted $\approx_{stat}$ non-Planted

$$m \leq (\log n)^2$$

$$\boxed{m = \frac{k \log n}{\Delta} \qquad m \leq \frac{(\log n)^2}{\Delta}}$$

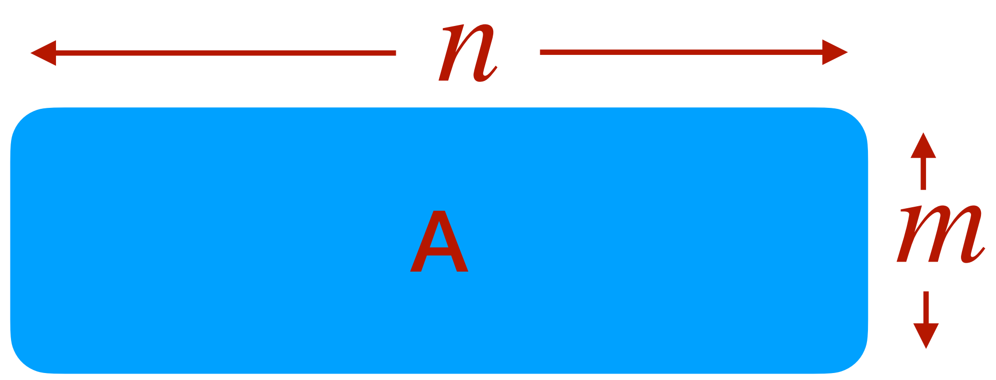
$\Longrightarrow$ $2^{m^{0.51}}$ Hardness assumed

Decryption is possible only if k <= log(n) given Ber. error is constant- more #1's, more error accumulation

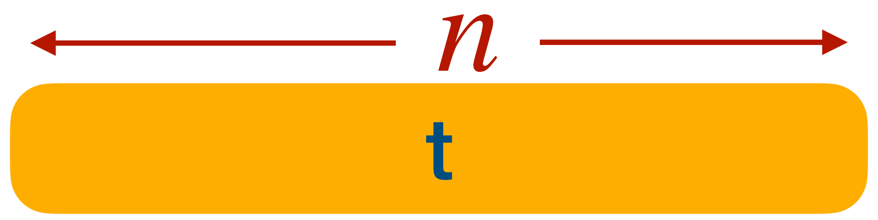
PKE

(All matrix and vector elements are in {0,1})

Public key :

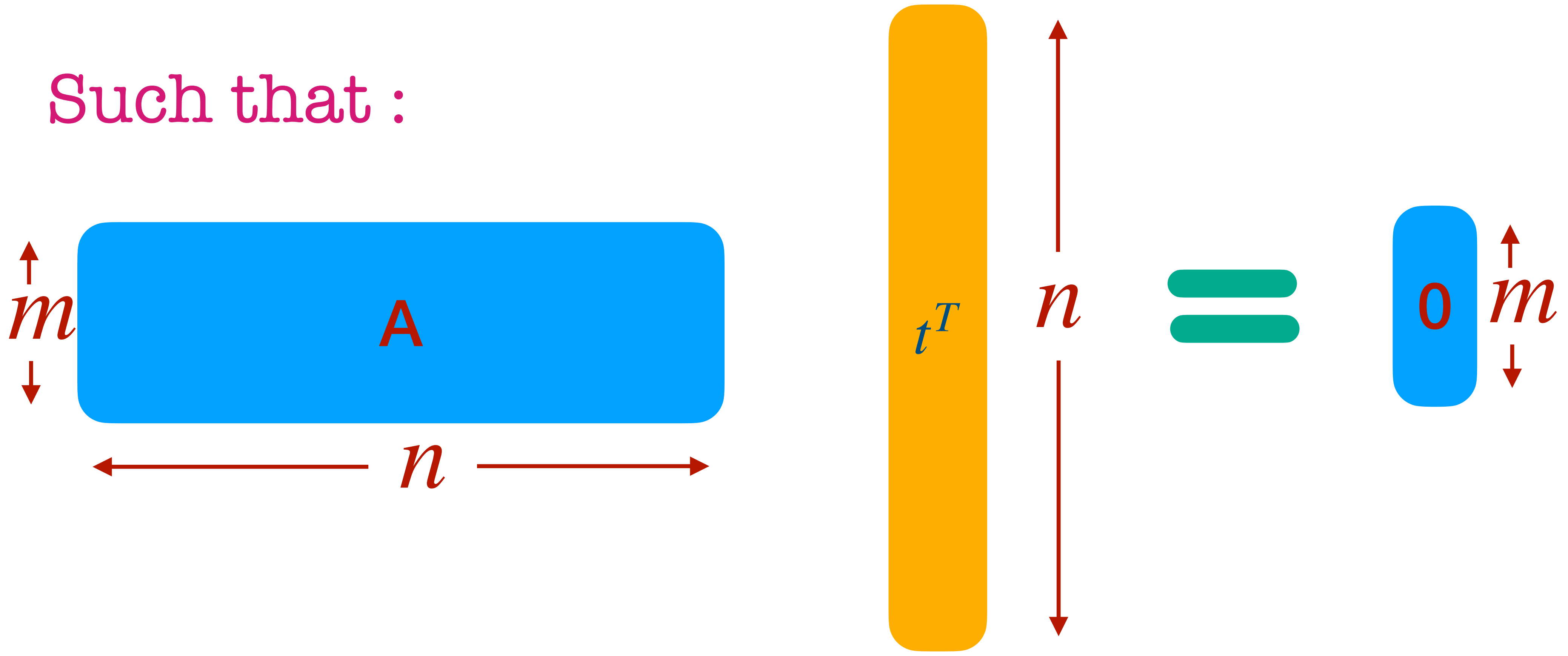


Secret key :



