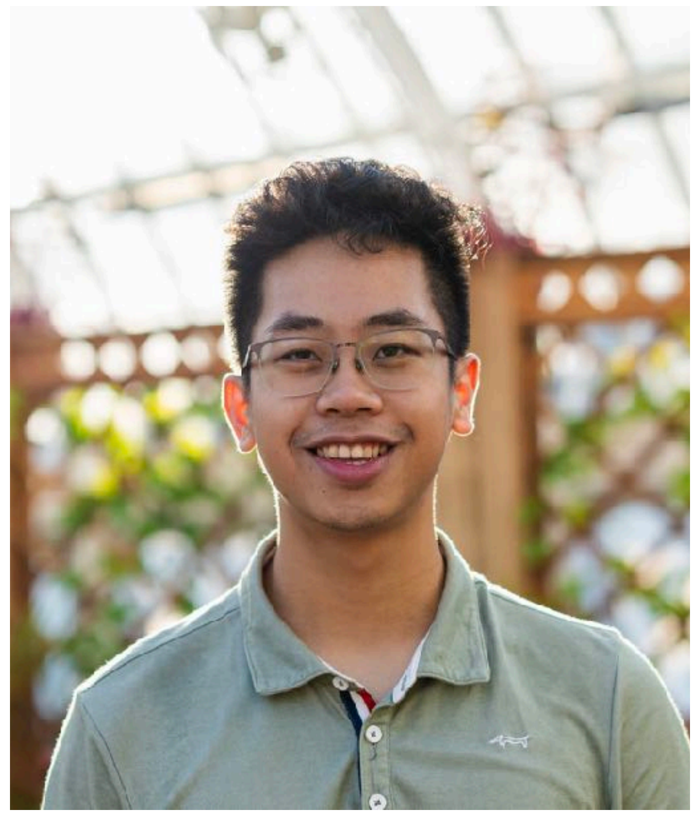
Lossy Cryptography from Code-Based Assumptions

Quang Dao

Aayush Jain



Crypto 2024

Post-Quantum Cryptography

Post-Quantum Cryptography

Exciting time for standardization & industry adoption!

CRYSTALS

Cryptographic Suite for Algebraic Lattices

FALCON

SPHINCS$^+$

Stateless hash-based signatures

Defending against future threats: Cloudflare goes post-quantum

10/03/2022

Quantum Resistance and the Signal Protocol

ehrenkret on 19 Sep 2023

February 21, 2024

iMessage with PQ3: The new state of the art in quantum-secure messaging at scale

How Google is preparing for a post-quantum world

July 6, 2022

Post-Quantum Cryptography

Exciting time for standardization & industry adoption!

CRYSTALS
Cryptographic Suite for Algebraic Lattices

FALCON

SPHINCS+
Stateless hash-based signatures

Defending against future threats: Cloudflare goes post-quantum
10/03/2022

Quantum Resistance and the Signal Protocol
ehrenkret on 19 Sep 2023

February 21, 2024
iMessage with PQ3: The new state of the art in quantum-secure messaging at scale

How Google is preparing for a post-quantum world
July 6, 2022

Need for advanced cryptography: Threshold Sign's, FHE, SNARGs, quantum, etc.

Post-Quantum Cryptography

Exciting time for standardization & industry adoption!

CRYSTALS

Cryptographic Suite for Algebraic Lattices

FALCON

SPHINCS+

Stateless hash-based signatures

Defending against future threats: Cloudflare goes post-quantum

10/03/2022

Quantum Resistance and the Signal Protocol

ehrenkret on 19 Sep 2023

February 21, 2024

iMessage with PQ3: The new state of the art in quantum-secure messaging at scale

How Google is preparing for a post-quantum world

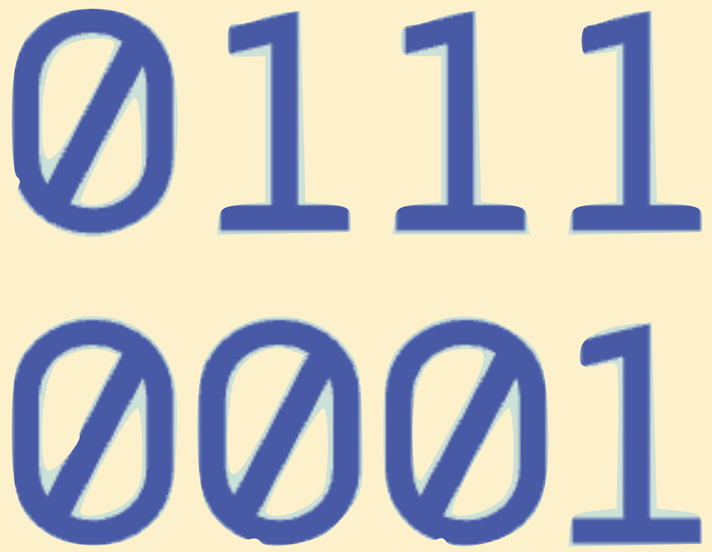
July 6, 2022

Need for advanced cryptography: Threshold Sign's, FHE, SNARGs, quantum, etc.

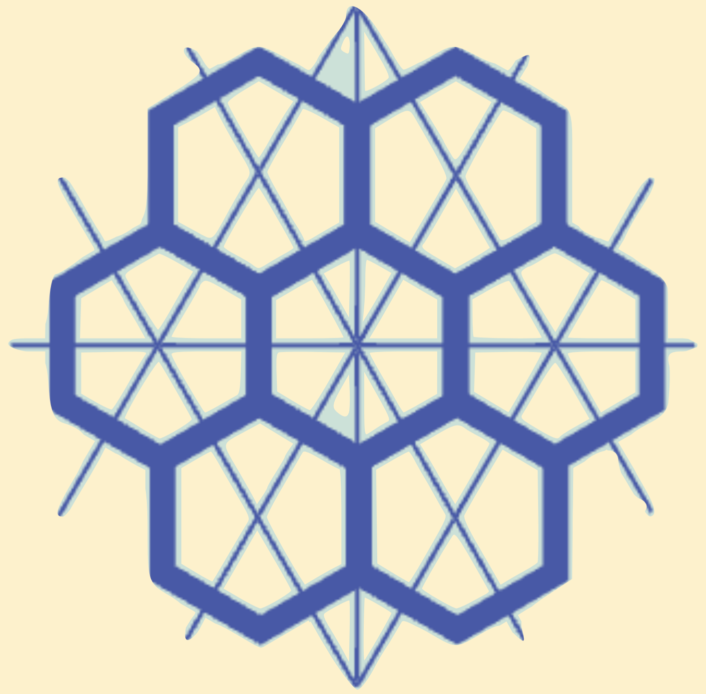
Problem: Not enough ways to construct post-quantum primitives!

The Current State of Post-Quantum Crypto

Code-based

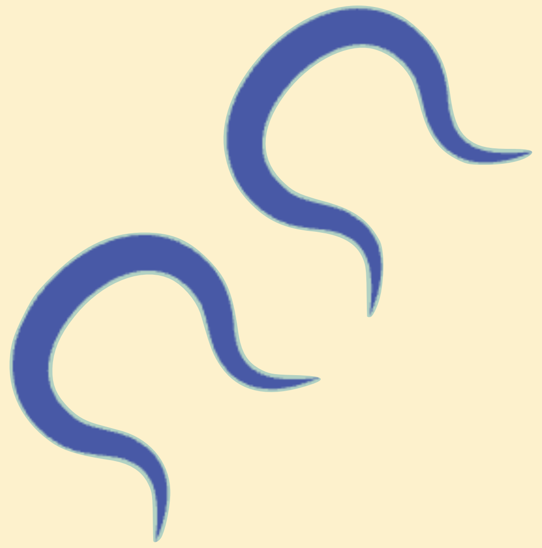


Lattice-based



Multivariate-based

Isogeny-based

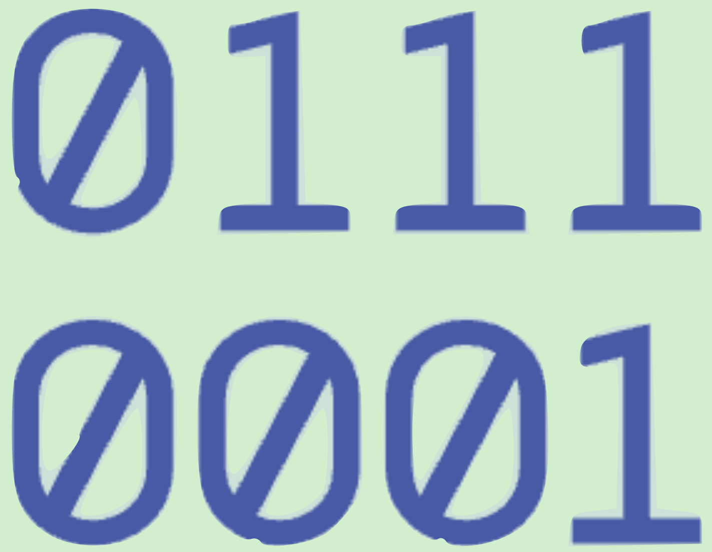


Images from Quanta Magazine

The Current State of Post-Quantum Crypto

One-way Functions

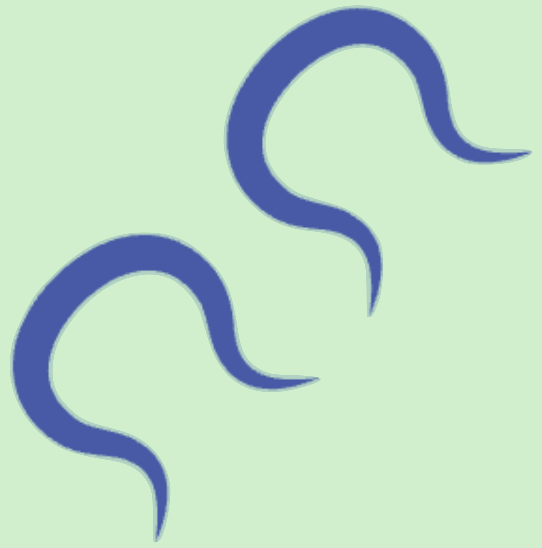
Code-based



Lattice-based

Multivariate-based

Isogeny-based

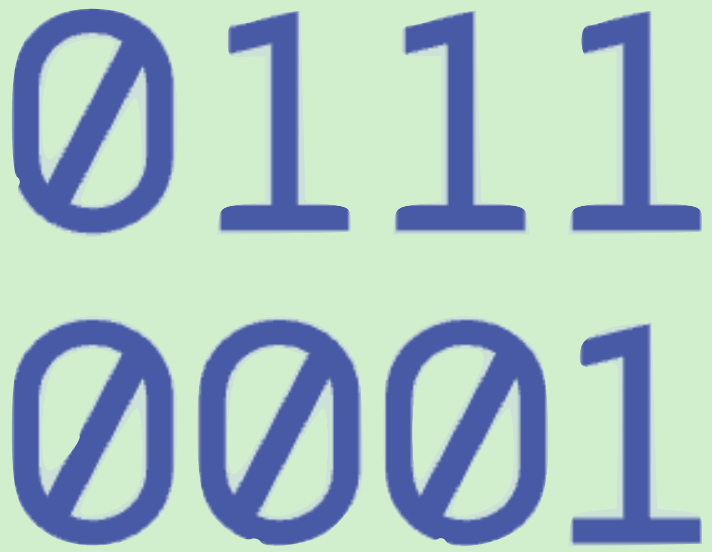


Images from Quanta Magazine

The Current State of Post-Quantum Crypto

Public-Key Encryption

Code-based

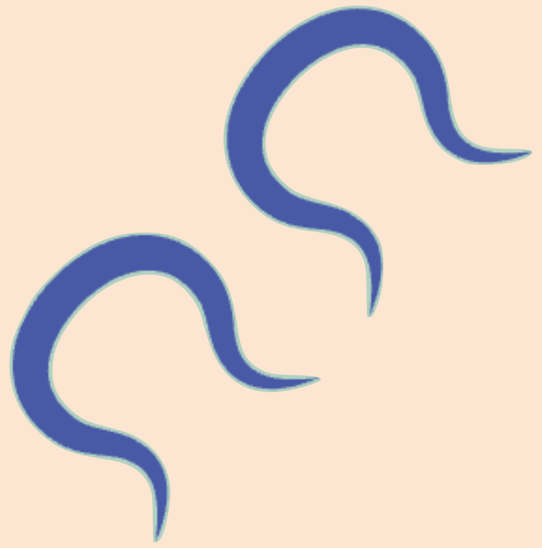


Lattice-based

Multivariate-based

many proposals, most are broken

Isogeny-based



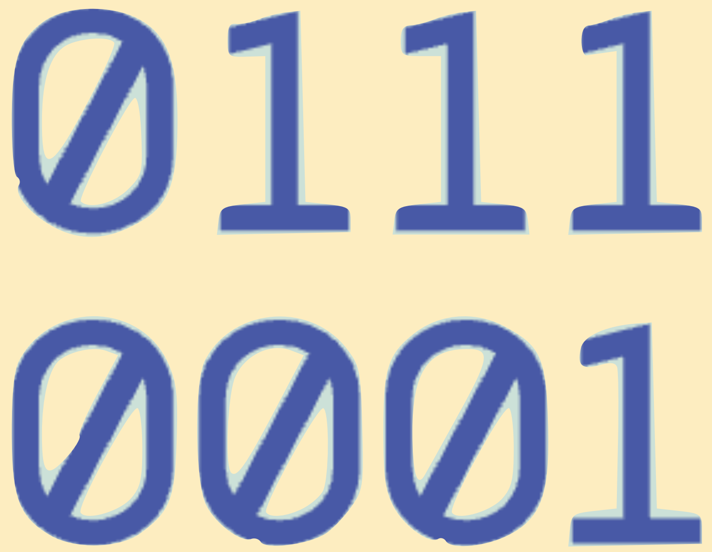
SIDH broken, but still CSIDH, new proposals, etc.

