Statistical decoding 2.0: Reducing decoding to LPN (RLPN)

Kévin Carrier [1], Thomas Debris-Alazard [2], Charles Meyer-Hilfiger [3], Jean-Pierre Tillich [3]

[1]CY Cergy Paris Université, [2]Inria Paris-Saclay, [3]Inria Paris

Asiacrypt 2022

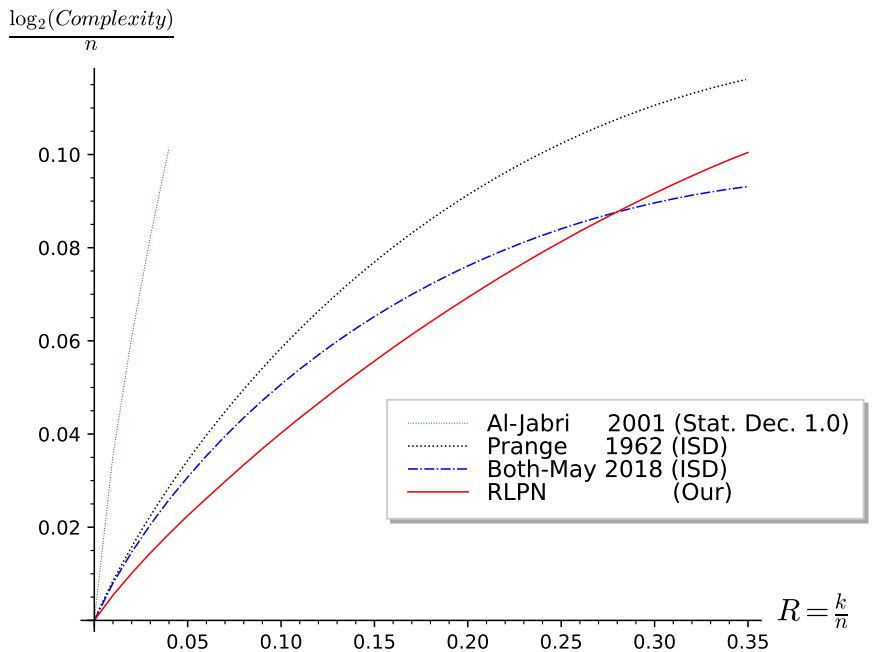


Table of Contents

Decoding problem and code-based cryptography

Security of all code-based primitives $\rightarrow$ Hardness of decoding linear codes

$\Downarrow$

Code-based primitives

- **PKE, KEM (NIST):** McEliece, BIKE, HQC
- **Signatures:** Stern + Fiat-Shamir, CFS, Wave

Complexity of the best decoding algorithm: tool for setting up parameters

Decoding problem

Linear code

A binary linear code $\mathscr{C}$ of length n and dimension k is a linear subspace of $\mathbb{F}_2^n$ of dimension k.

$$\rightarrow \text{Rate } R = \frac{k}{n}$$

Decoding at distance t

Given $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in \mathscr{C}$ and $|\mathbf{e}| = t$.

Find $\mathbf{c} \in \mathscr{C}$ s.t $|\mathbf{c} - \mathbf{y}| = t$.

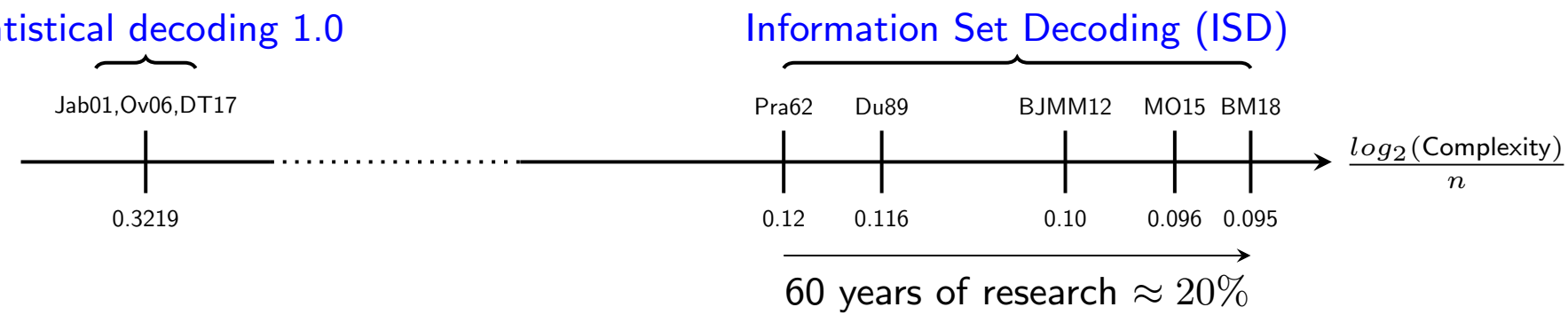
$|\mathbf{e}|$ is Hamming weight of $\mathbf{e}$: number of non-zero coordinates.