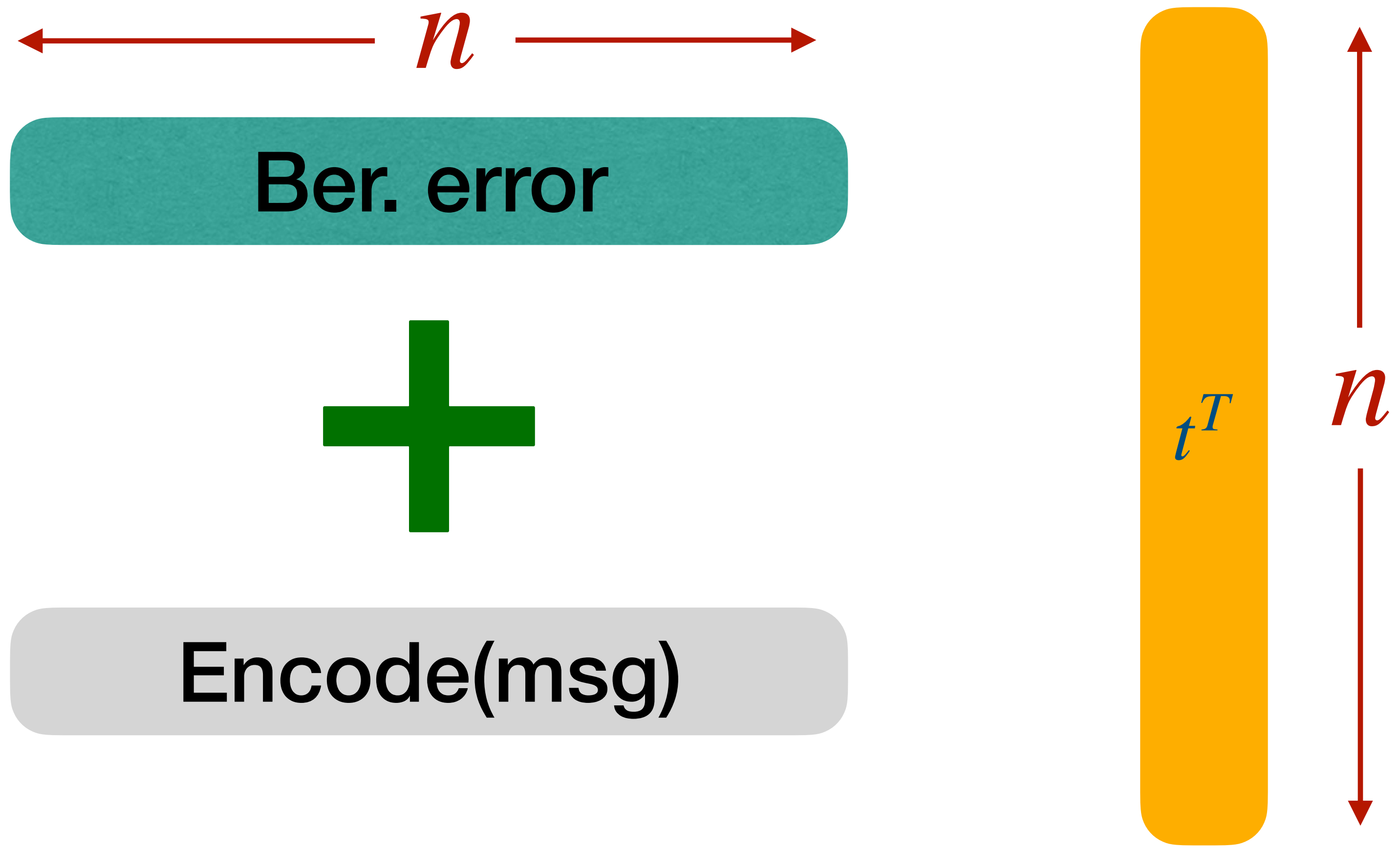
#1's = k

Such that :



With k-XOR :

Decryption :



Without k-XOR :

$\Delta \geq 1$

Planted $\approx_{stat}$ non-Planted

$m \leq (\log n)^2$

$$\boxed{\begin{array}{l} m = \dfrac{k \log n}{\Delta} \\[2ex] m \leq \dfrac{(\log n)^2}{\Delta} \end{array}}$$

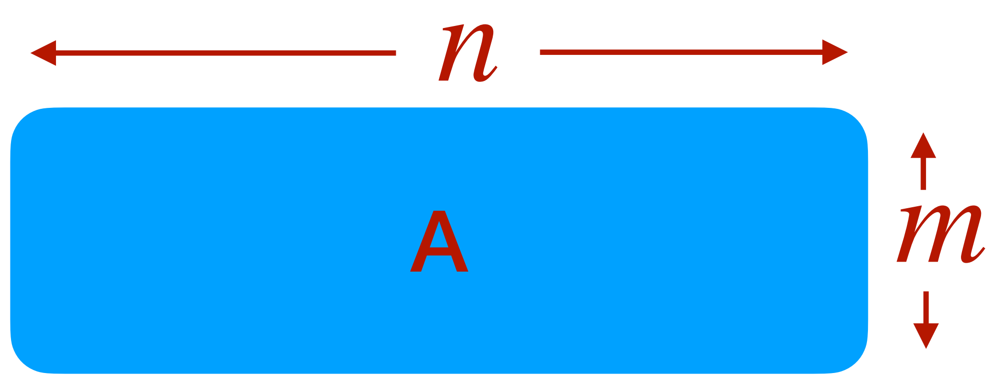
$\Longrightarrow$ $2^{m^{0.51}}$ Hardness assumed