Images from Quanta Magazine

The Current State of Post-Quantum Crypto

Identity-Based Encryption

Code-based

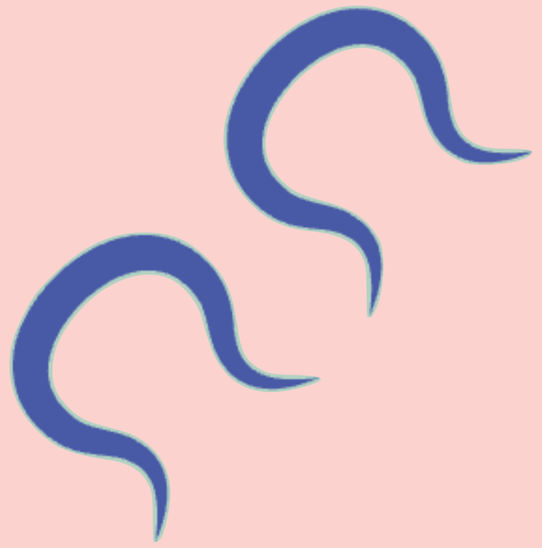


* Only quasi-polynomially secure

Lattice-based

Multivariate-based

Isogeny-based

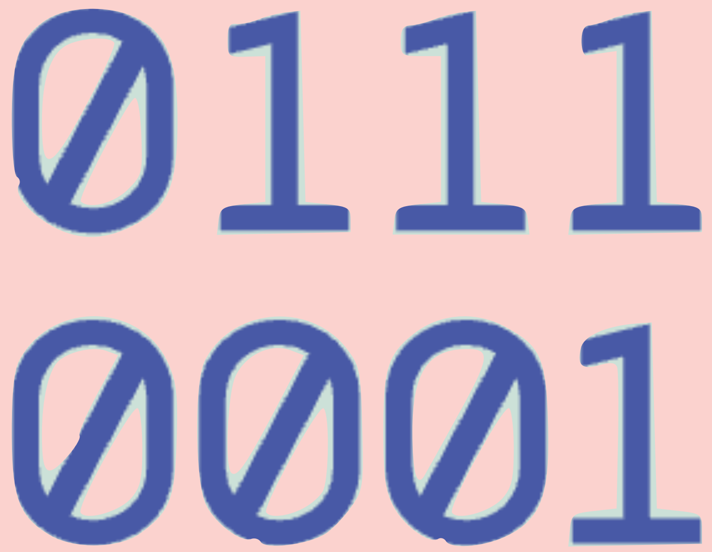


Images from Quanta Magazine

The Current State of Post-Quantum Crypto

Proof Systems (BARGs, SNARGs), Advanced Encryption (ABE, FE, FHE), …

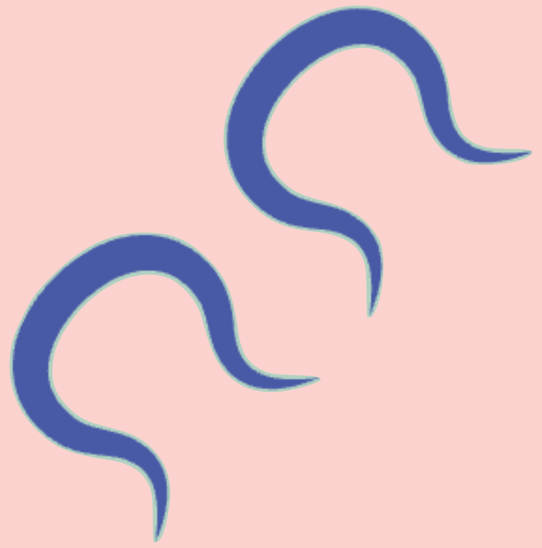
Code-based



Lattice-based

Multivariate-based

Isogeny-based

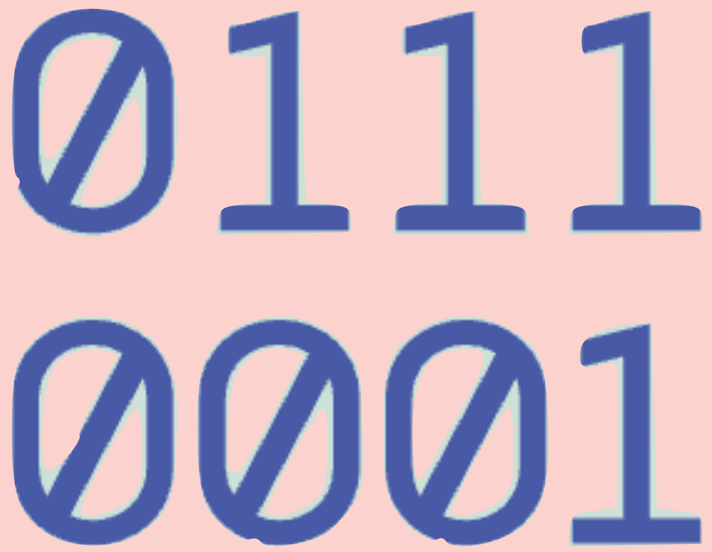


Images from Quanta Magazine

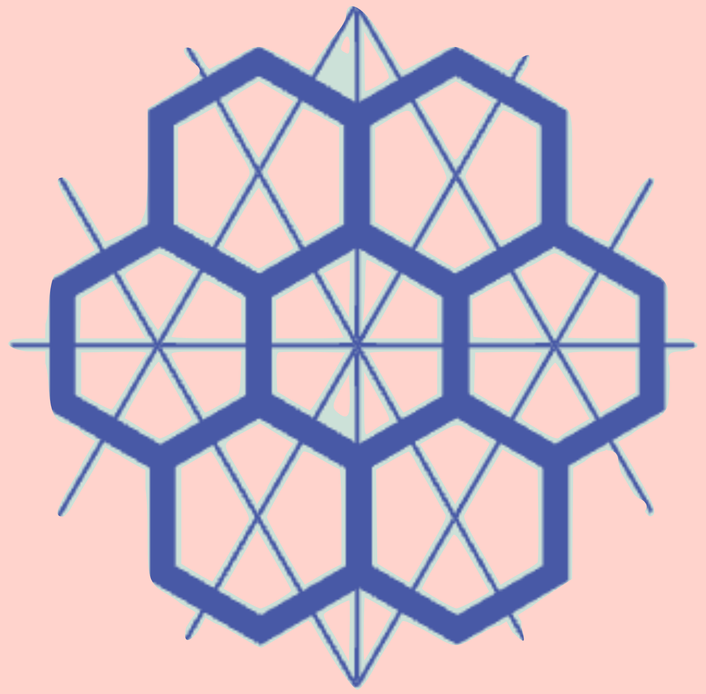
The Current State of Post-Quantum Crypto

Program Obfuscation

Code-based

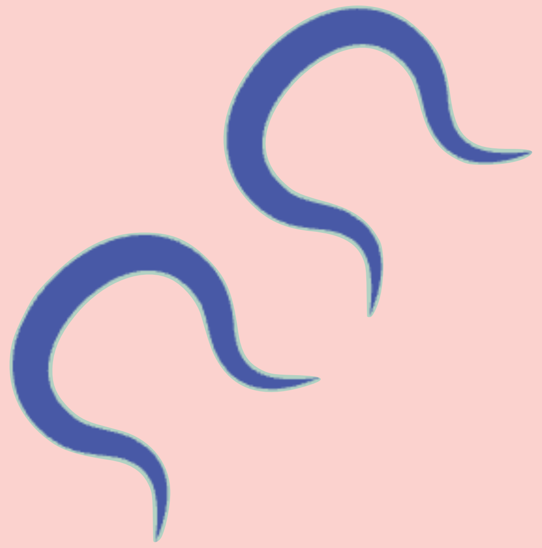


Lattice-based



Multivariate-based

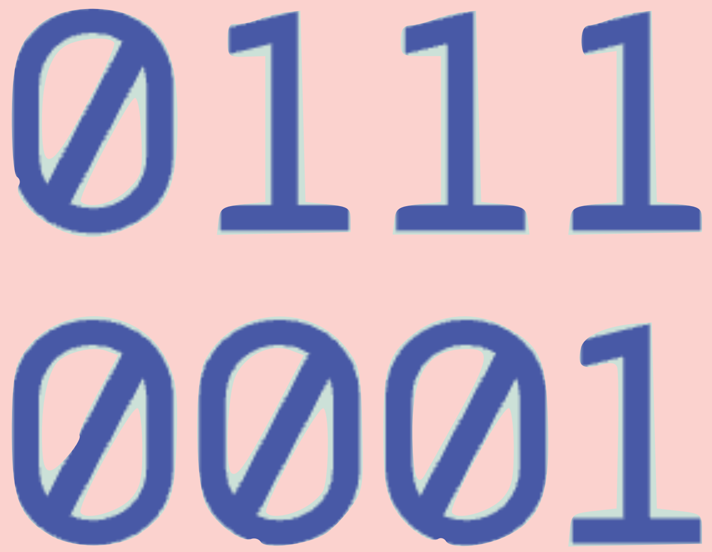
Isogeny-based

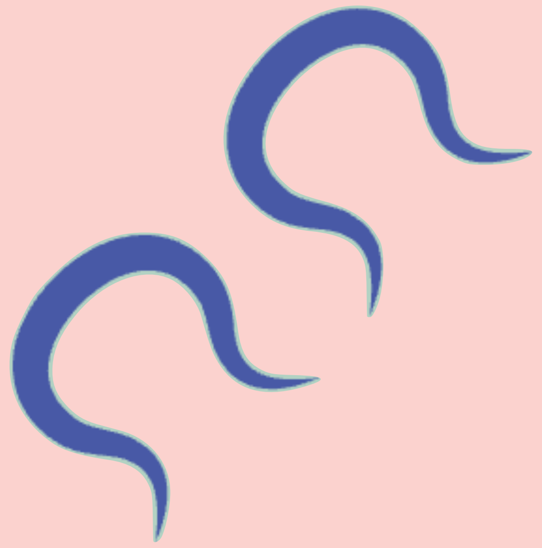


Images from Quanta Magazine

The Current State of Post-Quantum Crypto

Proof Systems (BARGs, SNARGs), Advanced Encryption (ABE, FE, FHE), …

Code-based	Lattice-based
	

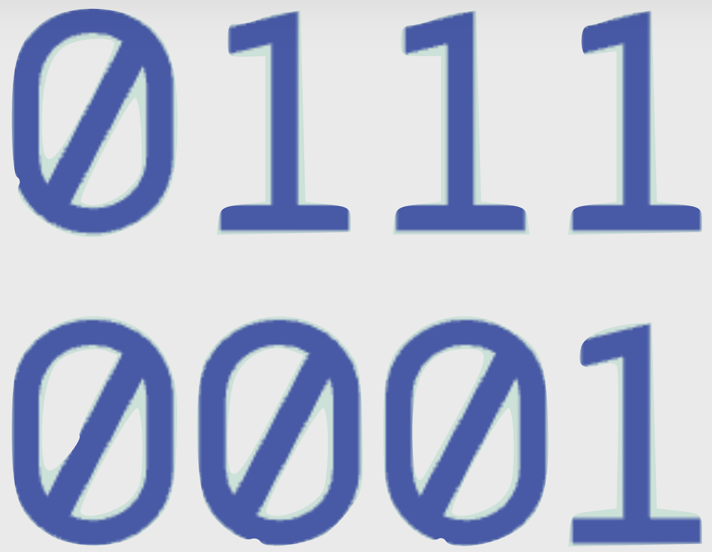
Multivariate-based	Isogeny-based
	

Images from Quanta Magazine

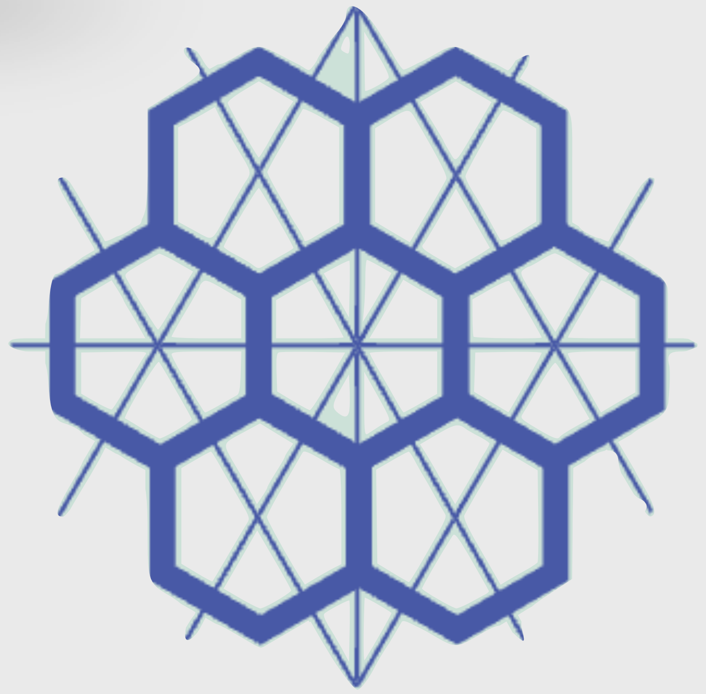
The Current State of Post-Quantum Crypto

Proof Systems (BARGs, SNARGs), Advanced Encryption (ABE, FE, FHE), …

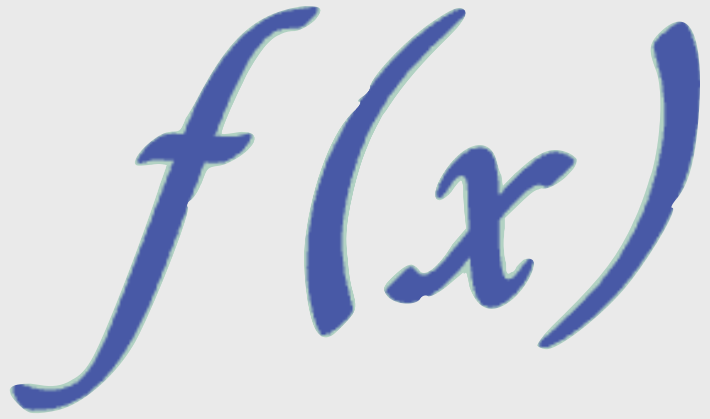
Why should we care about diversity of assumptions?



ttice-based



Multivariate-based



Isogeny-based

Images from Quanta Magazine

The Current State of Post-Quantum Crypto

Proof Systems (BARGs, SNARGs), Advanced Encryption (ABE, FE, FHE), …

Why should we care about diversity of assumptions?

ttice-based

1. **Hedge against advances in cryptanalysis**
   - Continual attempts to break lattices
2. **Different assumptions give different algebraic structures:**
   - Enable new feasibility results
   - Improved practical performance
3. **Cross-pollination with other areas:**
   - Coding theory, number theory, algebraic geometry, etc.

Images from Quanta Magazine

Code-Based vs. Lattice-Based Cryptography

Code-Based vs. Lattice-Based Cryptography

$$\left(\overbrace{A \;,\; s \cdot A}^{\text{Linear algebra}} + \overbrace{e}^{\text{Noise}} \right) \approx_c \left(A \;,\; u \right)$$

$$(A, sA + e) \approx_c (A, u)$$

Code-Based vs. Lattice-Based Cryptography

$$\left(\overbrace{A \;,\; s \cdot A}^{\text{Linear algebra}} + \overbrace{e}^{\text{Noise}} \right) \approx_c \left(A \;,\; u \right)$$

$$(A, sA + e) \approx_c (A, u)$$

Different noise models (sparse vs. small-magnitude) lead to:

- Little understanding of worst-case hardness
- Huge gap in cryptographic constructions

Code-Based vs. Lattice-Based Cryptography

$$\left(\overbrace{A \, , \, s \cdot A}^{\text{Linear algebra}} + \overbrace{e}^{\text{Noise}} \right) \approx_c \left(A \, , \, u \right)$$

$$(A, sA + e) \approx_c (A, u)$$

Different noise models (sparse vs. small-magnitude) lead to:

- Little understanding of worst-case hardness
- Huge gap in cryptographic constructions

Can we build more advanced primitives from code-based assumptions?

Our Result

Our Result

A new code-based assumption: Dense-Sparse LPN

- Variant of Learning Parity with Noise (LPN) with ***structured*** matrix distribution
- Initial cryptanalysis shows resistance to known attacks (linear tests, etc.)

Our Result

A new code-based assumption: Dense-Sparse LPN

- Variant of Learning Parity with Noise (LPN) with ***structured*** matrix distribution
- Initial cryptanalysis shows resistance to known attacks (linear tests, etc.)

We construct lossy trapdoor functions (LTDFs) from Dense-Sparse LPN

- Simple-to-state primitive with many applications such as CCA-secure PKE, etc.
- In the post-quantum setting, only achieved by lattices [PW08]

Our Result

A new code-based assumption: Dense-Sparse LPN

- Variant of Learning Parity with Noise (LPN) with ***structured*** matrix distribution
- Initial cryptanalysis shows resistance to known attacks (linear tests, etc.)

We construct lossy trapdoor functions (LTDFs) from Dense-Sparse LPN

- Simple-to-state primitive with many applications such as CCA-secure PKE, etc.
- In the post-quantum setting, only achieved by lattices **[PW08]**

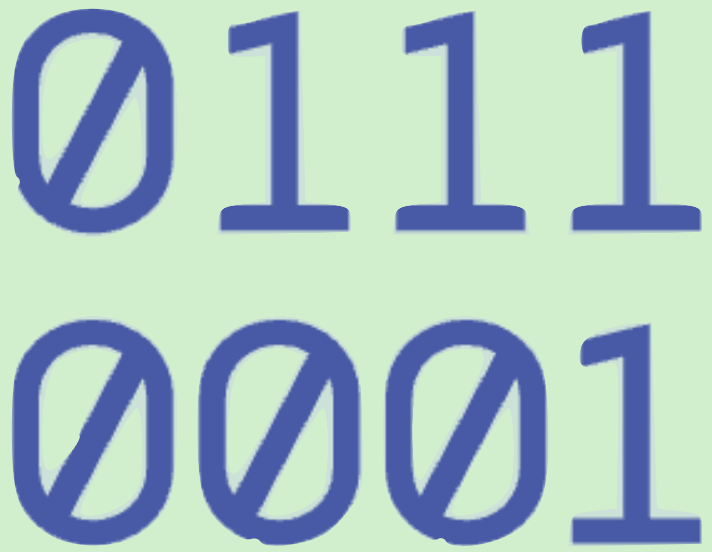
Why a new assumption?