State of the art

Setting: Decoding at hardest distance $t = d_{GV} = n\, h_2^{-1}\,(1 - R)$

$(h_2(x) = -x \log_2(x) - (1 - x) \log_2(1 - x))$

For hardest rate:



Statistical decoding is uncompetitive against ISD's

Our contribution

New algorithm (RLPN): Based on statistical decoding 1.0

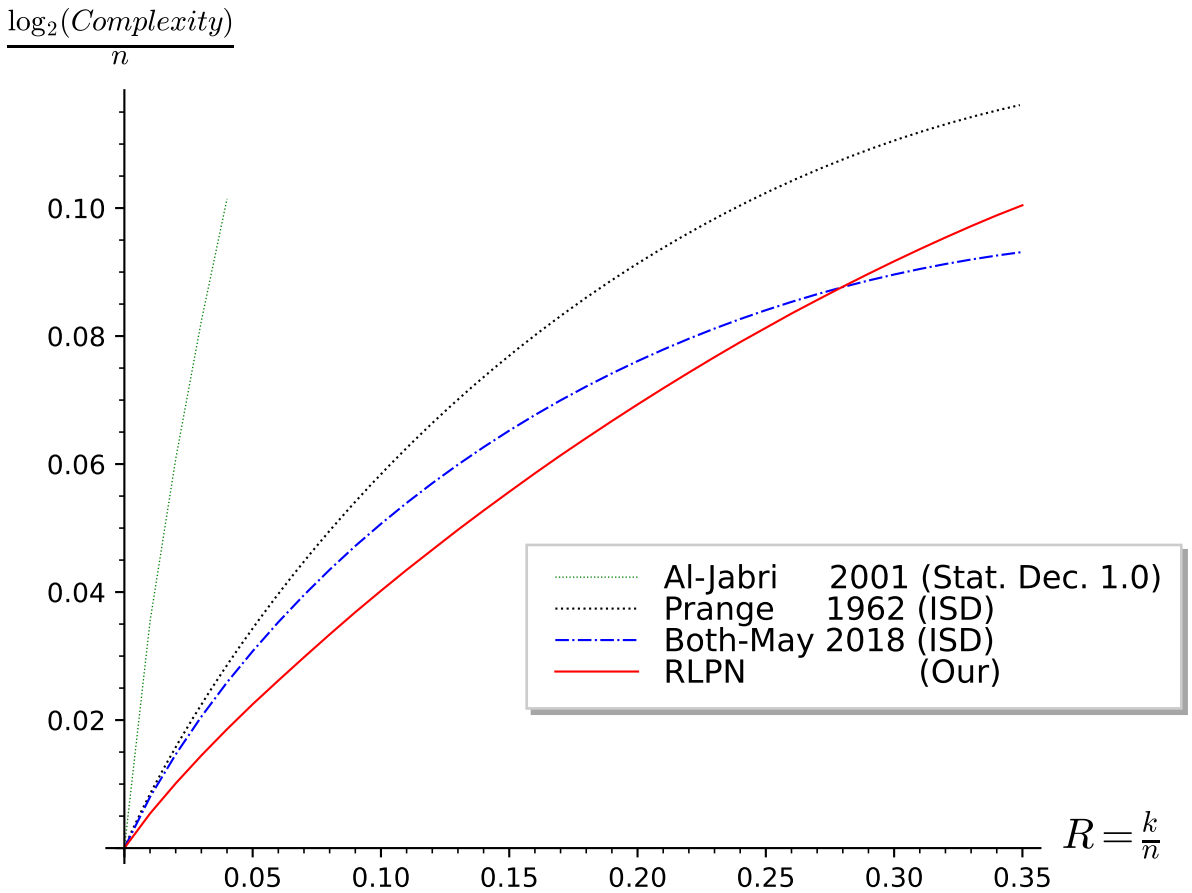
(Stat. 1.0) $\rightarrow$ Use parity checks to decode one bit

(RLPN) $\rightarrow$ Use parity checks to reduce to LPN then use LPN solver

$\rightarrow$ Groundbreaking improvement over Statistical decoding 1.0

Results

- $R \approx 0.02 \quad \rightarrow \quad \text{RLPN} \approx \sqrt{2^{Rn}} \quad \text{vs} \quad \text{ISD} \approx 2^{Rn}$
- $R < 0.3 \quad \rightarrow \quad$ RLPN better than all ISD



First time in 60 years that ISD's are beaten for a significant range.