Decryption is possible only if k <= log(n) given Ber. error is constant- more #1's, more error accumulation

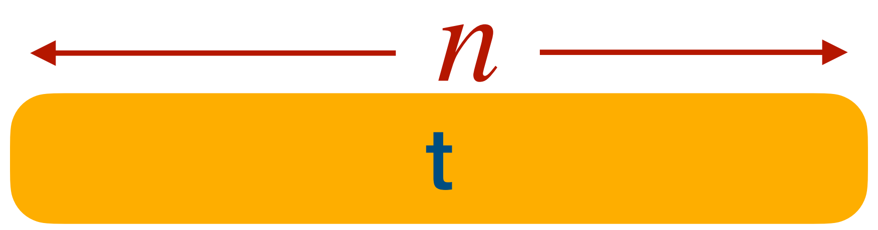
PKE

(All matrix and vector elements are in {0,1})

Public key :

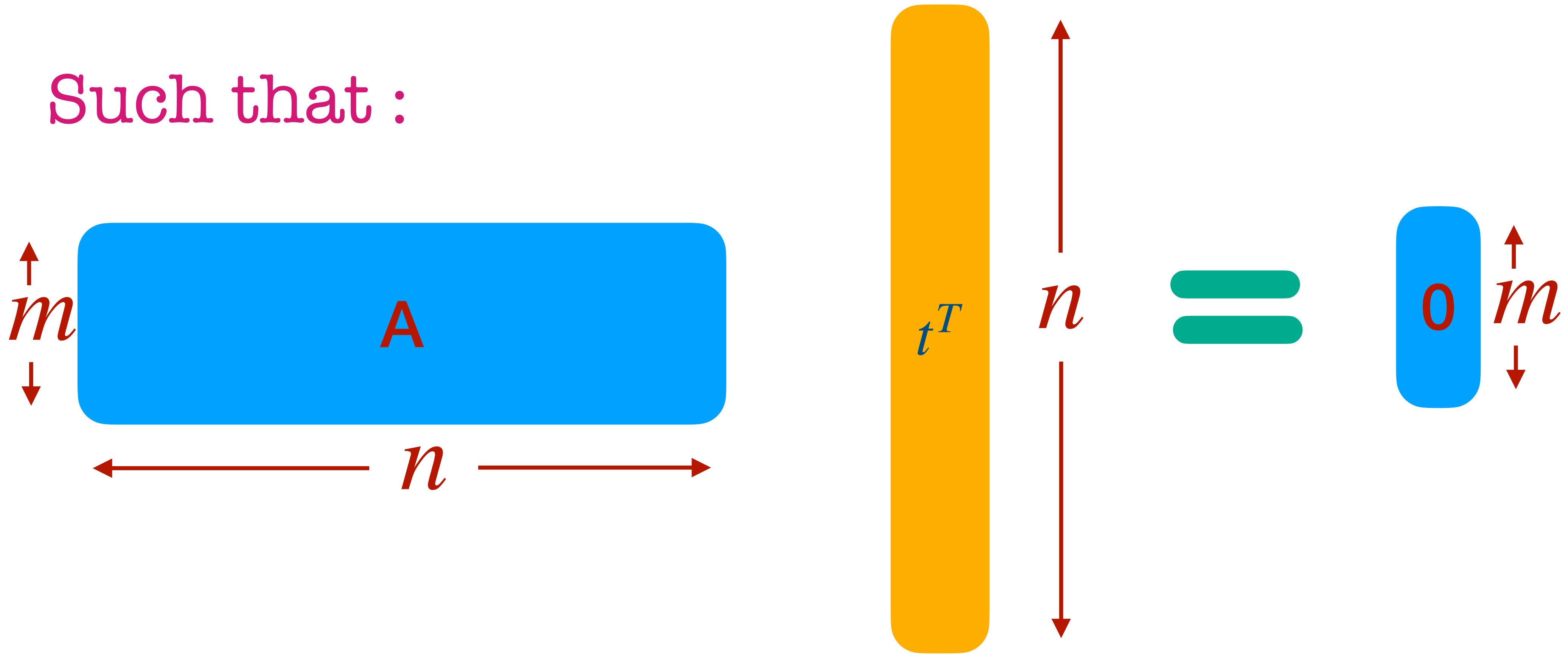


Secret key :