- Overcome a barrier in noise management for LPN
- Circumvent a **new** attack against Sparse LPN (in relevant parameter regime)

Our Result

Lossy Trapdoor Functions

Code-based



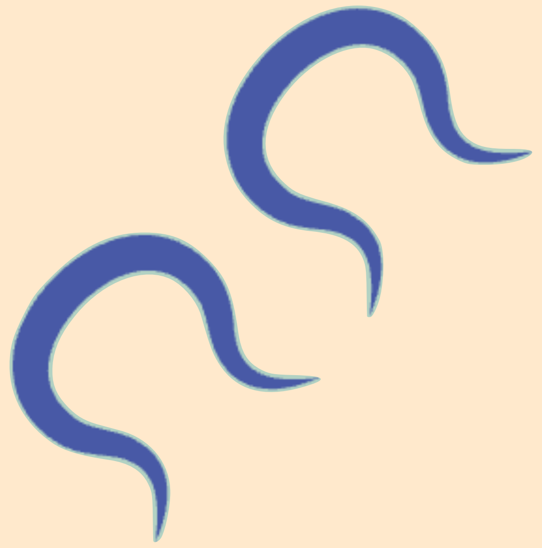
Ours!

Lattice-based

[PW08]

Multivariate-based

Isogeny-based



only achieve constant # bits of lossiness, not sufficient for many applications

Images from Quanta Magazine

Talk Outline

1. LTDF Template from Noisy Learning Problems

   (and why it fails from LPN)

2. Introducing Dense-Sparse LPN

3. Cryptanalysis & Open Questions

Talk Outline

1. LTDF Template from Noisy Learning Problems

   (and why it fails from LPN)

2. Introducing Dense-Sparse LPN

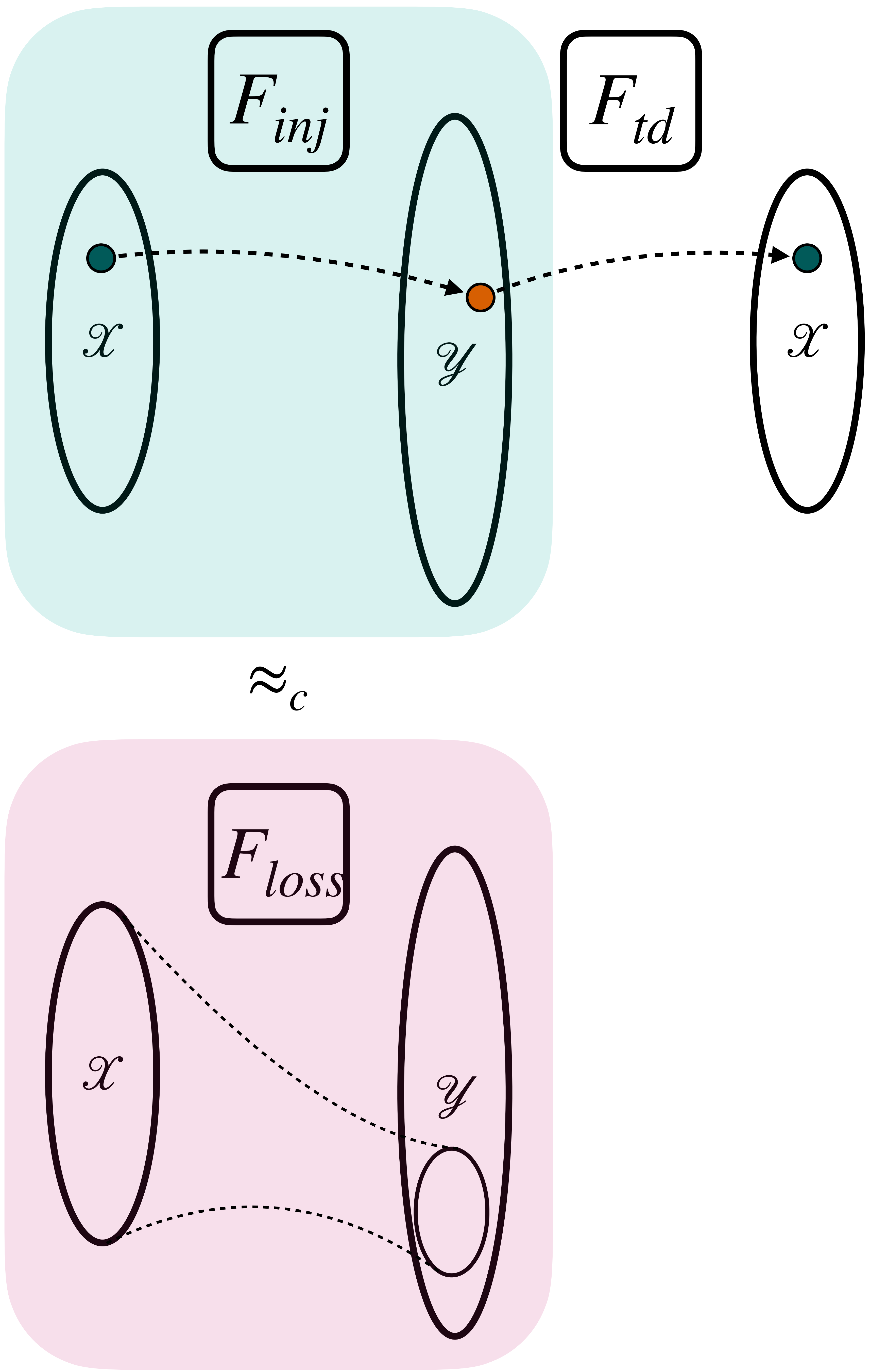
3. Cryptanalysis & Open Questions

Lossy Trapdoor Functions [PW08]

Lossy Trapdoor Functions [PW08]

$\mathsf{Setup}(1^\lambda, \mathsf{inj}) \to (\mathsf{ek}, \mathsf{td}) \quad \approx_c \quad \mathsf{Setup}(1^\lambda, \mathsf{loss}) \to \mathsf{ek}$

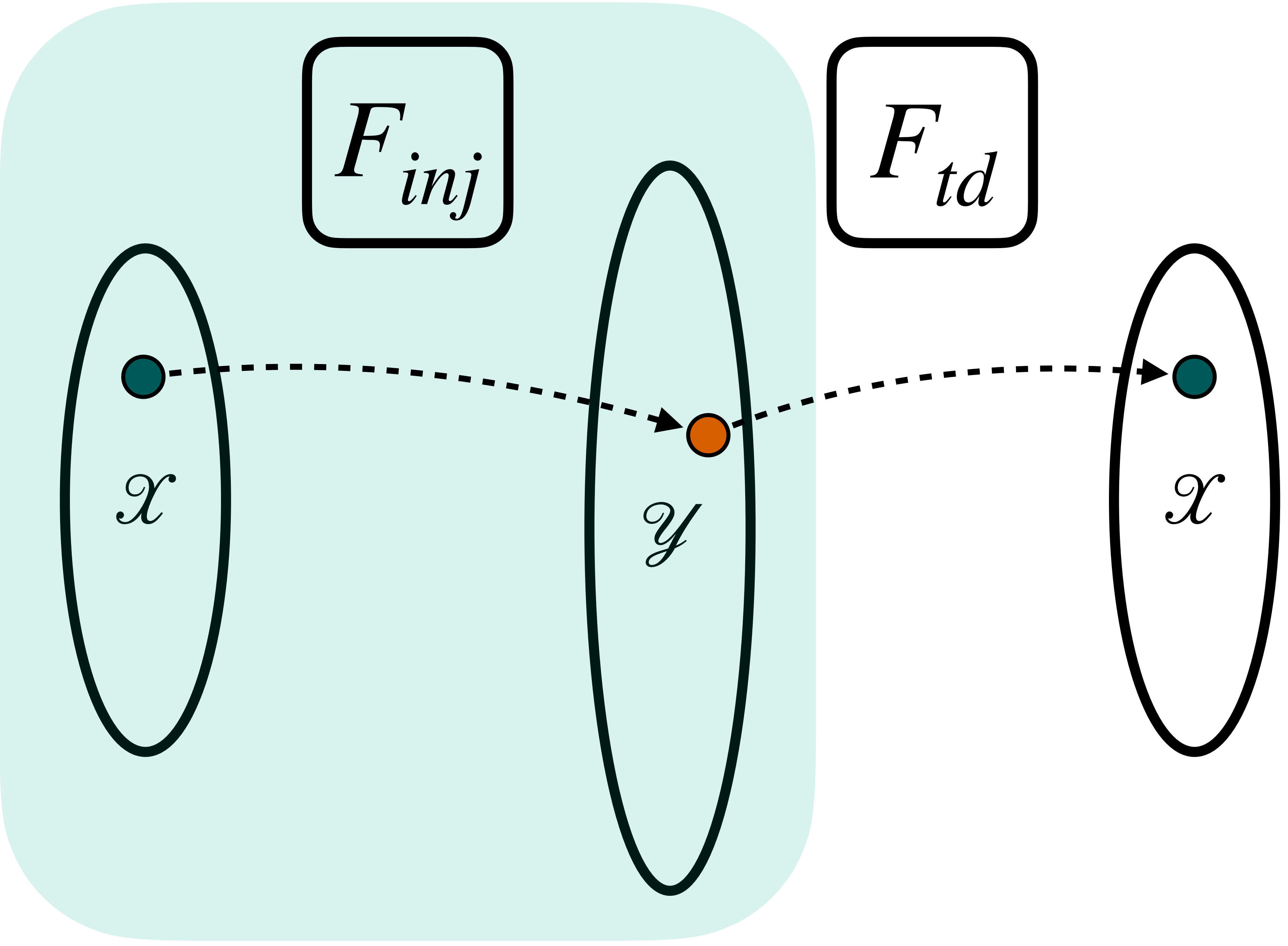
$\mathsf{Eval}(\mathsf{ek}, x) \to y \qquad\qquad \mathsf{Invert}(\mathsf{td}, y) \to x$



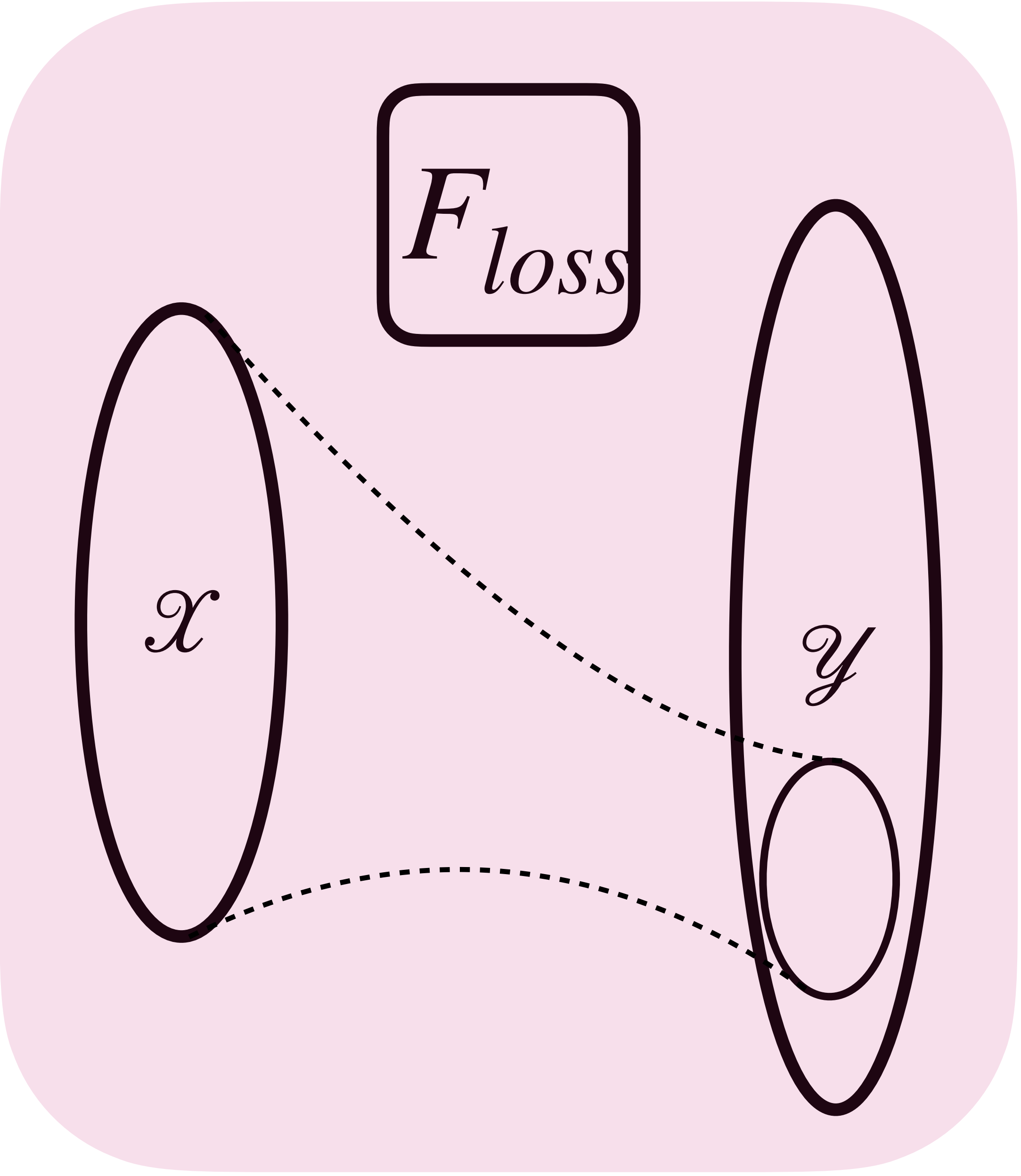
Lossy Trapdoor Functions [PW08]

$\mathsf{Setup}(1^\lambda, \mathsf{inj}) \rightarrow (\mathsf{ek}, \mathsf{td}) \quad \approx_c \quad \mathsf{Setup}(1^\lambda, \mathsf{loss}) \rightarrow \mathsf{ek}$

$\mathsf{Eval}(\mathsf{ek}, x) \rightarrow y \qquad \mathsf{Invert}(\mathsf{td}, y) \rightarrow x$



$\approx_c$



Applications of LTDFs:

CCA-secure encryption

Deterministic encryption

Collision-resistant hash functions

Analyzing OAEP

Selective opening security

Incompressible encodings

Computational extractors

Point function obfuscation

Pseudo-entropy functions

…and more!