Table of Contents

Dual code

$\mathscr{C}$ binary linear code of length n and dimension k

Dual of $\mathscr{C}$

$$\mathscr{C}^{\perp} \stackrel{\triangle}{=} \{\mathbf{h} \in \mathbb{F}_2^n \ : \langle \mathbf{c}, \mathbf{h} \rangle = 0 \text{ for all } \mathbf{c} \in \mathscr{C}\}$$

$\rightarrow \mathscr{C}^{\perp}$ code of length n and dimension $n-k$

$$\langle \mathbf{y}, \mathbf{h} \rangle = \underbrace{\langle \mathbf{c}, \mathbf{h} \rangle}_{=0} + \langle \mathbf{e}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle \qquad (\mathbf{y} = \mathbf{c} + \mathbf{e})$$

Statistical decoding 1.0 : original idea by Al Jabri (2001)

Idea: Recover 1 coordinate of $\mathbf{e} \to \mathbf{e}_1$

Let $\mathbf{h} \in \mathscr{C}^\perp$ s.t $\mathbf{h}_1 = 1$:

$$\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle = \sum_{i=1}^{n} \mathbf{e}_i \mathbf{h}_i = \mathbf{e}_1 + \underbrace{\sum_{i=2}^{n} \mathbf{e}_i \mathbf{h}_i}_{\text{noise}}$$

Recover $\mathbf{e}_1$ by majority voting from many samples $\langle \mathbf{y}, \mathbf{h} \rangle$ with $\mathbf{h} \in \mathscr{C}^\perp$

- If $\mathbf{e}_1 = 0 \to \langle \mathbf{y}, \mathbf{h} \rangle = \sum_{i=2}^{n} \mathbf{e}_i \mathbf{h}_i \qquad \sim \text{Bern}\left(\frac{1-\varepsilon}{2}\right)$
- If $\mathbf{e}_1 = 1 \to \langle \mathbf{y}, \mathbf{h} \rangle = 1 + \sum_{i=2}^{n} \mathbf{e}_i \mathbf{h}_i \quad \sim \text{Bern}\left(\frac{1+\varepsilon}{2}\right)$

Need $\widetilde{O}\left(\frac{1}{\varepsilon^2}\right)$ samples to distinguish distribution

$\to \varepsilon$ exponentially small, function of $n, t, w = |\mathbf{h}|$

Number of needed parity-checks $\mathbf{h}$ (function of $|\mathbf{h}| = w$)