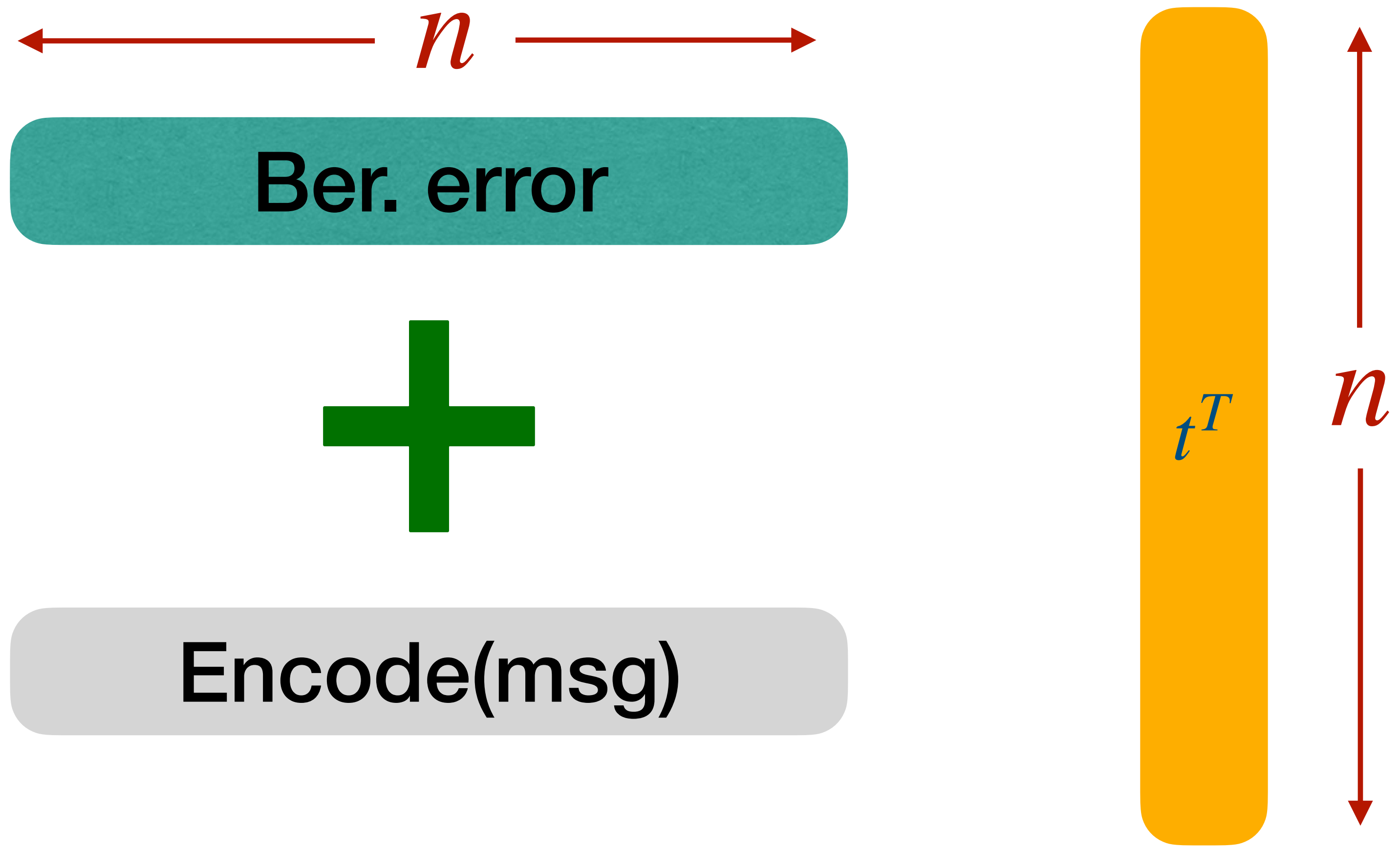


#1's = k

Such that :



Decryption :



Without k-XOR :

$\Delta \geq 1$

Planted $\approx_{stat}$ non-Planted

$m \leq (\log n)^2$

$\Longrightarrow$ $2^{m^{0.51}}$ Hardness assumed

$$m = \frac{k \log n}{\Delta}$$

$$m \leq \frac{(\log n)^2}{\Delta}$$

With k-XOR :

$\Delta < 1$

Planted $\approx_{comp}$ non-Planted

$$\Delta = \frac{1}{(\log n)^{\alpha}}$$

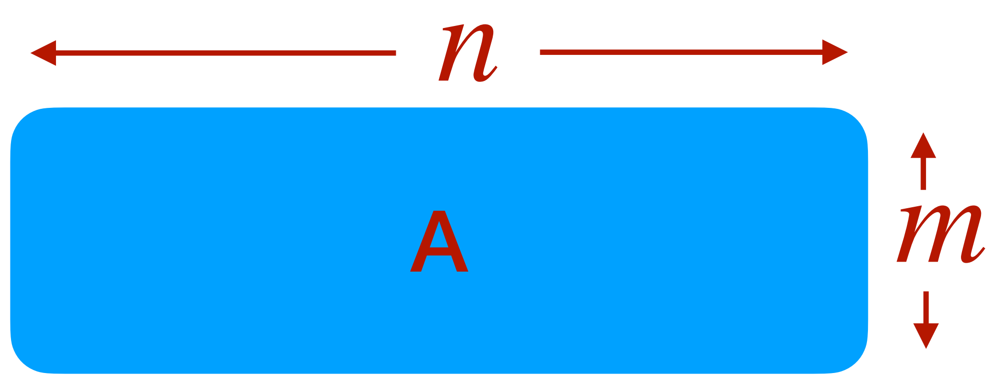
$m \leq (\log n)^{2+\alpha}$

Decryption is possible only if k <= log(n) given Ber. error is constant- more #1's, more error accumulation

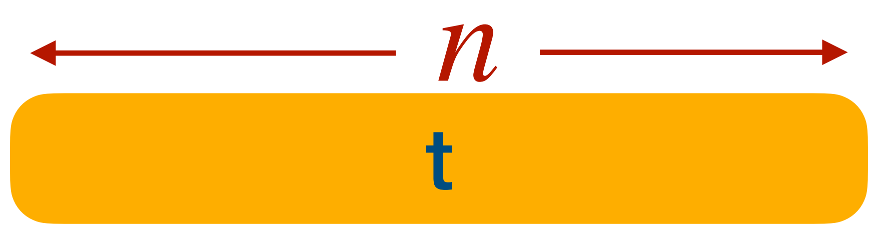
PKE

(All matrix and vector elements are in {0,1})

Public key :