Noisy Learning Problems: LWE vs. LPN

$$\left(\boxed{A \xleftarrow{\$} \mathcal{R}^{n\times m}} \;,\; \boxed{S \xleftarrow{\$} \mathcal{R}^{\ell\times n}} \cdot \boxed{A} + \boxed{E \xleftarrow{\$} \chi^{\ell\times m}} \right) \approx_c \left(\boxed{A \xleftarrow{\$} \mathcal{R}^{n\times m}} \;,\; \boxed{U \xleftarrow{\$} \mathcal{R}^{\ell\times m}} \right)$$

$$(A, S\cdot A + E) \approx_c (A, U)$$

Noisy Learning Problems: LWE vs. LPN

$$\left(\boxed{A \xleftarrow{\$} \mathscr{R}^{n\times m}}, \boxed{S \xleftarrow{\$} \mathscr{R}^{\ell\times n}} \cdot \boxed{A} + \boxed{E \xleftarrow{\$} \chi^{\ell\times m}} \right) \approx_c \left(\boxed{A \xleftarrow{\$} \mathscr{R}^{n\times m}}, \boxed{U \xleftarrow{\$} \mathscr{R}^{\ell\times m}} \right)$$

$$(A, S\cdot A + E) \approx_c (A, U)$$

<u>Learning with Errors:</u> [Regev05]

$\mathscr{R} = \mathbb{Z}/q\mathbb{Z}, \chi =$ Discrete Gaussian(α)

Entries of E are **<u>small</u>**

<u>Learning Parity with Noise:</u> [BKFL94]

$\mathscr{R} = \mathbb{F}_q$ (usually $q = 2$), $\chi =$ Bernoulli(ϵ)

Entries of E are **<u>mostly zero</u>**

LTDF Template from Noisy Learning Problem

LTDF Template from Noisy Learning Problem

Lossy Mode: $\mathsf{ek} = \big(A, B := S \cdot A + E\big)$ $\approx_c$ **Injective:** $\begin{cases} \mathsf{ek} = \big(A, B := S \cdot A + E + C\big) \\ \mathsf{td} = S \end{cases}$

suitable code

LTDF Template from Noisy Learning Problem

Lossy Mode: ek $= \big(A, B := S \cdot A + E\big)$ $\approx_c$ **Injective:** $\begin{cases} \text{ek} = \big(A, B := S \cdot A + E + C\big) \\ \text{td} = S \end{cases}$

suitable code

Function: $F : \text{Supp}(\chi)^m \to \mathbb{F}_2^{n+\ell}$ ($\text{Supp}(\chi)$ = support of error distribution)

LTDF Template from Noisy Learning Problem

Lossy Mode: $\mathsf{ek} = \big(A, B := S \cdot A + E\big) \quad \approx_c \quad$ **Injective:** $\begin{cases} \mathsf{ek} = \big(A, B := S \cdot A + E + C\big) \\ \mathsf{td} = S \end{cases}$

suitable code

Function: $F : \mathsf{Supp}(\chi)^m \to \mathbb{F}_2^{n+\ell}$ $\quad$ ($\mathsf{Supp}(\chi)$ = support of error distribution)

Evaluation: $F\big((A, B), x\big) = \big(A \cdot x, B \cdot x\big) \; = \; (A \cdot x, S \cdot A \cdot x + E \cdot x)$

Proof sketch: in lossy mode, second argument = first argument + noise

$\Longrightarrow$ requires $x \mapsto A \cdot x$ be ***compressing***, and $E \cdot x$ remains ***low noise***

LTDF Template from Noisy Learning Problem

Lossy Mode: $\mathsf{ek} = \big(A, B := S \cdot A + E\big)$ $\approx_c$ **Injective:** $\begin{cases} \mathsf{ek} = \big(A, B := S \cdot A + E + C\big) \\ \mathsf{td} = S \end{cases}$

suitable code

Function: $F : \mathsf{Supp}(\chi)^m \to \mathbb{F}_2^{n+\ell}$ ($\mathsf{Supp}(\chi)$ = support of error distribution)

Evaluation: $F\big(\,(A, B)\,, x\big) = \big(A \cdot x, B \cdot x\big) \;=\; (A \cdot x, S \cdot A \cdot x + E \cdot x)$

Proof sketch: in lossy mode, second argument = first argument + noise

$\Longrightarrow$ requires $x \mapsto A \cdot x$ be ***compressing***, and $E \cdot x$ remains ***low noise***

Inversion: $F^{-1}\big(S, (y_1, y_2)\,\big) = \mathsf{Decode}_C(y_2 - S \cdot y_1) \;=\; \mathsf{Decode}_C(C \cdot x + E \cdot x)$

Proof sketch: in injective mode, recover x via ***decoding*** from noise using C

LTDF Template from Noisy Learning Problem

Lossy Mode: ek $= \big(A, B := S \cdot A + E\big)$ $\approx_c$ **Injective:** $\begin{cases} \text{ek} = \big(A, B := S \cdot A + E + C\big) \\ \text{td} = S \end{cases}$

suitable code

Function: $F : \text{Supp}(\chi)^m \to \mathbb{F}_2^{n+\ell}$ ($\text{Supp}(\chi)$ = support of error distribution)

Evaluation: $F\big((A, B), x\big) = \big(A \cdot x, B \cdot x\big) = (A \cdot x, S \cdot A \cdot x + E \cdot x)$ focus on lossiness

Proof sketch: in lossy mode, second argument = first argument + noise

$\Longrightarrow$ requires $x \mapsto A \cdot x$ be ***compressing***, and $E \cdot x$ remains ***low noise***

Inversion: $F^{-1}\big(S, (y_1, y_2)\big) = \text{Decode}_C(y_2 - S \cdot y_1) = \text{Decode}_C(C \cdot x + E \cdot x)$

Proof sketch: in injective mode, recover x via ***decoding*** from noise using C

LTDF Template Fails for LPN

LTDF Template Fails for LPN

Hash Function: $\mathbb{F}_2^m \supsetneq \{t\text{-sparse}\} \ni x \quad \mapsto \quad A \cdot x \in \mathbb{F}_2^n$

Compression: $\binom{m}{t} \approx \left(\frac{m}{t}\right)^t > 2^n \quad \overset{\text{requires}}{\Longrightarrow} \quad m = n^{1+\Omega(1)}, \quad t = \Omega\left(\frac{n}{\log n}\right)$

LTDF Template Fails for LPN

Hash Function: $\mathbb{F}_2^m \supsetneq \{t\text{-sparse}\} \ni x \quad \mapsto \quad A \cdot x \in \mathbb{F}_2^n$

Compression: $\binom{m}{t} \approx \left(\frac{m}{t}\right)^t > 2^n \quad \overset{\text{requires}}{\Longrightarrow} \quad m = n^{1+\Omega(1)}, \quad t = \Omega\left(\frac{n}{\log n}\right)$

Accumulated Noise: $E \xleftarrow{\$} \mathrm{Ber}(\epsilon)^{\ell \times m} \quad \Longrightarrow \quad E \cdot x \xleftarrow{\$} \mathrm{Ber}(\delta)^{\ell}$

Noise growth: $\delta \approx \epsilon t \leq O(1) \quad \overset{\text{requires}}{\Longrightarrow} \quad \epsilon = O\left(\frac{\log n}{n}\right)$

LTDF Template Fails for LPN

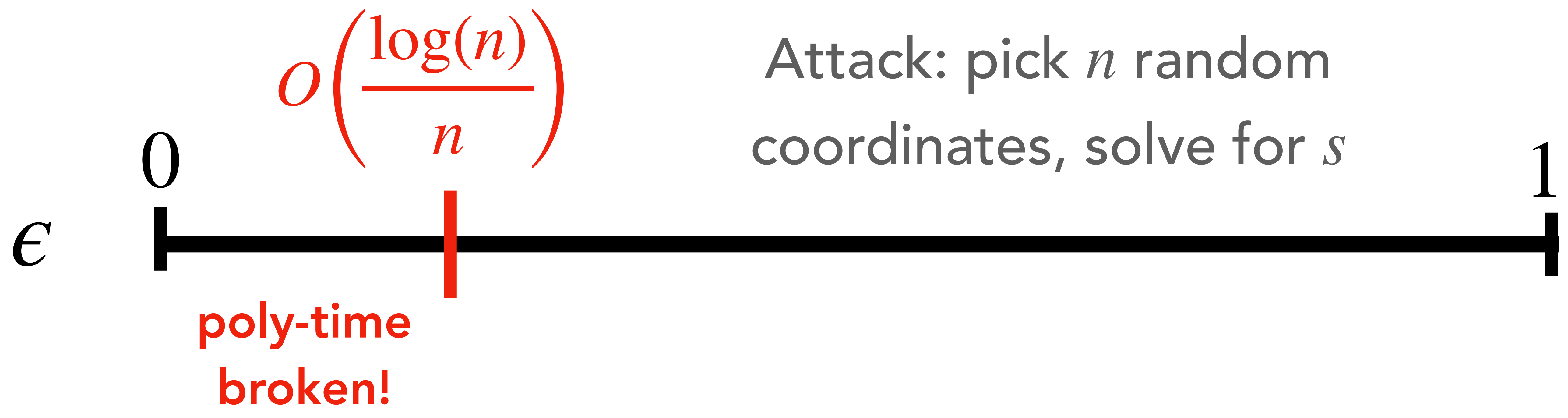
Hash Function: $\mathbb{F}_2^m \supsetneq \{t\text{-sparse}\} \ni x \quad \mapsto \quad A \cdot x \in \mathbb{F}_2^n$

Compression: $\binom{m}{t} \approx \left(\frac{m}{t}\right)^t > 2^n \quad \overset{\text{requires}}{\Longrightarrow} \quad m = n^{1+\Omega(1)}, \quad t = \Omega\left(\frac{n}{\log n}\right)$

Accumulated Noise: $E \xleftarrow{\$} \mathrm{Ber}(\epsilon)^{\ell \times m} \quad \Longrightarrow \quad E \cdot x \xleftarrow{\$} \mathrm{Ber}(\delta)^{\ell}$

Noise growth: $\delta \approx \epsilon t \leq O(1) \quad \overset{\text{requires}}{\Longrightarrow} \quad \epsilon = O\left(\frac{\log n}{n}\right)$

LPN Security:



Attack: pick n random coordinates, solve for s

LTDF Template Fails for LPN

Hash Function: $\mathbb{F}_2^m \supsetneq \{t\text{-sparse}\} \ni x \quad \mapsto \quad A \cdot x \in \mathbb{F}_2^n$

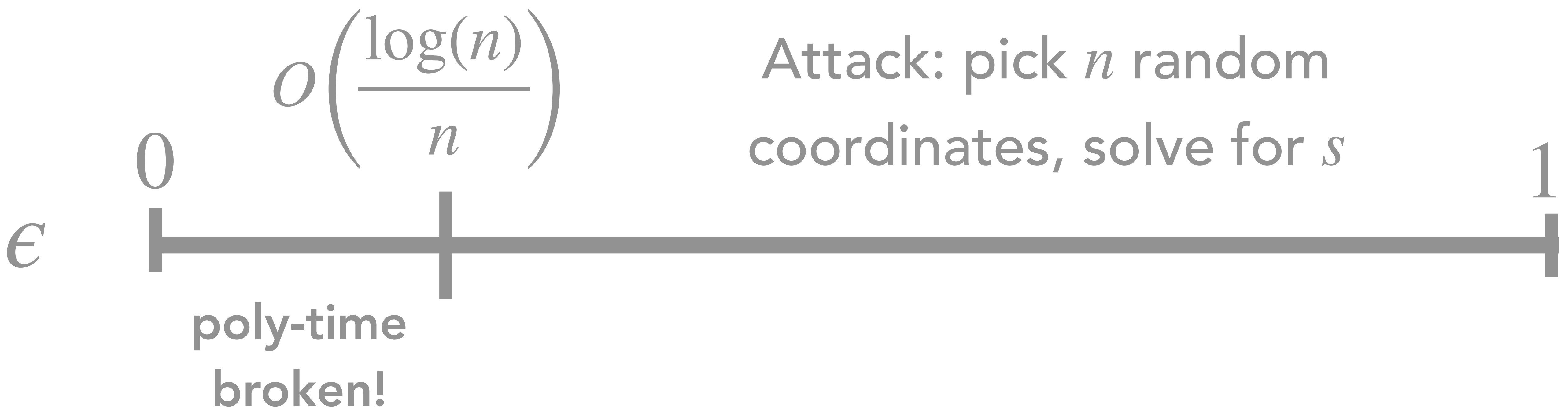
Compression: $\binom{m}{t} \approx \left(\frac{m}{t}\right)^t > 2^n \overset{\text{requires}}{\Longrightarrow} m = n^{1+\Omega(1)}, \quad t = \Omega\left(\frac{n}{\log n}\right)$

Accur

Can we achieve compression with better parameters?

ise growth: $\delta \approx \epsilon t \leq O(1) \overset{\text{requires}}{\Longrightarrow} \epsilon = O\left(\frac{\log n}{n}\right)$

LPN Security:



LTDF Template Fails for LPN

Hash Function: $\mathbb{F}_2^m \supsetneq \{t\text{-sparse}\} \ni x \quad \mapsto \quad A \cdot x \in \mathbb{F}_2^n$

Compression: $\binom{m}{t} \approx \left(\frac{m}{t}\right)^t > 2^n \overset{\text{requires}}{\Longrightarrow} m = n^{1+\Omega(1)}, \quad t = \Omega\left(\frac{n}{\log n}\right)$

Accur

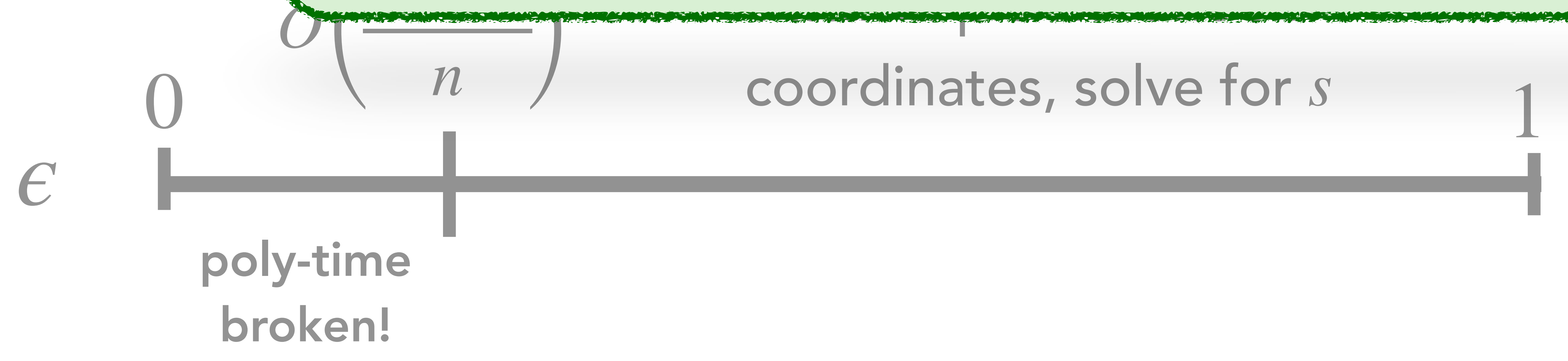
Can we achieve compression with better parameters?

ise growth: $\delta \approx \epsilon t \leq O($

Yes, by restricting the hash range

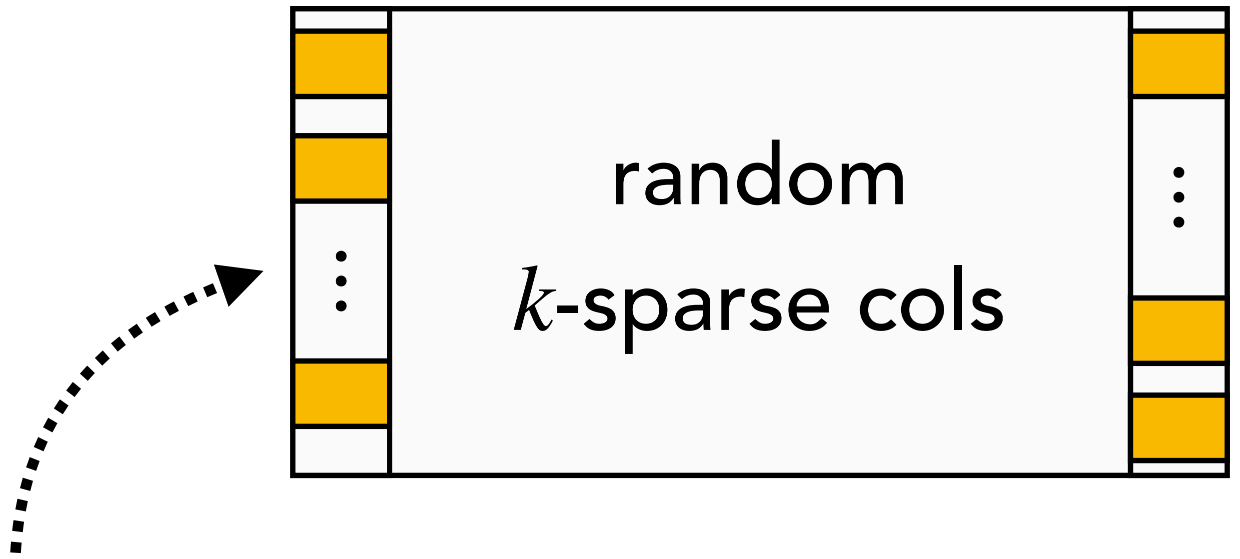
(via changing distribution of A)!