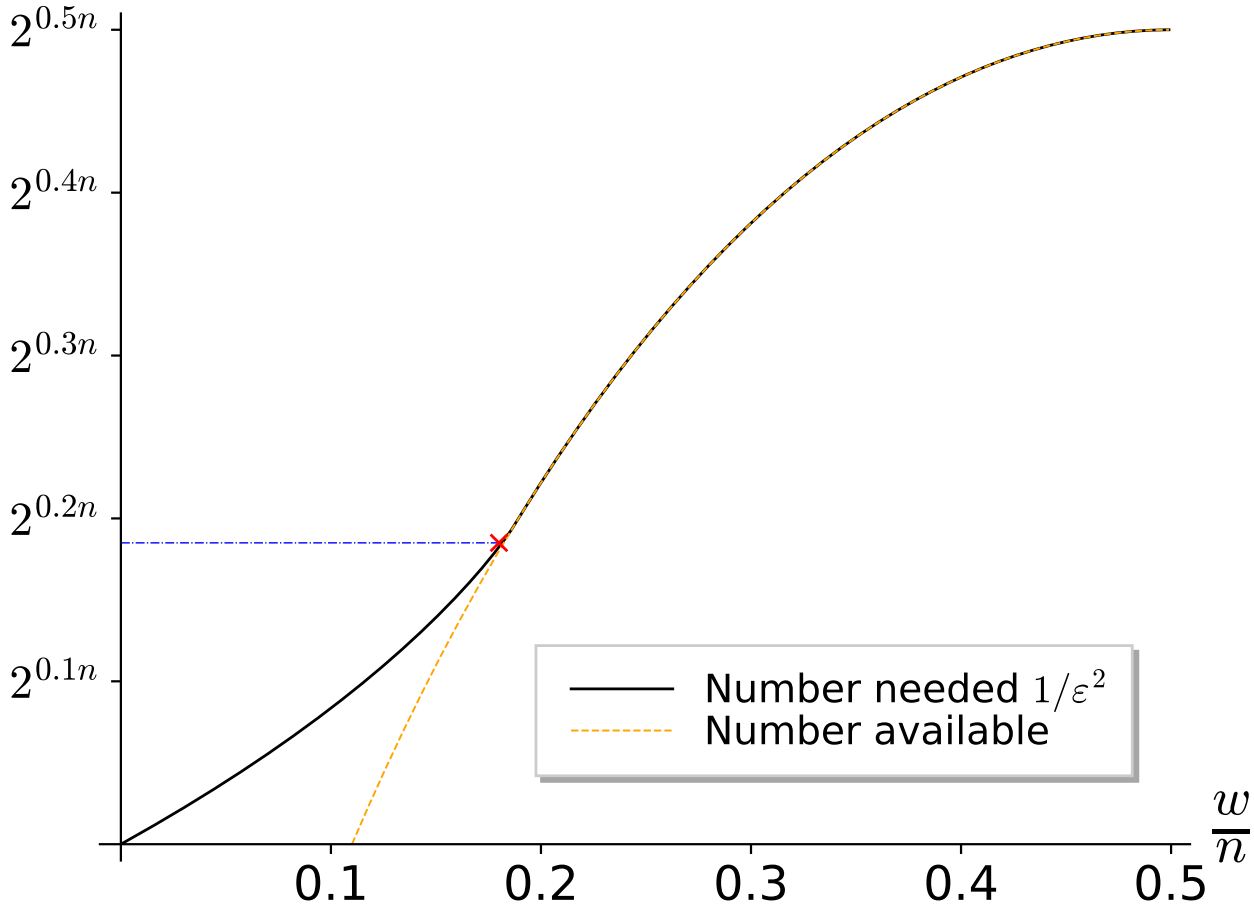


Figure: Number of needed samples $1/\varepsilon^2$ where $\frac{t}{n} = \delta_{GV}(R)$ and $R = \frac{1}{2}$

Table of Contents

Idea of RLPN: Recover a bigger part of $\mathbf{e}$

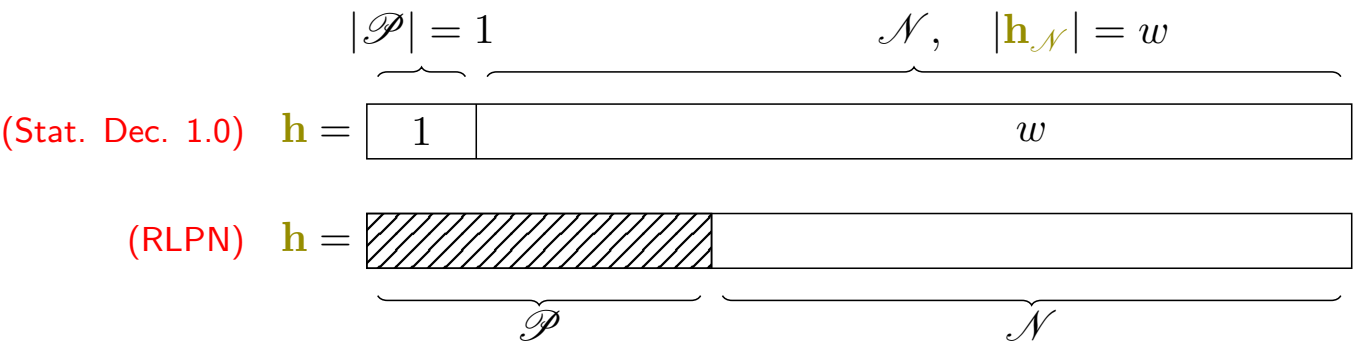
Choose a set of position $\mathscr{P}$ of size s, $\mathscr{N}$ complementary of $\mathscr{P}$

Recover $\mathbf{e}_{\mathscr{P}}$?

$$\downarrow$$

$$\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \underbrace{\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle}_{\text{noise}} \qquad (\mathbf{h} \in \mathscr{C}^{\perp})$$

Recover secret $\mathbf{e}_{\mathscr{P}}$ from many samples $\langle \mathbf{y}, \mathbf{h} \rangle$ given by $\mathbf{h} \in \mathscr{C}^{\perp}$



$\mathscr{P}$ bigger $\Rightarrow$ $\mathscr{N}$ smaller $\Rightarrow$ Noise becomes smaller!