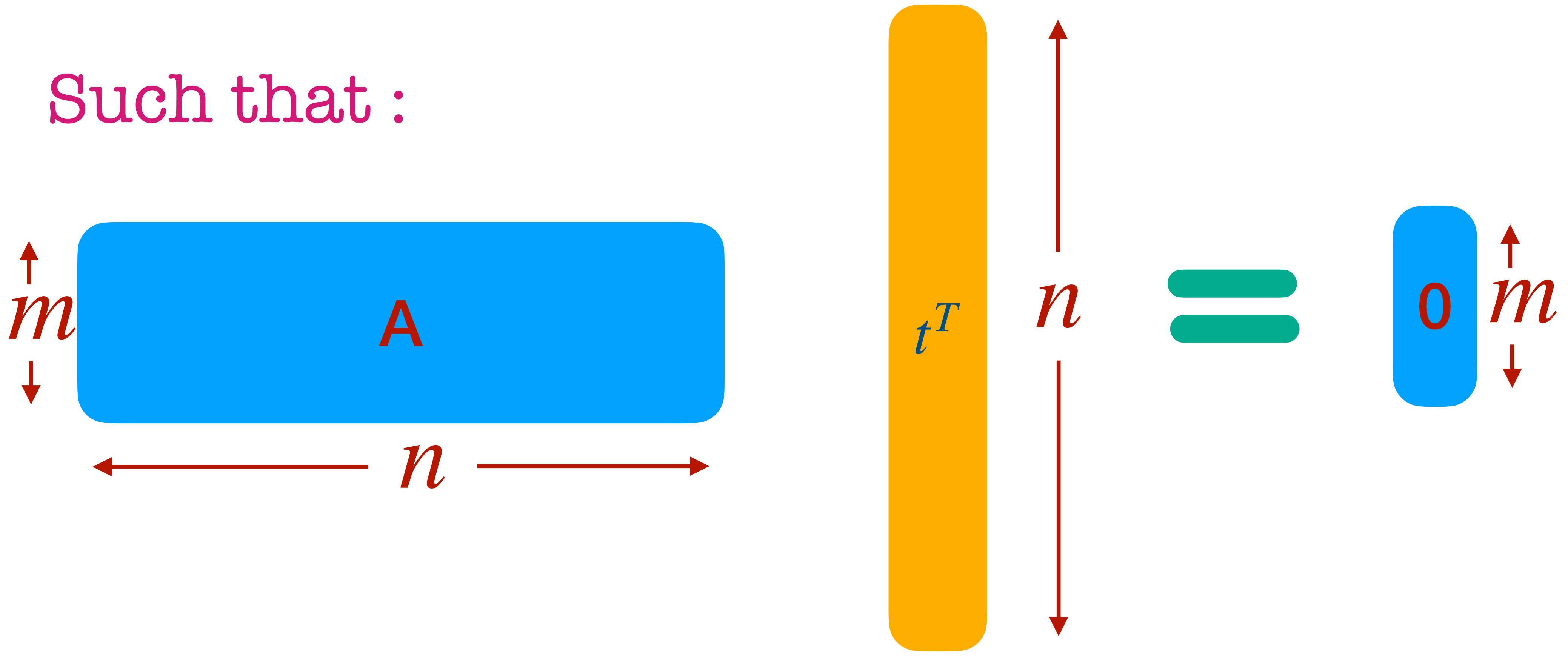


Secret key :