LPN Security:



Sparse Learning Parity with Noise

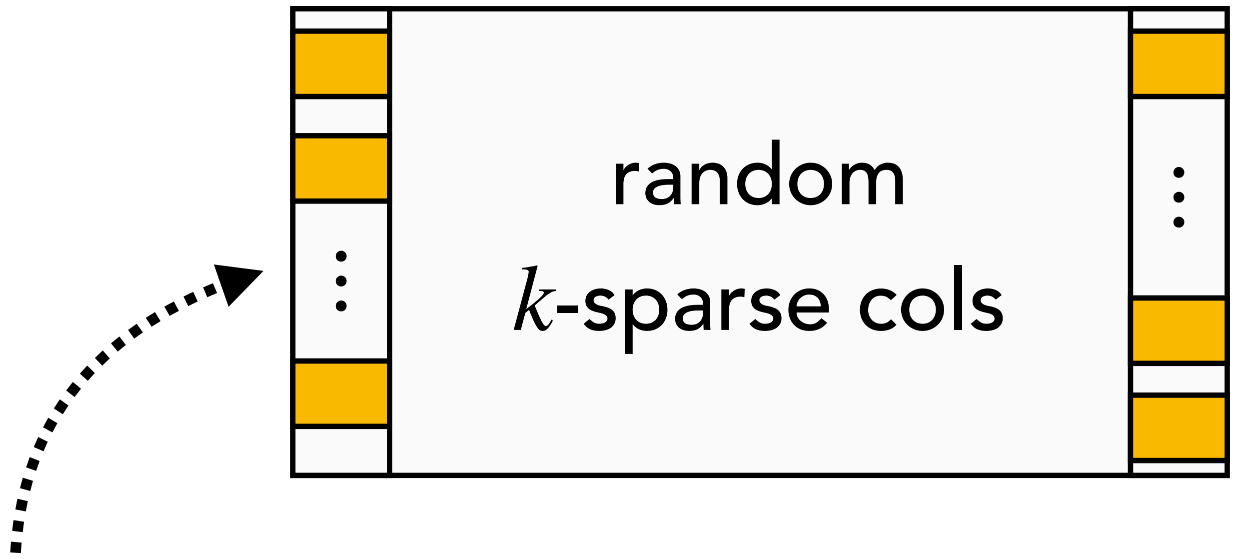
Sparse Learning Parity with Noise



$$(A, S \cdot A + E) \approx_c (A, U)$$

$$\left(\boxed{\begin{array}{c} A \xleftarrow{\$} \mathbb{F}_2^{n \times m} \\ k\text{-sparse cols} \end{array}} \,,\; \boxed{S \xleftarrow{\$} \mathbb{F}_2^{\ell \times n}} \cdot \boxed{\quad A \quad} + \boxed{E \xleftarrow{\$} \mathsf{Ber}(\epsilon)^{\ell \times m}} \right) \approx_c \left(\boxed{\begin{array}{c} A \xleftarrow{\$} \mathbb{F}_2^{n \times m} \\ k\text{-sparse cols} \end{array}} \,,\; \boxed{U \xleftarrow{\$} \mathbb{F}_2^{\ell \times m}} \right)$$

Sparse Learning Parity with Noise



$$(A, S \cdot A + E) \approx_c (A, U)$$

$$\left(\boxed{\begin{array}{c} A \xleftarrow{\$} \mathbb{F}_2^{n \times m} \\ k\text{-sparse cols} \end{array}}\ ,\ \boxed{S \xleftarrow{\$} \mathbb{F}_2^{\ell \times n}} \cdot \boxed{A} + \boxed{E \xleftarrow{\$} \mathsf{Ber}(\epsilon)^{\ell \times m}} \right) \approx_c \left(\boxed{\begin{array}{c} A \xleftarrow{\$} \mathbb{F}_2^{n \times m} \\ k\text{-sparse cols} \end{array}}\ ,\ \boxed{U \xleftarrow{\$} \mathbb{F}_2^{\ell \times m}} \right)$$

Well-studied variant of LPN [Alekhnovich03] with prior cryptographic applications

(PKE [ABW10], correlated randomness [ADI+17, AK23, BCG+23], HSS [DIJK23], etc.)

Our Setting: $m \ll n^{k/2}$, k **is constant*** **or slightly super-constant** $(\approx \log \log n)$

* Requires non-uniformly random distribution of A [AK19]

LTDF Template Also Fails for Sparse LPN

LTDF Template Also Fails for Sparse LPN

Hash Function: $\mathbb{F}_2^m \supsetneq \{t\text{-sparse}\} \ni x \quad \mapsto \quad A \cdot x \in \{\leq kt\text{-sparse}\} \subsetneq \mathbb{F}_2^n$

Compression: $\binom{m}{t} > \binom{n}{\leq kt}$ $\overset{\text{requires}}{\Longrightarrow}$ $t = \Omega\left(\left(\frac{n^k}{m}\right)^{1/(k-1)}\right)$

Example:

$k = 6$

$m = n^2$

$t \approx n^{0.8}$

LTDF Template Also Fails for Sparse LPN

Hash Function: $\mathbb{F}_2^m \supsetneq \{t\text{-sparse}\} \ni x \quad \mapsto \quad A \cdot x \in \{\leq kt\text{-sparse}\} \subsetneq \mathbb{F}_2^n$

Compression: $\binom{m}{t} > \binom{n}{\leq kt}$ **requires** $\Longrightarrow$ $t = \Omega\left(\left(\frac{n^k}{m}\right)^{1/(k-1)}\right)$

Accumulated Noise: $E \xleftarrow{\$} \mathrm{Ber}(\epsilon)^{\ell \times m} \Longrightarrow E \cdot x \xleftarrow{\$} \mathrm{Ber}(\delta)^{\ell}$

Noise rate: $\delta \approx \epsilon \cdot t = O(1)$ **requires** $\Longrightarrow$ $\epsilon = O\left(\frac{1}{t}\right) = O\left(\left(\frac{m}{n^k}\right)^{1/(k-1)}\right)$

Example:

$k = 6$

$m = n^2$

$t \approx n^{0.8}$

$\epsilon \approx n^{-0.8}$

LTDF Template Also Fails for Sparse LPN

Hash Function: $\mathbb{F}_2^m \supsetneq \{t\text{-sparse}\} \ni x \quad \mapsto \quad A \cdot x \in \{\leq kt\text{-sparse}\} \subsetneq \mathbb{F}_2^n$

Compression: $\binom{m}{t} > \binom{n}{\leq kt}$ **requires** $\Longrightarrow$ $t = \Omega\left(\left(\frac{n^k}{m}\right)^{1/(k-1)}\right)$

Accumulated Noise: $E \xleftarrow{\$} \mathrm{Ber}(\epsilon)^{\ell \times m} \quad \Longrightarrow \quad E \cdot x \xleftarrow{\$} \mathrm{Ber}(\delta)^{\ell}$

Noise rate: $\delta \approx \epsilon \cdot t = O(1)$ **requires** $\Longrightarrow$ $\epsilon = O\left(\frac{1}{t}\right) = O\left(\left(\frac{m}{n^k}\right)^{1/(k-1)}\right)$

Example:

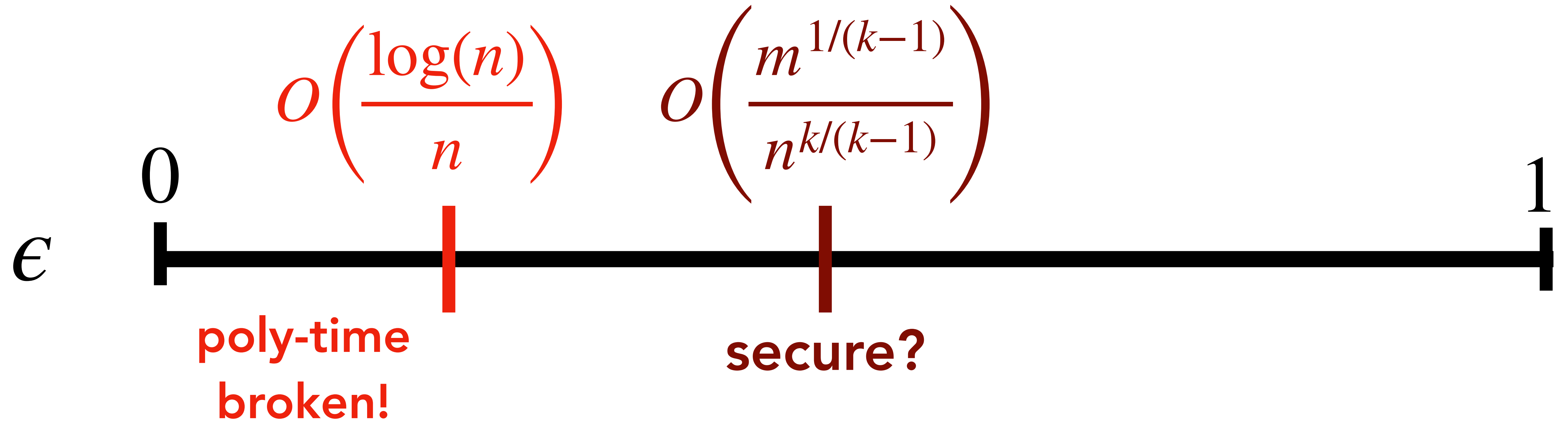
$k = 6$

$m = n^2$

$t \approx n^{0.8}$

$\epsilon \approx n^{-0.8}$

Sparse LPN Security:



LTDF Template Also Fails for Sparse LPN

Hash Function: $\mathbb{F}_2^m \supsetneq \{t\text{-sparse}\} \ni x \quad \mapsto \quad A \cdot x \in \{\leq kt\text{-sparse}\} \subsetneq \mathbb{F}_2^n$

Compression: $\binom{m}{t} > \binom{n}{\leq kt} \overset{\text{requires}}{\Longrightarrow} t = \Omega\left(\left(\frac{n^k}{m}\right)^{1/(k-1)}\right)$

Accumulated Noise: $E \xleftarrow{\$} \mathrm{Ber}(\epsilon)^{\ell \times m} \Longrightarrow E \cdot x \xleftarrow{\$} \mathrm{Ber}(\delta)^{\ell}$

Noise rate: $\delta \approx \epsilon \cdot t = O(1) \overset{\text{requires}}{\Longrightarrow} \epsilon = O\left(\frac{1}{t}\right) = O\left(\left(\frac{m}{n^k}\right)^{1/(k-1)}\right)$

Example:

$k = 6$

$m = n^2$

$t \approx n^{0.8}$

$\epsilon \approx n^{-0.8}$

Sparse LPN Security:

ϵ: 0 ... $O\left(\frac{\log(n)}{n}\right)$ ... $O\left(\frac{m^{1/(k-1)}}{n^{k/(k-1)}}\right)$ ... 1

poly-time broken!

secure?

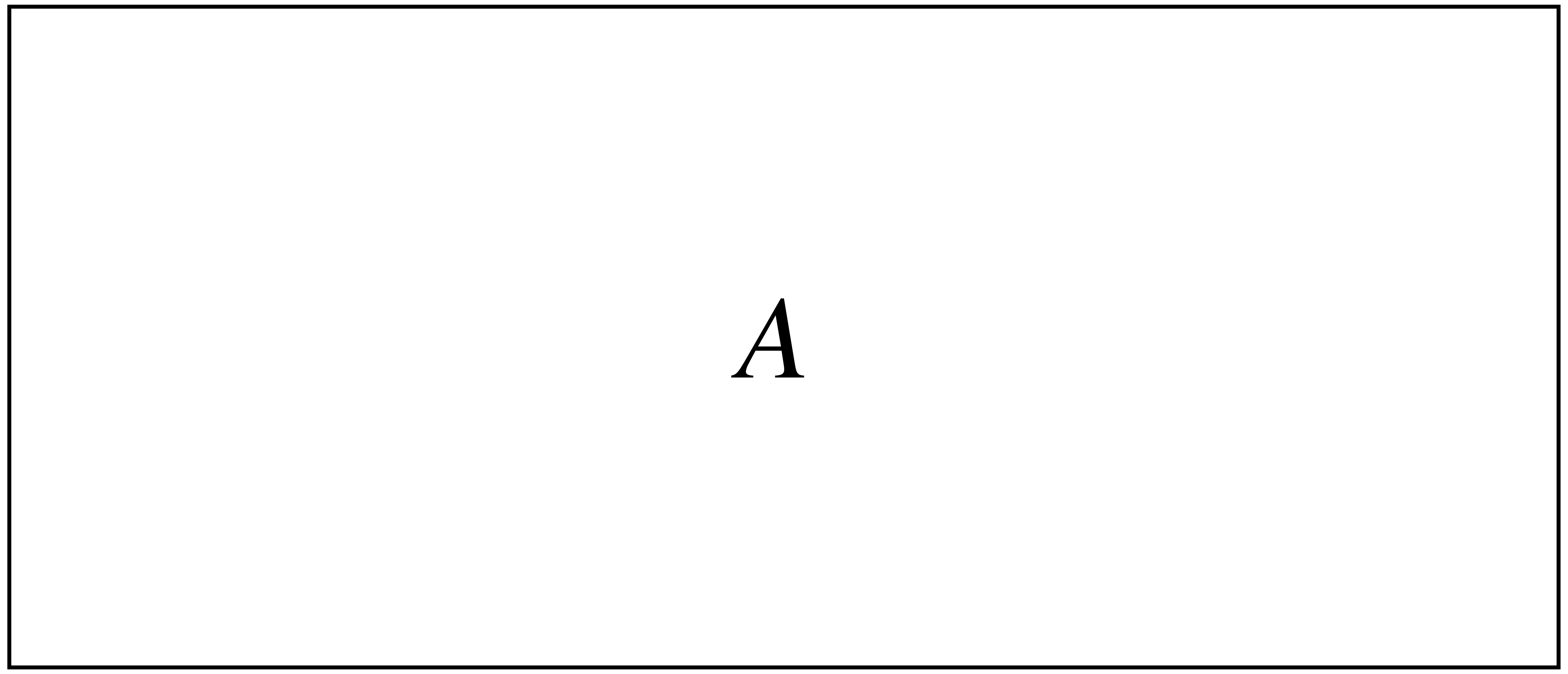
no! (our new attack)

Attack on Sparse LPN in Compression Regime

Attack on Sparse LPN in Compression Regime

random k-sparse

Given $\left(A \in \mathbb{F}_2^{n\times m}, u \in \mathbb{F}_2^{1\times m}\right)$:



Attack on Sparse LPN in Compression Regime

random k-sparse

Given $\left(A \in \mathbb{F}_2^{n\times m}, u \in \mathbb{F}_2^{1\times m}\right)$:

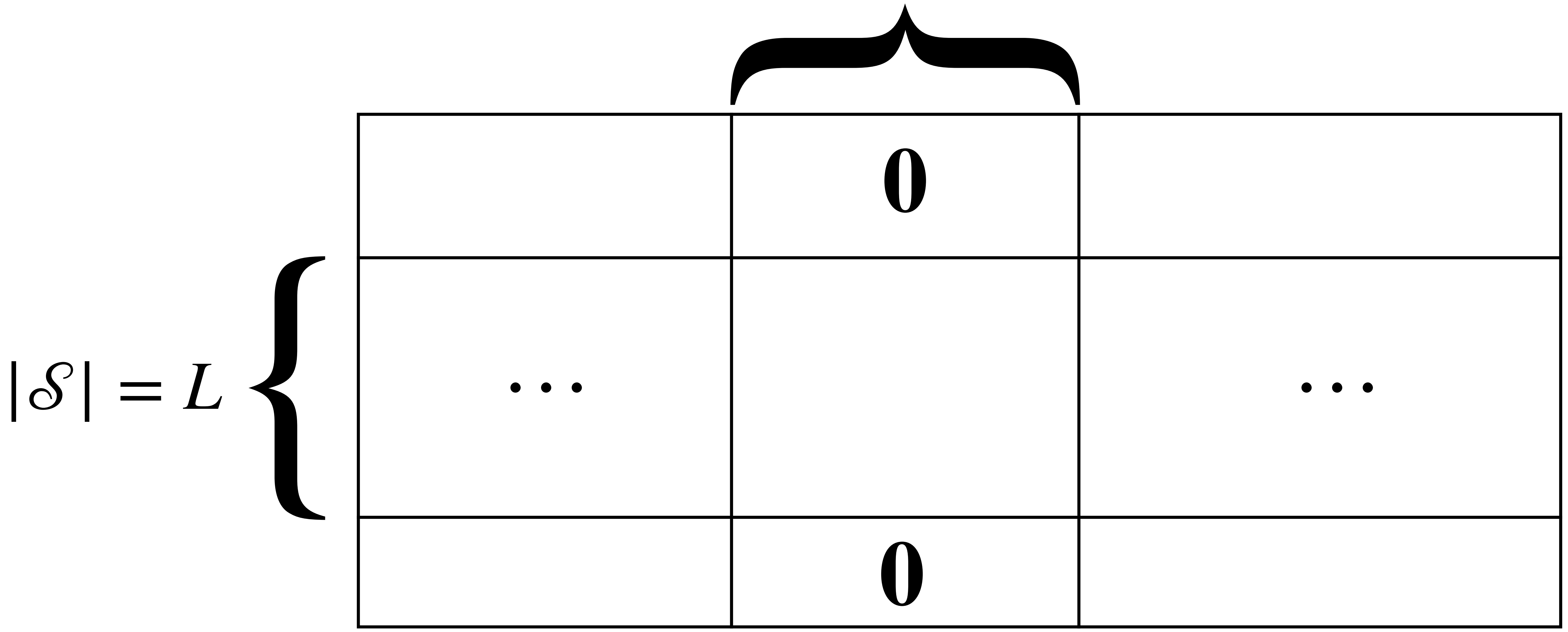
1. Pick a random subset $\mathcal{S}$ of size L