Idea of RLPN: Recover a bigger part of $\mathbf{e}$

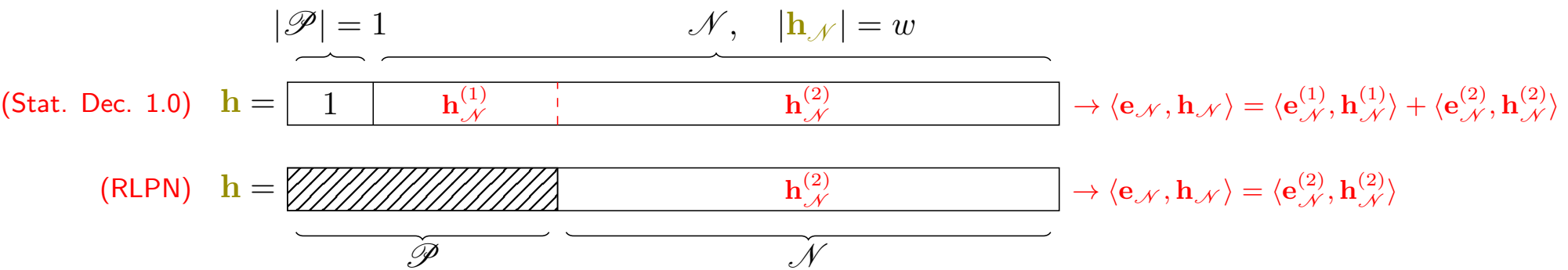
Choose a set of position $\mathscr{P}$ of size s, $\mathscr{N}$ complementary of $\mathscr{P}$

Recover $\mathbf{e}_{\mathscr{P}}$?

$$\downarrow$$

$$\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \underbrace{\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle}_{\text{noise}} \qquad (\mathbf{h} \in \mathscr{C}^{\perp})$$

Recover secret $\mathbf{e}_{\mathscr{P}}$ from many samples $\langle \mathbf{y}, \mathbf{h} \rangle$ given by $\mathbf{h} \in \mathscr{C}^{\perp}$



$\mathscr{P}$ bigger $\Rightarrow$ $\mathscr{N}$ smaller $\Rightarrow$ Noise becomes smaller!

Number of needed parity checks $\mathbf{h}$

When $\mathbb{P}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle = 1\right) = \frac{1-\varepsilon}{2}$

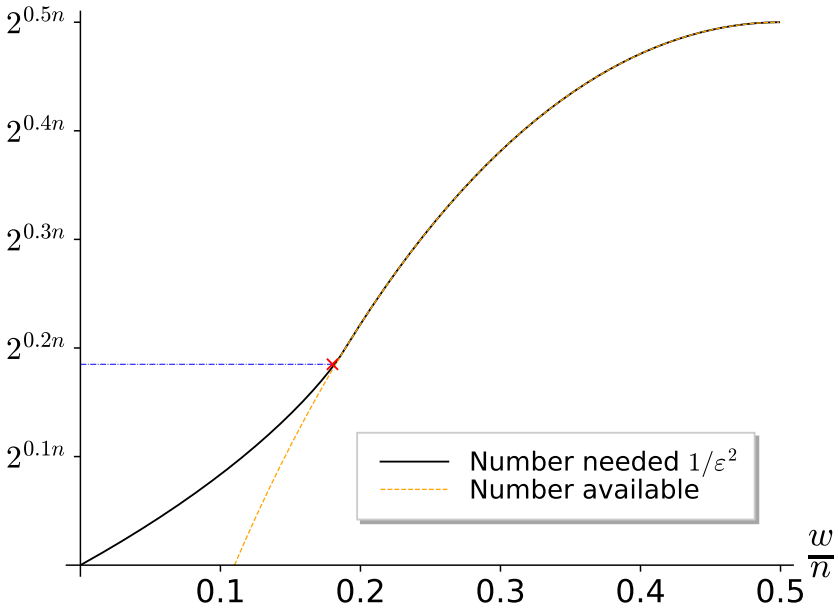
Information theory $\rightarrow \frac{1}{\varepsilon^2}$ sample needed to recover $\mathbf{e}_{\mathscr{P}}$