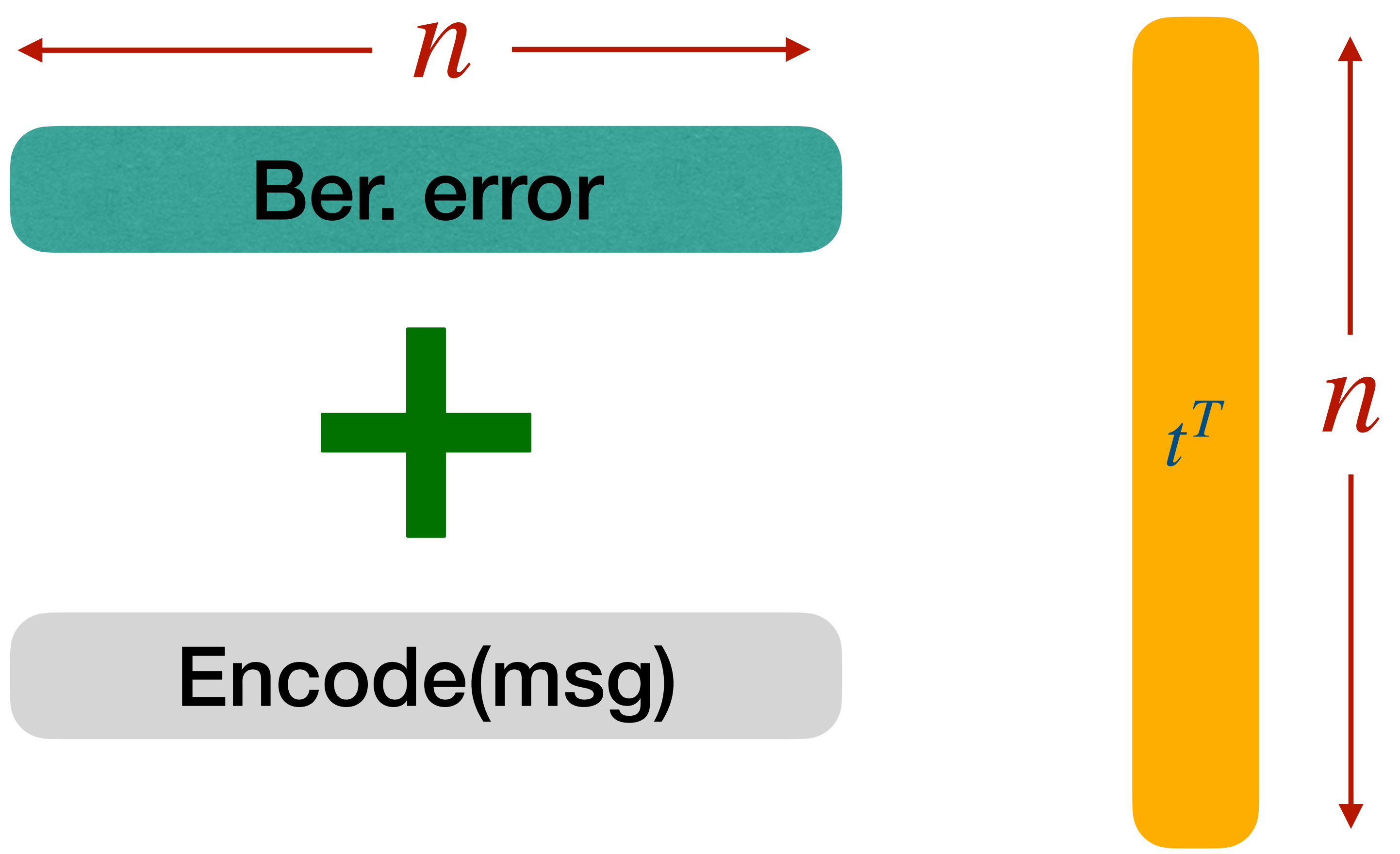


#1's = k

Such that :



Decryption :



Without k-XOR :

$\Delta \geq 1$

Planted $\approx_{stat}$ non-Planted

$m \leq (\log n)^2$

$\Longrightarrow$ $2^{m^{0.51}}$ Hardness assumed

$$\boxed{\begin{aligned} m &= \frac{k \log n}{\Delta} \\ m &\leq \frac{(\log n)^2}{\Delta} \end{aligned}}$$

With k-XOR :

$\Delta < 1$

Planted $\approx_{comp}$ non-Planted

$\Delta = \dfrac{1}{(\log n)^{\alpha}}$

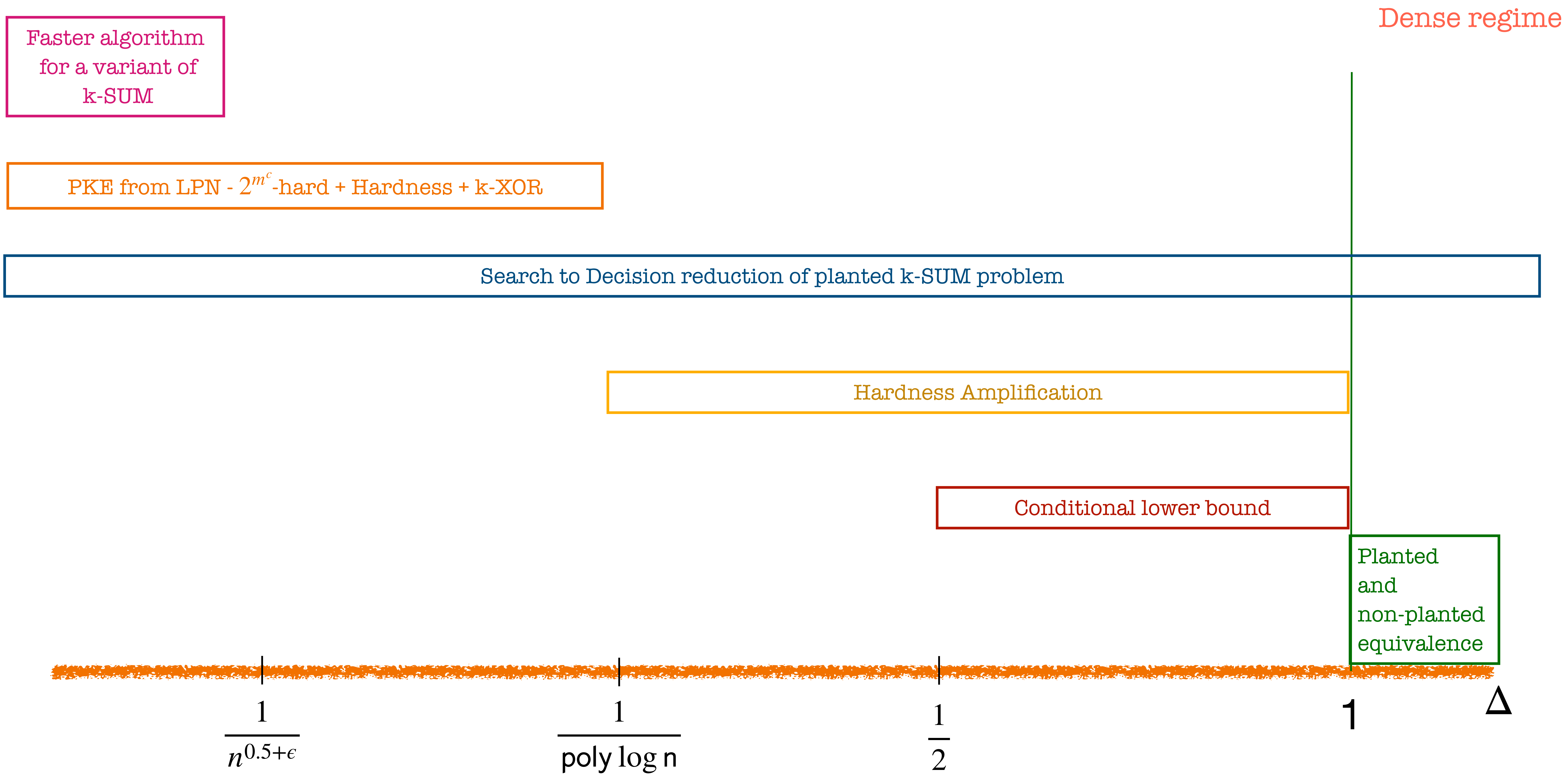
$m \leq (\log n)^{2+\alpha}$

$\Longrightarrow$ 2^{m^c} Hardness assumed

$: c \in (0,0.5)$

Decryption is possible only if k <= log(n) given Ber. error is constant- more #1's, more error accumulation