Attack on Sparse LPN in Compression Regime

random k-sparse

Given $\left(A \in \mathbb{F}_2^{n\times m}, u \in \mathbb{F}_2^{1\times m}\right)$:

1. Pick a random subset $\mathcal{S}$ of size L
2. Find all columns whose non-zero entries are all in $\mathcal{S}$

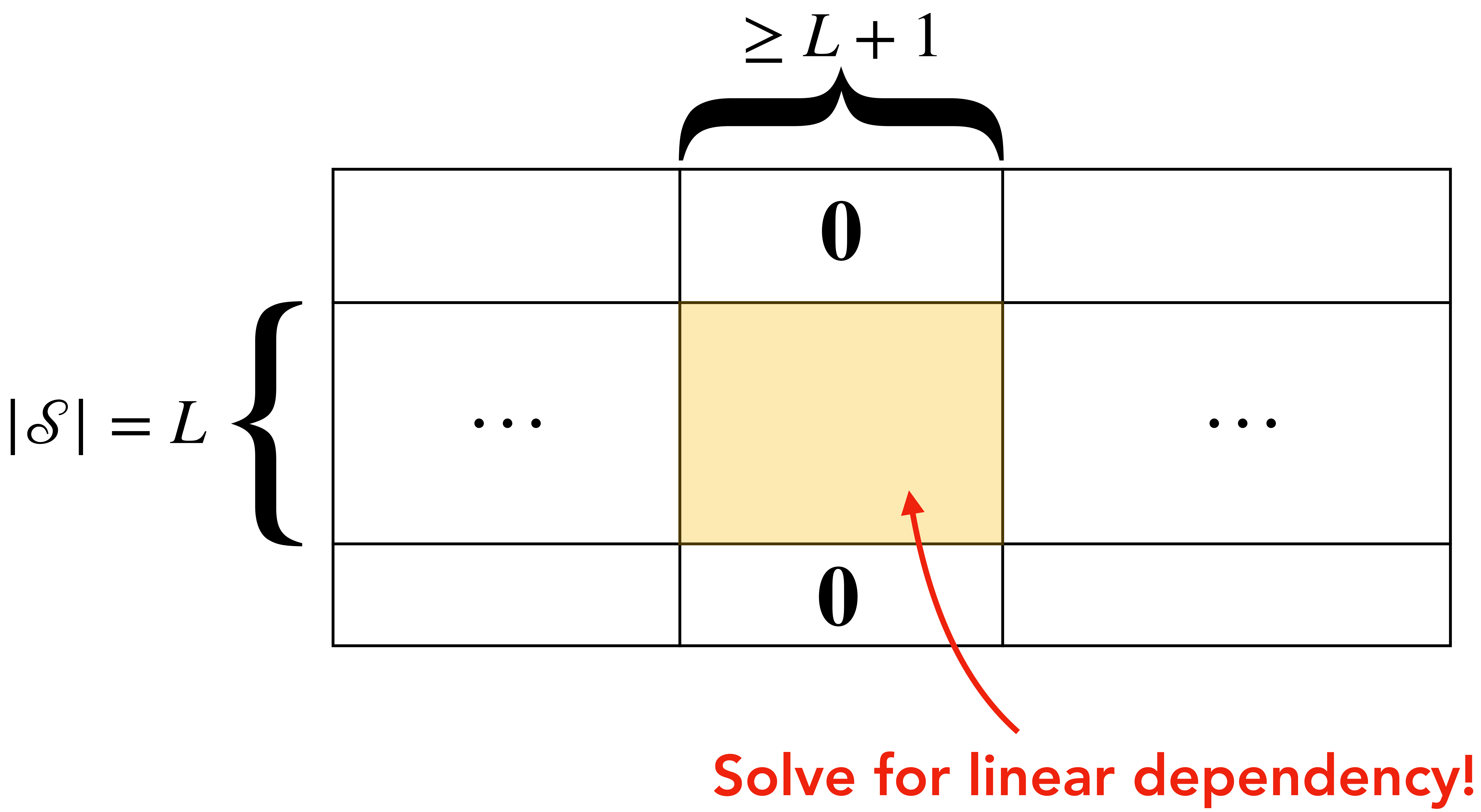


Attack on Sparse LPN in Compression Regime

random k-sparse

Given $\left(A \in \mathbb{F}_2^{n\times m}, u \in \mathbb{F}_2^{1\times m}\right)$:

1. Pick a random subset $\mathcal{S}$ of size L
2. Find all columns whose non-zero entries are all in $\mathcal{S}$
3. If there are $> L$ columns, find a linear dependency, e.g. $\leq L$-sparse $x \in \mathbb{F}_2^m$ such that $A \cdot x = 0^n$



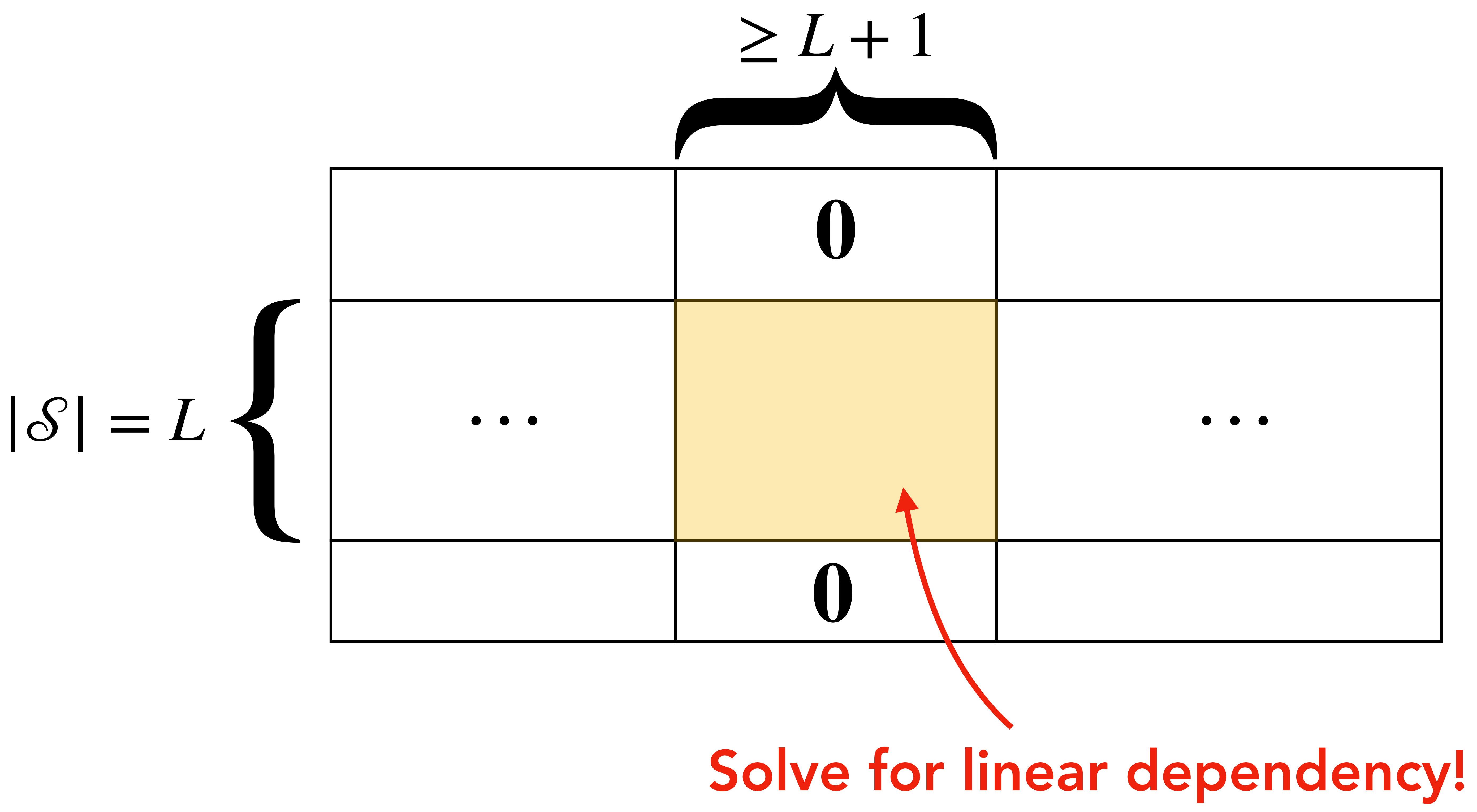
Attack on Sparse LPN in Compression Regime

random k-sparse

Given $\left(A \in \mathbb{F}_2^{n\times m}, u \in \mathbb{F}_2^{1\times m}\right)$:

1. Pick a random subset $\mathcal{S}$ of size L
2. Find all columns whose non-zero entries are all in $\mathcal{S}$
3. If there are $> L$ columns, find a linear dependency, e.g. $\leq L$-sparse $x \in \mathbb{F}_2^m$ such that $A \cdot x = 0^n$
4. Compute $\langle u, x \rangle$ to detect bias.

(more likely 0 if $u = sA + e$)



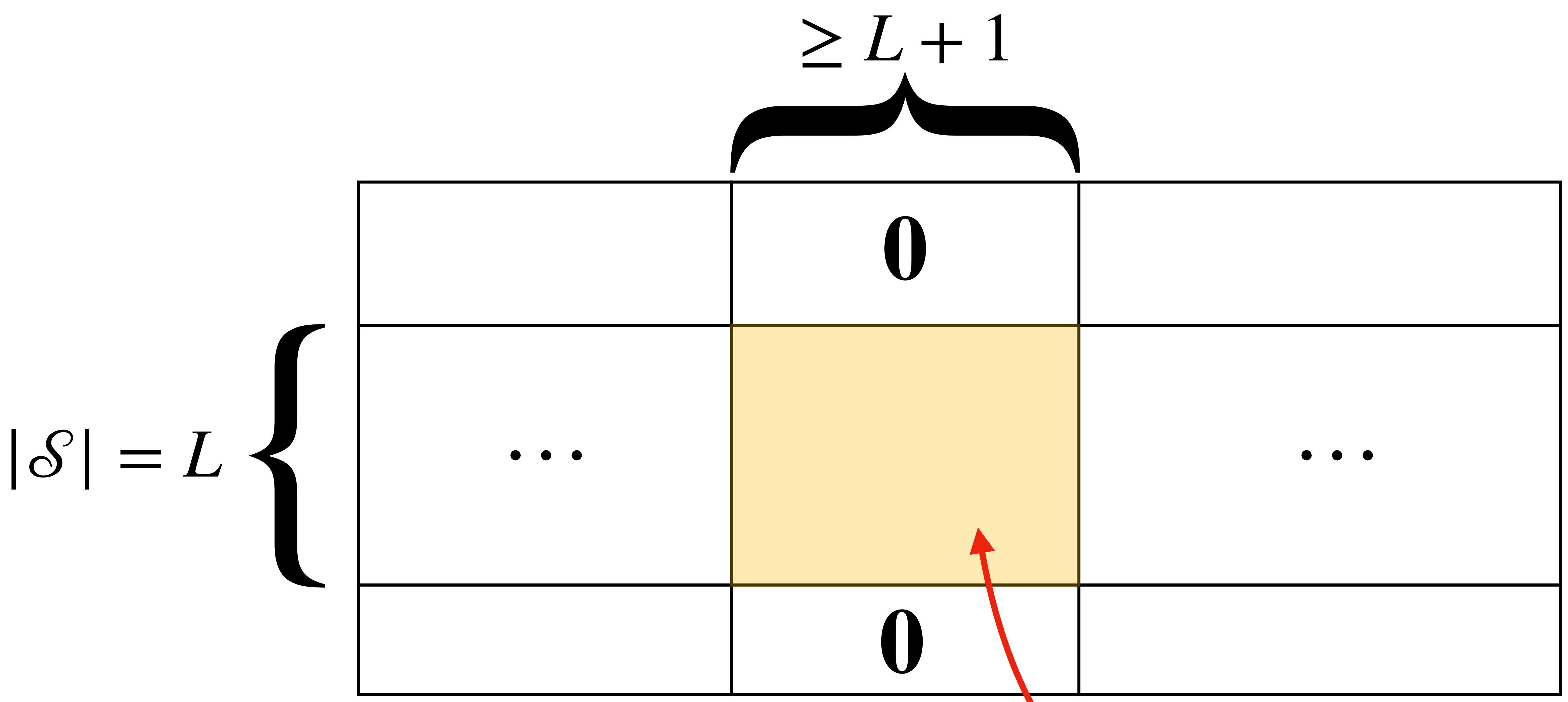
Attack on Sparse LPN in Compression Regime

random k-sparse

Given $\left(A \in \mathbb{F}_2^{n\times m}, u \in \mathbb{F}_2^{1\times m}\right)$:

1. Pick a random subset $\mathcal{S}$ of size L
2. Find all columns whose non-zero entries are all in $\mathcal{S}$
3. If there are $> L$ columns, find a linear dependency, e.g. $\leq L$-sparse $x \in \mathbb{F}_2^m$ such that $A \cdot x = 0^n$
4. Compute $\langle u, x\rangle$ to detect bias.

(more likely 0 if $u = sA + e$)



Solve for linear dependency!

Want: $\mathbb{E}[\#\,\text{cols}] = m \cdot \dfrac{\binom{L}{k}}{\binom{n}{k}} > L \iff L \approx \left(\dfrac{n^k}{m}\right)^{\frac{1}{k-1}}$

Same parameters for compression!