Summary of results mentioned

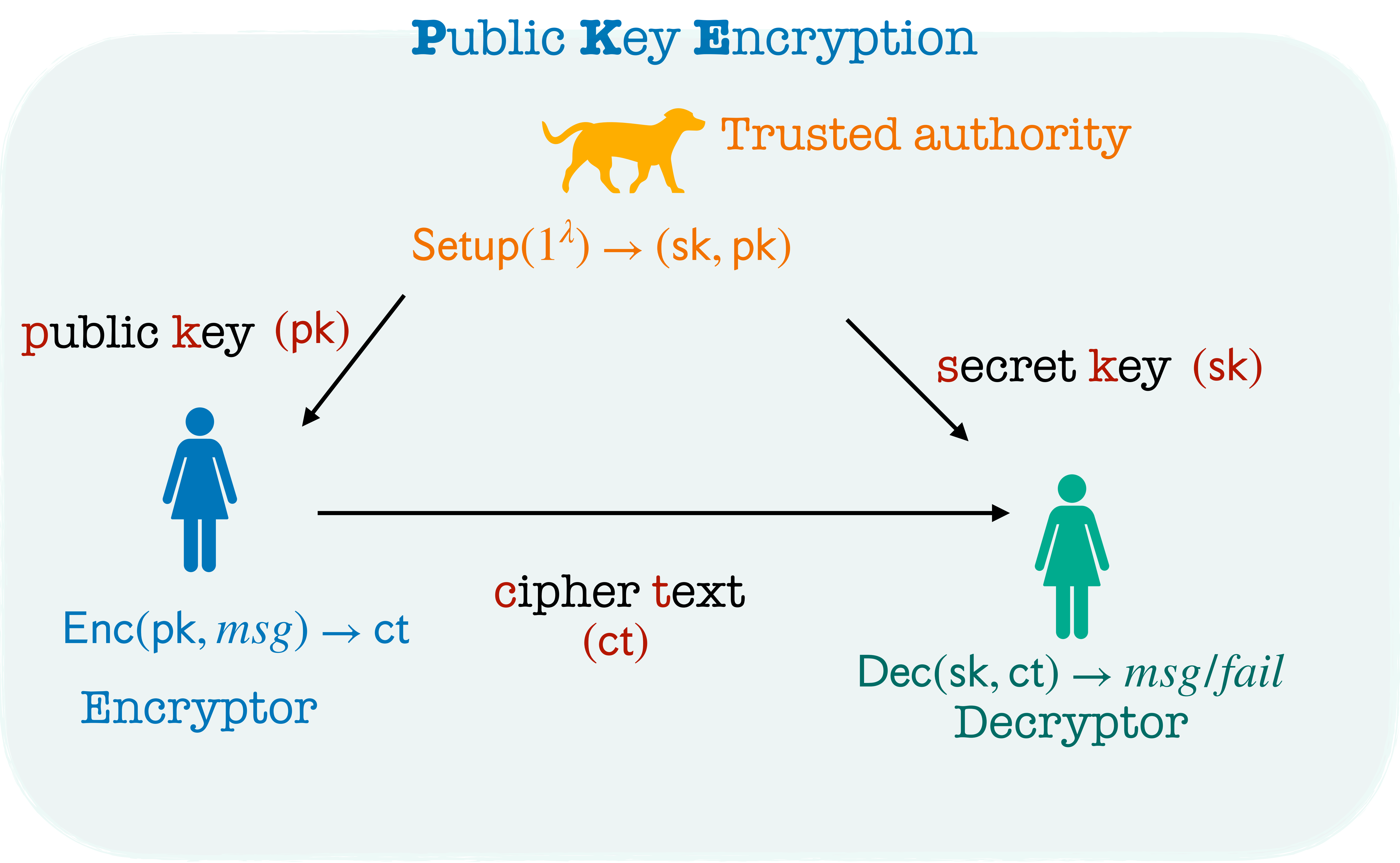


Conjecture : At $\Delta = 1$ best runtime of k-SUM algorithm is $n^{\lceil \frac{k}{2} \rceil - o(1)}$ [Pet15, LLW19, DKK21].

Questions ?

Thank you !