Attack on Sparse LPN in Compression Regime

random k-sparse

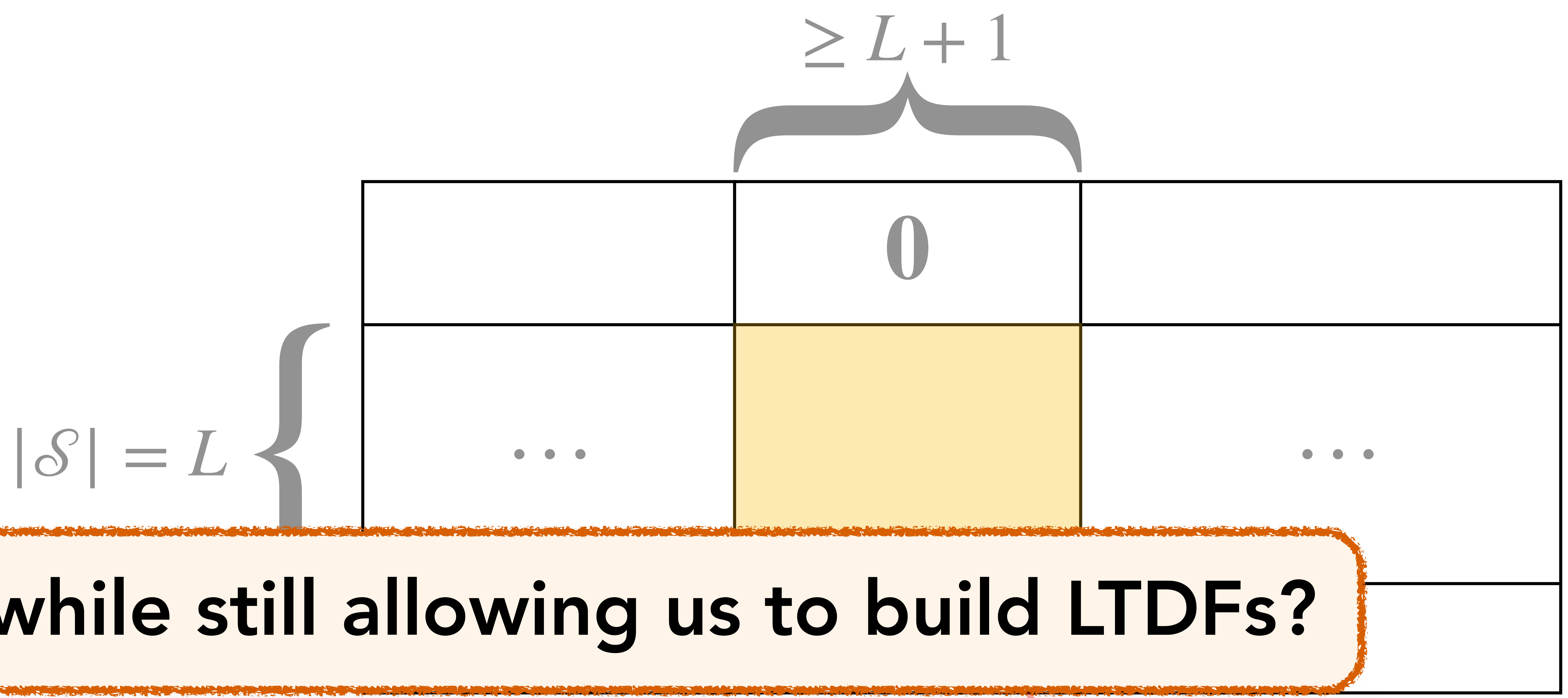
Given $\left(A \in \mathbb{F}_2^{n\times m}, u \in \mathbb{F}_2^{1\times m}\right)$:

1. Pick a random subset $\mathcal{S}$ of size L
2. Find ...

   ent...

3. If there are $> L$ columns, find a linear dependency, e.g. $\leq L$-sparse $x \in \mathbb{F}_2^m$ such that $A \cdot x = 0^n$
4. Compute $\langle u, x\rangle$ to detect bias.

   (more likely 0 if $u = sA + e$)



Can we avoid this attack while still allowing us to build LTDFs?

Solve for linear dependency!

Want: $\mathbb{E}[\#\,\text{cols}] = m \cdot \frac{\binom{L}{k}}{\binom{n}{k}} > L \iff L \approx \left(\frac{n^k}{m}\right)^{\frac{1}{k-1}}$

Same parameters for compression!

Attack on Sparse LPN in Compression Regime

random k-sparse

Given $\left(A \in \mathbb{F}_2^{n \times m}, u \in \mathbb{F}_2^{1 \times m}\right)$:

1. Pick a random subset $\mathcal{S}$ of size L
2. Fin[...]

   ent[...]

3. [...] there are $> L$ columns, [...]

   [...]ear dependency, e.g. $\le$[...]

   $x \in \mathbb{F}_2^m$ such that $A \cdot x = 0$[...]

4. Compute $\langle u, x \rangle$ to detect bias.

   (more likely 0 if $u = sA + e$)

$$\ge L + 1$$

$$|\mathcal{S}| = L \left\{ \begin{array}{|c|c|c|} \hline & \mathbf{0} & \\ \hline \cdots & & \cdots \\ \hline & & \\ \hline \end{array} \right.$$

Can we avoid this attack while still allowing us to build LTDFs?

Perhaps, by masking sparsity pattern of A!

[...]pendency!

Want: $\mathbb{E}[\#\text{ cols}] = m \cdot \frac{[\ldots]}{\binom{n}{k}} > L \iff L \approx [\ldots] \left(\frac{n^k}{n[\ldots]}\right)^{\frac{1}{k-1}}$

Same parameters for compression!

Talk Outline

1. LTDF Template from Noisy Learning Problems

   (and why it fails from LPN)

2. Introducing Dense-Sparse LPN

3. Cryptanalysis & Open Questions

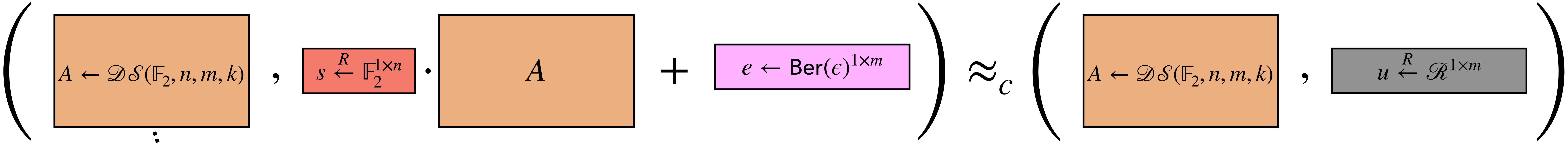
Dense-Sparse LPN

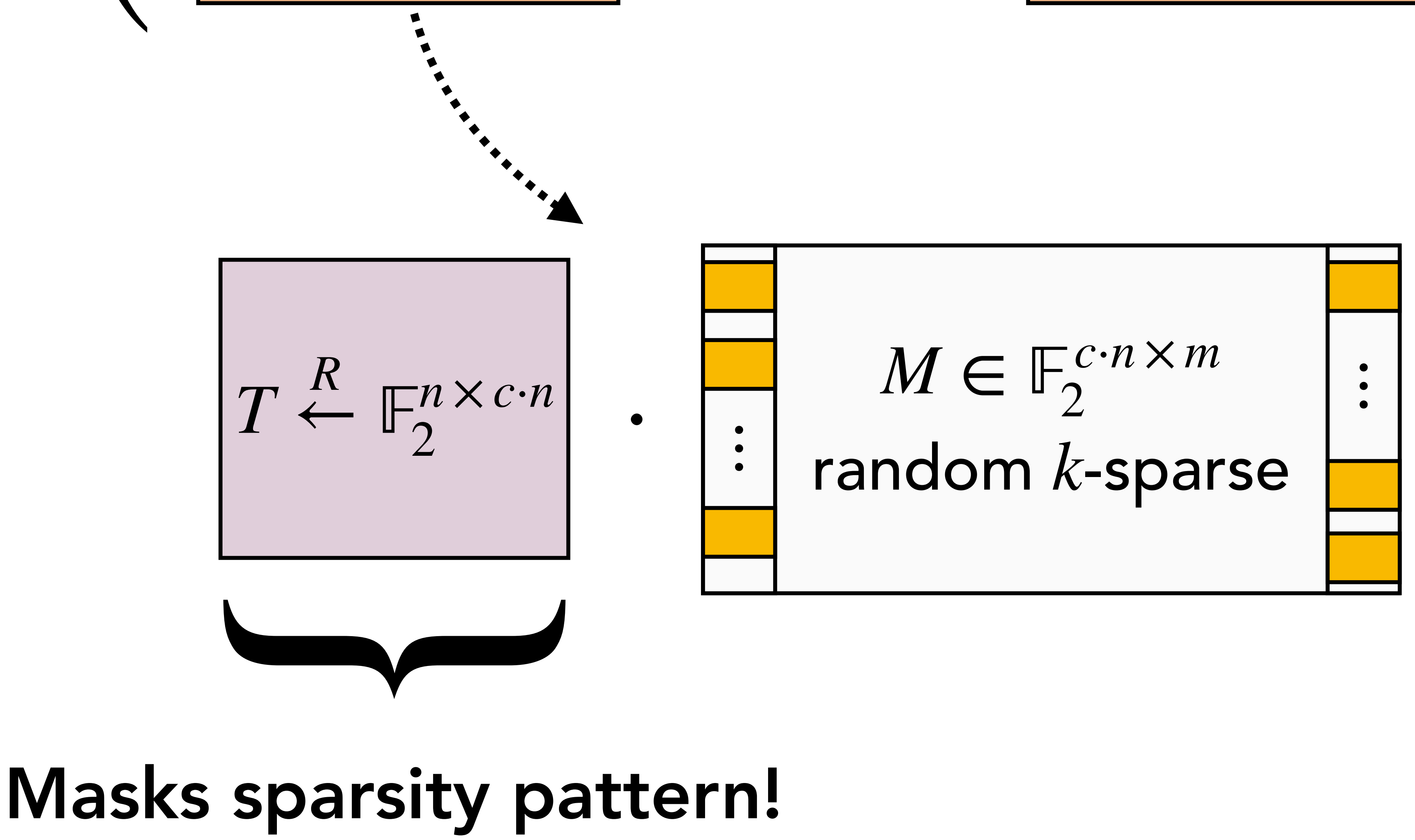
$$(A, sA + e) \approx_c (A, u)$$

$$\left(\boxed{A \leftarrow \mathcal{DS}(\mathbb{F}_2, n, m, k)} \,,\, \boxed{s \xleftarrow{R} \mathbb{F}_2^{1\times n}} \cdot \boxed{A} + \boxed{e \leftarrow \mathsf{Ber}(\epsilon)^{1\times m}} \right) \approx_c \left(\boxed{A \leftarrow \mathcal{DS}(\mathbb{F}_2, n, m, k)} \,,\, \boxed{u \xleftarrow{R} \mathcal{R}^{1\times m}} \right)$$

Dense-Sparse LPN

$$(A, sA + e) \approx_c (A, u)$$





where $A = T \cdot M$ with $T \in \mathbb{F}_2^{n\times c\cdot n}$ random (dense),

$M \in \mathbb{F}_2^{c\cdot n\times m}$ is random k-sparse, $c > 1$ any constant

(say $c = 1.1$)

Dense-Sparse LPN

$$(A, sA + e) \approx_c (A, u)$$

$$\left(A \leftarrow \mathscr{DS}(\mathbb{F}_2, n, m, k) \;,\; s \xleftarrow{R} \mathbb{F}_2^{1\times n} \cdot A + e \leftarrow \mathsf{Ber}(\epsilon)^{1\times m} \right) \approx_c \left(A \leftarrow \mathscr{DS}(\mathbb{F}_2, n, m, k) \;,\; u \xleftarrow{R} \mathscr{R}^{1\times m} \right)$$

$$\underbrace{T \xleftarrow{R} \mathbb{F}_2^{n\times c\cdot n}}_{\text{Masks sparsity pattern!}} \cdot \begin{matrix} M \in \mathbb{F}_2^{c\cdot n \times m} \\ \text{random } k\text{-sparse} \end{matrix}$$

where $A = T \cdot M$ with $T \in \mathbb{F}_2^{n\times c\cdot n}$ random (dense), $M \in \mathbb{F}_2^{c\cdot n\times m}$ is random k-sparse, $c > 1$ any constant

(say $c = 1.1$)

Inspiration from McEliece: hide code via linear transformation

Dense-Sparse LPN

$$(A, sA + e) \approx_c (A, u)$$

$$\left(\boxed{A \leftarrow \mathcal{DS}(\mathbb{F}_2, n, m, k)}\ ,\ \boxed{s \xleftarrow{R} \mathbb{F}_2^{1\times n}} \cdot \boxed{A} + \boxed{e \leftarrow \text{Ber}(\epsilon)^{1\times m}} \right) \approx_c \left(\boxed{A \leftarrow \mathcal{DS}(\mathbb{F}_2, n, m, k)}\ ,\ \boxed{u \xleftarrow{R} \mathcal{R}^{1\times m}} \right)$$

$$\underbrace{\boxed{T \xleftarrow{R} \mathbb{F}_2^{n\times c\cdot n}}} \cdot \boxed{\begin{matrix} M \in \mathbb{F}_2^{c\cdot n\times m} \\ \text{random } k\text{-sparse} \end{matrix}}$$

Masks sparsity pattern!

where $A = T \cdot M$ with $T \in \mathbb{F}_2^{n\times c\cdot n}$ random (dense),

$M \in \mathbb{F}_2^{c\cdot n\times m}$ is random k-sparse, $c > 1$ any constant

(say $c = 1.1$)

Inspiration from McEliece: hide code via linear transformation

Now LTDF construction works! ($T \cdot M \cdot x$ has image size at most $M \cdot x$)

Putting Things Together: LTDF Parameters

Putting Things Together: LTDF Parameters

<u>Theorem:</u> LTDF from Dense-Sparse LPN that loses a factor $D > 1$ in lossy mode...

$\Longrightarrow$ requires Dense-Sparse LPN with $m \ll n^{k/2}$ and $\epsilon \ll \left(\frac{m}{n^{Dk}}\right)^{\frac{1}{Dk-1}}$

Putting Things Together: LTDF Parameters

Theorem: LTDF from Dense-Sparse LPN that loses a factor $D > 1$ in lossy mode…

$\implies$ requires Dense-Sparse LPN with $m \ll n^{k/2}$ and $\epsilon \ll \left(\frac{m}{n^{Dk}}\right)^{\frac{1}{Dk-1}}$

Concrete settings: $k = 6$, $m = n^2$, any $\delta > 0$

- $D = 10$ (loses $90\,\%$ of input): $\epsilon = n^{-\frac{58}{59} - \delta} \approx n^{-0.984}$
- $D = 2$ (loses $50\,\%$ of input): $\epsilon = n^{-\frac{10}{11} - \delta} \approx n^{-0.91}$
- $D \to 1$: $\epsilon = n^{-\frac{6}{7} - \delta} \approx n^{-0.86}$

Putting Things Together: LTDF Parameters

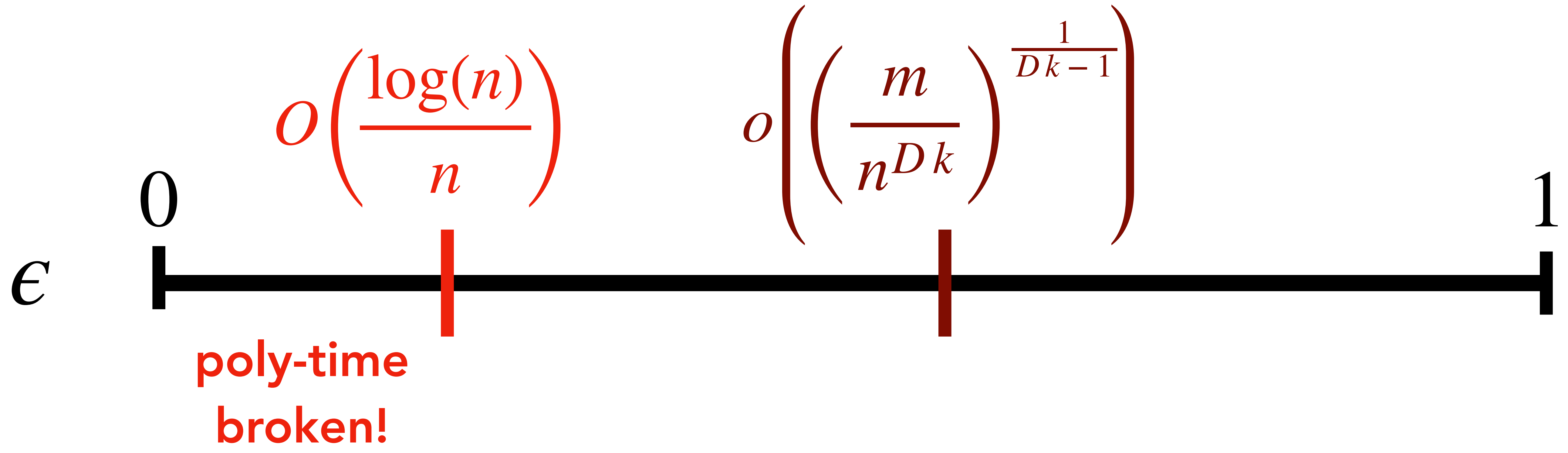
Theorem: LTDF from Dense-Sparse LPN that loses a factor $D > 1$ in lossy mode…

$\implies$ requires Dense-Sparse LPN with $m \ll n^{k/2}$ and $\epsilon \ll \left(\frac{m}{n^{Dk}}\right)^{\frac{1}{Dk-1}}$

Concrete settings: $k = 6$, $m = n^2$, any $\delta > 0$

- $D = 10$ (loses $90\,\%$ of input): $\epsilon = n^{-\frac{58}{59} - \delta} \approx n^{-0.984}$
- $D = 2$ (loses $50\,\%$ of input): $\epsilon = n^{-\frac{10}{11} - \delta} \approx n^{-0.91}$
- $D \to 1$: $\epsilon = n^{-\frac{6}{7} - \delta} \approx n^{-0.86}$

How secure is this?



Talk Outline

1. LTDF Template from Noisy Learning Problems

   (and why it fails from LPN)

2. Introducing Dense-Sparse LPN

3. Cryptanalysis & Open Questions

Summary of Cryptanalysis

$$(A, s \cdot A + e) \approx_c (A, u)$$

$$\left(\boxed{A \leftarrow \mathscr{D}\mathcal{S}(\mathbb{F}_2, n, m, k)} \,,\, \boxed{s \xleftarrow{R} \mathbb{F}_2^{1\times n}} \cdot \boxed{A} + \boxed{e \leftarrow \mathsf{Ber}(\epsilon)^{1\times m}} \right) \approx_c \left(\boxed{A \leftarrow \mathscr{D}\mathcal{S}(\mathbb{F}_2, n, m, k)} \,,\, \boxed{u \xleftarrow{R} \mathscr{R}^{1\times m}} \right)$$

Summary of Cryptanalysis

$$(A, s \cdot A + e) \approx_c (A, u)$$

$$\left(\boxed{A \leftarrow \mathscr{D}\mathscr{S}(\mathbb{F}_2, n, m, k)}\ ,\ \boxed{s \xleftarrow{R} \mathbb{F}_2^{1\times n}} \cdot \boxed{A} + \boxed{e \leftarrow \mathrm{Ber}(\epsilon)^{1\times m}} \right) \approx_c \left(\boxed{A \leftarrow \mathscr{D}\mathscr{S}(\mathbb{F}_2, n, m, k)}\ ,\ \boxed{u \xleftarrow{R} \mathscr{R}^{1\times m}} \right)$$

1. **Information Set Decoding: guess error coordinates of e $\implies$ time $2^{\tilde{\Omega}(\epsilon \cdot n)}$**

2. **Find a sparse vector x in the (right) kernel of $A = T \cdot M$ $\implies$ time $2^{\tilde{\Omega}(n^{\delta})}$**

   (inherited from M such that $M \cdot x = 0$)

   Parameter: $m = n^{1+\left(\frac{k}{2}-1\right)(1-\delta)}$

3. **Decompose Dense-Sparse matrix $\implies$ time $2^{\tilde{\Omega}(n)}$**

Summary of Cryptanalysis

$$(A, s \cdot A + e) \approx_c (A, u)$$

$$\left(\boxed{A \leftarrow \mathscr{DS}(\mathbb{F}_2, n, m, k)}\ ,\ \boxed{s \xleftarrow{R} \mathbb{F}_2^{1\times n}} \cdot \boxed{A} + \boxed{e \leftarrow \mathsf{Ber}(\epsilon)^{1\times m}} \right) \approx_c \left(\boxed{A \leftarrow \mathscr{DS}(\mathbb{F}_2, n, m, k)}\ ,\ \boxed{u \xleftarrow{R} \mathscr{R}^{1\times m}} \right)$$

1. Information Set Decoding: guess error coordinates of e $\implies$ time $2^{\tilde{\Omega}(\epsilon\cdot n)}$

2. Find a sparse vector x in the (right) kernel of $A = T \cdot M$ $\implies$ time $2^{\tilde{\Omega}(n^\delta)}$

   (inherited from M such that $M \cdot x = 0$) Parameter: $m = n^{1+\left(\frac{k}{2}-1\right)(1-\delta)}$

3. Decompose Dense-Sparse matrix $\implies$ time $2^{\tilde{\Omega}(n)}$

Conjectured Security: secure against attackers w/ time $\ll 2^{\min\left(\tilde{O}(\epsilon\cdot n),\, \tilde{O}(n^\delta)\right)}$

Decomposing a Dense-Sparse Matrix

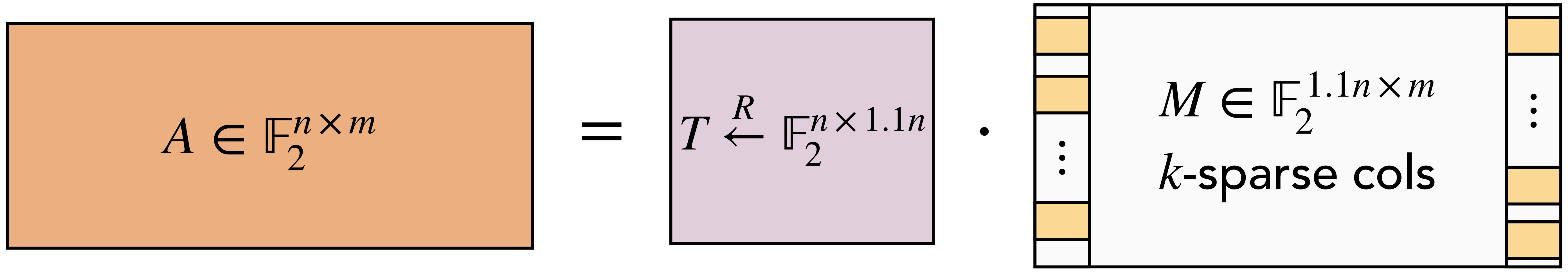
Decomposing a Dense-Sparse Matrix

$A \in \mathbb{F}_2^{n \times m}$ $=$ $T \xleftarrow{R} \mathbb{F}_2^{n \times 1.1n}$ $\cdot$ $M \in \mathbb{F}_2^{1.1n \times m}$ k-sparse cols

Find T, M from $A = T \cdot M \implies$ break DS-LPN with compression parameters!

(using our earlier Sparse LPN attack on M)

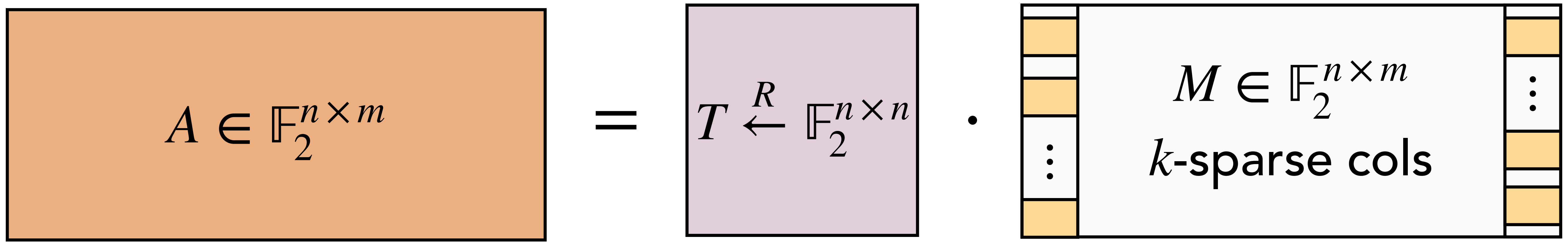
Decomposing a Dense-Sparse Matrix



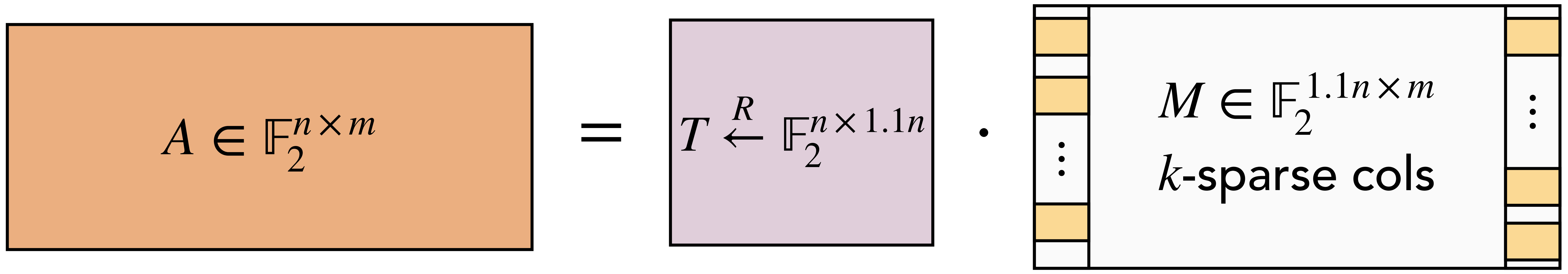
Find T, M from $A = T \cdot M \implies$ break DS-LPN with compression parameters!

(using our earlier Sparse LPN attack on M)

Why do we need $c > 1$? Suppose not...



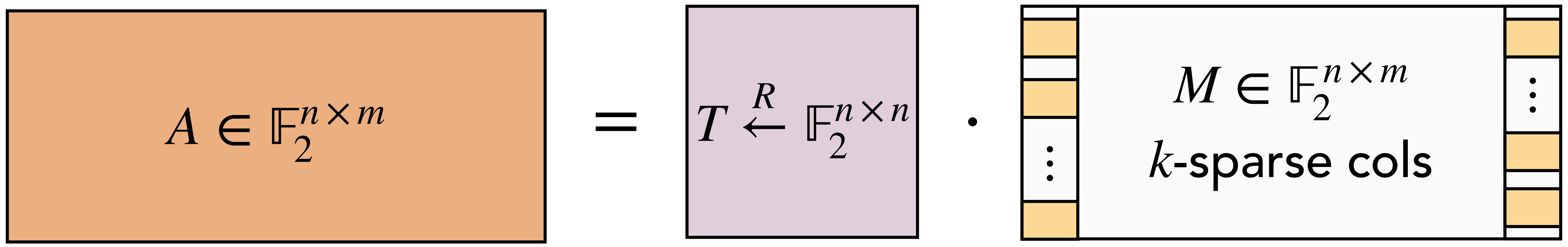
Decomposing a Dense-Sparse Matrix



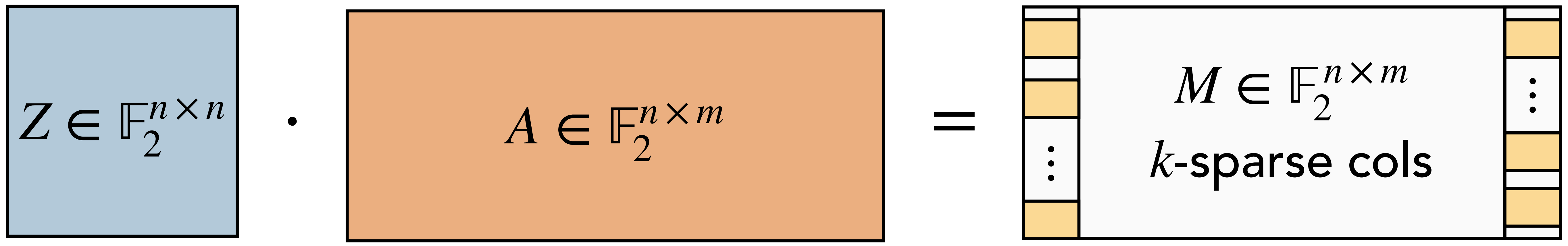
Find T, M from $A = T \cdot M \implies$ break DS-LPN with compression parameters!

(using our earlier Sparse LPN attack on M)

Why do we need $c > 1$? Suppose not...



This is <u>insecure</u>! We can find Z such that $Z \cdot A$ is sparse



Decomposing a Dense-Sparse Matrix

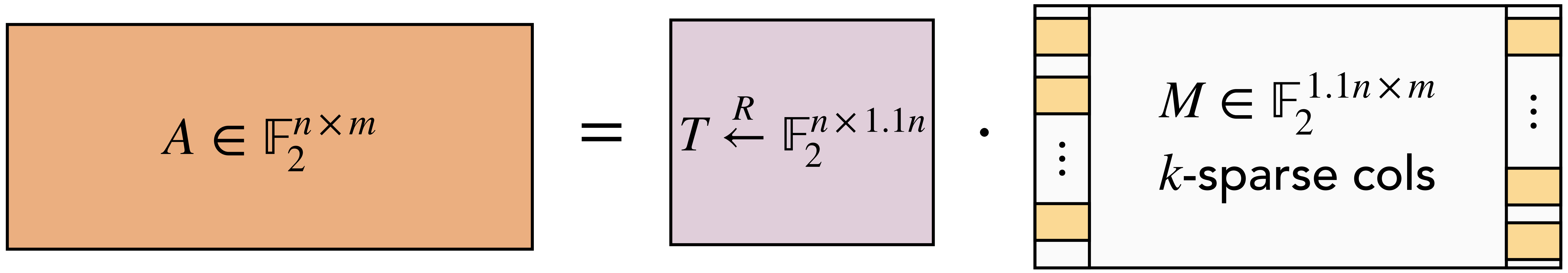
$$A \in \mathbb{F}_2^{n\times m} = T \xleftarrow{R} \mathbb{F}_2^{n\times 1.1n} \cdot \underbrace{M \in \mathbb{F}_2^{1.1n\times m}}_{k\text{-sparse cols}}$$

$$a_{1,j} \cdot z_1 + a_{2,j} \cdot z_2 + \cdots + a_{n,j} \cdot z_n = m_j \qquad \forall j \in [m]$$

This is <u>insecure</u>! We can find Z such that $Z \cdot A$ is sparse

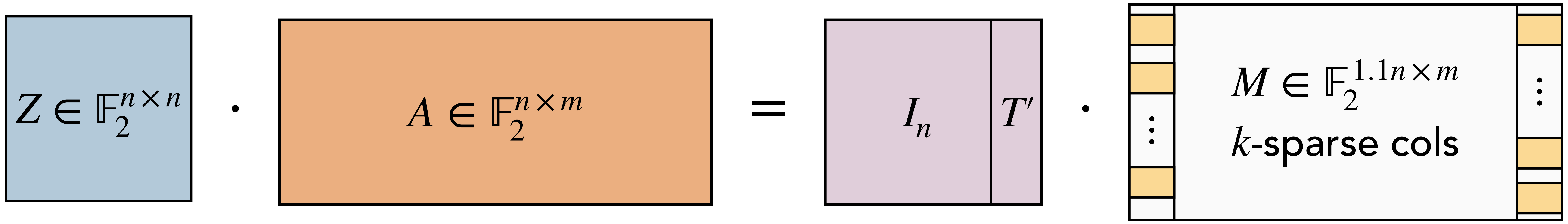
$$Z \in \mathbb{F}_2^{n\times n} \cdot A \in \mathbb{F}_2^{n\times m} = \underbrace{M \in \mathbb{F}_2^{n\times m}}_{k\text{-sparse cols}}$$

Decomposing a Dense-Sparse Matrix



The attack breaks down when $T \in \mathbb{F}_2^{n\times 1.1n}$ is rectangular

$\Longrightarrow$ time complexity is now $2^{\tilde{\Omega}(n)}$ due to guessing T'



Summary & Open Problems

Our Result: We introduce a **new** code-based assumption, Dense-Sparse LPN, and show how it gives rise to Lossy Trapdoor Functions.

Summary & Open Problems

<u>**Our Result:**</u> We introduce a **new** code-based assumption, **Dense-Sparse LPN**, and show how it gives rise to **Lossy Trapdoor Functions**.

<u>**Future Directions:**</u>

- <u>**Cryptanalysis:**</u>
  - <u>**Reductions:**</u> search-to-decision? worst-to-average-case?
  - <u>**Concrete parameters:**</u> we need help!
- <u>**Applications:**</u> PIR? Laconic OT? NIZK? IBE? ABE?
- <u>**Coding Theory:**</u> better constant-sparse matrix distributions?

Read our paper!
(ePrint 2024/175)

Thank you! Questions?