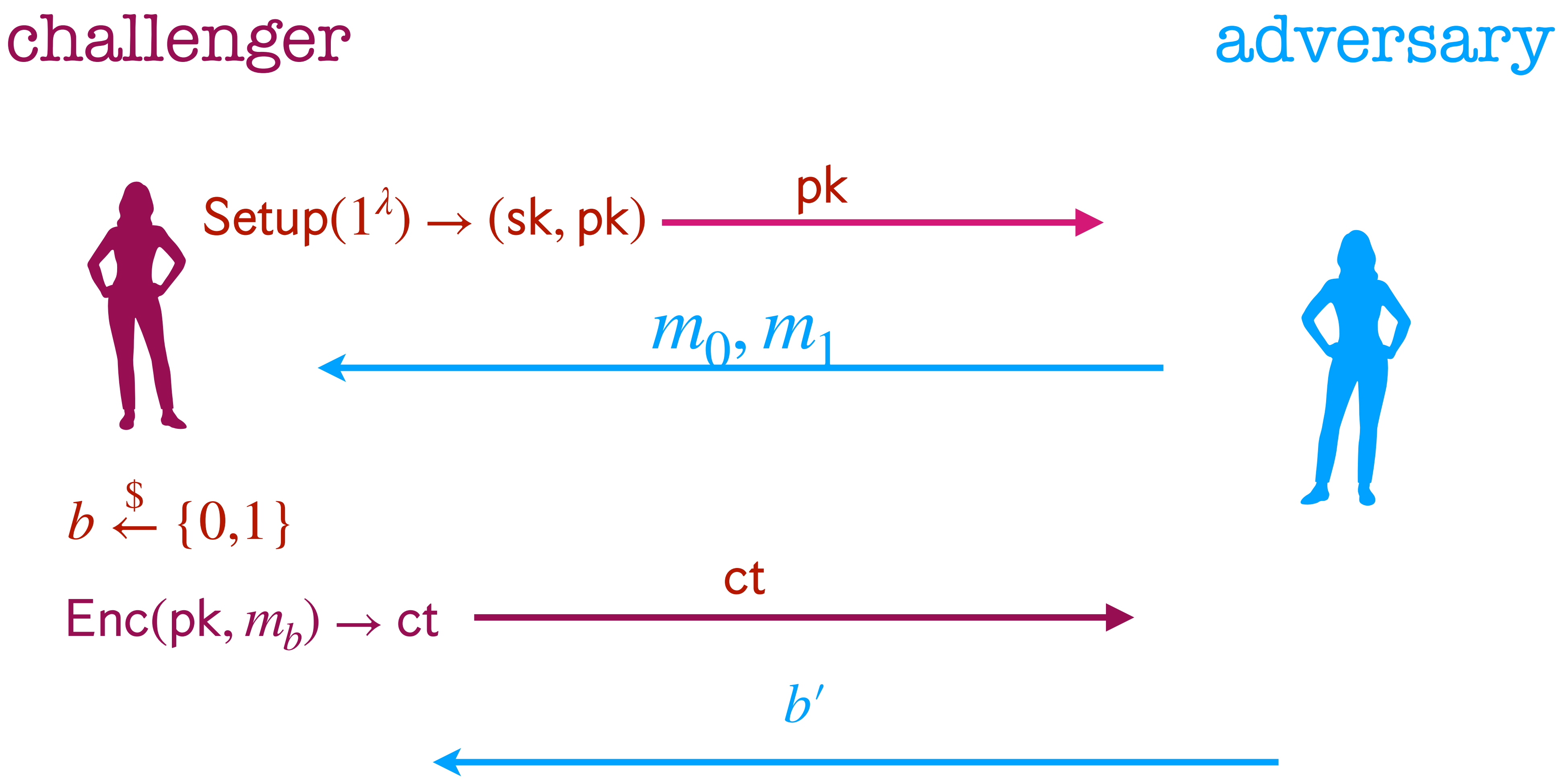
Extra slides - PKE



Correctness : $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) \rightarrow m$

Extra slides - PKE

Capturing the security of PKE



Adversary wins if b = b'

Conditional lower bound intuition - one approach

X

Instance

$\Delta = 1$

Convert

Such that solution is preserved in some form

Assume $\Delta' < 1$

X'

Instance

Δ'

a_1 a_2 a_3 . . . . . a_{n-2} a_{n-1} a_n

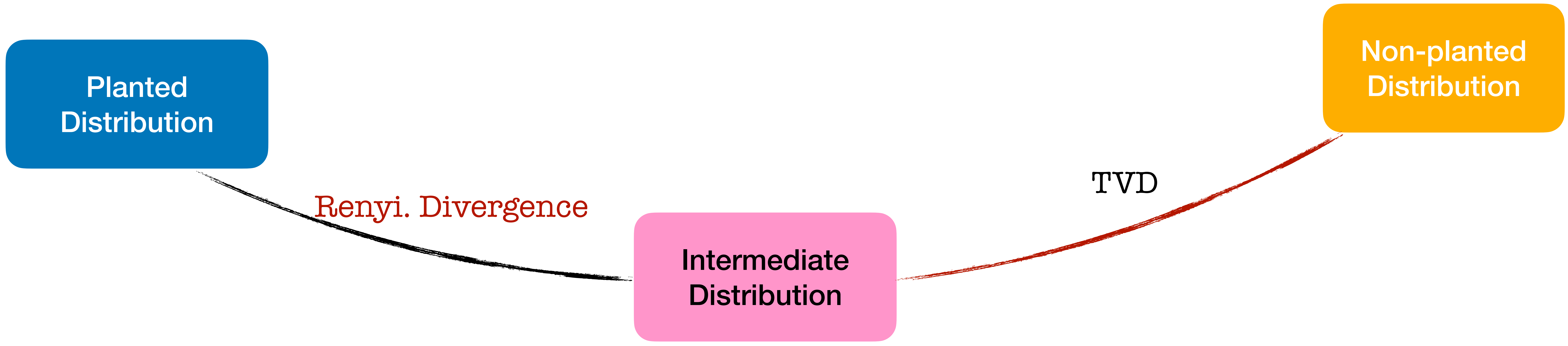
$$X' \subset X$$

$$\downarrow$$

$$|X'| = n^{\Delta'}$$

With high probability the solution gets copied too! Yay!

Planted and Non planted equivalence intuition



Find an optimal parameter value ℓ for a given value of n, k