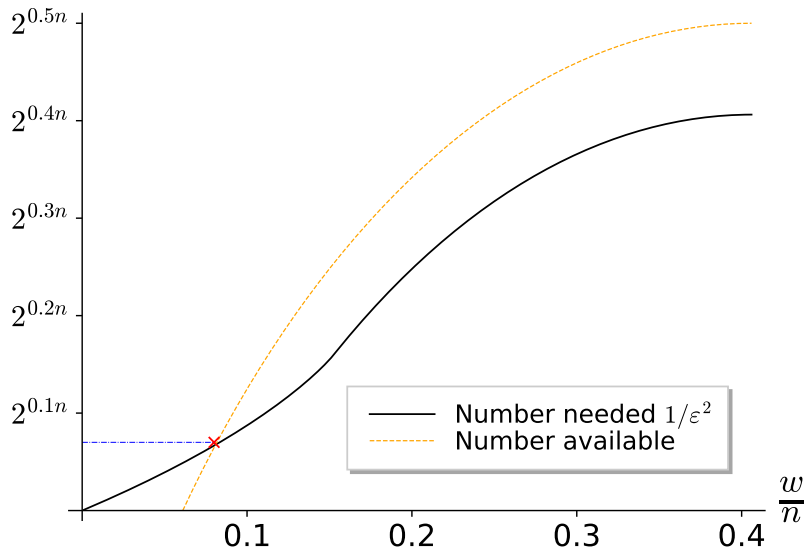
Stat. Dec. 1.0

$|\mathscr{P}| = 1$, $|\mathscr{N}| =$ n-1

RLPN

$|\mathscr{P}| = s$, $|\mathscr{N}| = n - s$





Recover secret

- When $|\mathscr{P}| = 1 \rightarrow$ Majority voting to recover $\mathbf{e}_{\mathscr{P}}$ (Stat. Dec. 1.0)
- When $|\mathscr{P}| = s \rightarrow$ How to recover $\mathbf{e}_{\mathscr{P}}$?

Reducing decoding to LPN problem

$$\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$$

LPN Samples

$$(\mathbf{a}, \langle \mathbf{s}, \mathbf{a} \rangle + e) \text{ where } \begin{cases} \mathbf{s} \stackrel{\triangle}{=} \mathbf{e}_{\mathscr{P}} & \text{(secret)} \\ \mathbf{a} \stackrel{\triangle}{=} \mathbf{h}_{\mathscr{P}} & \text{(known)} \\ e \stackrel{\triangle}{=} \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle \sim Bern(\frac{1-\varepsilon}{2}) & \text{(noise)} \end{cases}$$

Compute N vectors $\mathbf{h} \in \mathscr{C}^{\perp} \Rightarrow$ Get N LPN samples

Naive search to solve LPN

Exhaustive search over $\mathbf{e}_{\mathscr{P}} \in \mathbb{F}_2^s$:

$\rightarrow$ Find $\mathbf{x} \in \mathbb{F}_2^s$ s.t $\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle$ the most often

Complexity: $N2^s$

Speed-up with Fast Fourier Transform

Complexity: $s2^s$

RLPN Algorithm

RLPN

1. Choose complementary set of positions $\mathscr{P}$ and $\mathscr{N}$
2. Compute $N = \frac{1}{\varepsilon^2}$ vectors $\mathbf{h} \in \mathscr{C}^{\perp}$ s.t $|\mathbf{h}_{\mathscr{N}}| = w$ (using ISD)
3. Recover $\mathbf{e}_{\mathscr{P}}$ with Fast Fourier Transform

Complexity

$$T_{RLPN} = T_N + T_{FFT}$$

- $T_N \rightarrow$ Cost of computing $N = \frac{1}{\varepsilon^2}$ vectors $\mathbf{h}$
- $T_{FFT} = s2^s \rightarrow$ Cost of Fast Fourier transform

Table of Contents

Create parity checks of low weight w on $\mathscr{N}$? Dumer !

$$\mathscr{C}_{\mathscr{N}}^{\perp} = \underbrace{\{\mathbf{h}_{\mathscr{N}} : \mathbf{h} \in \mathscr{C}^{\perp}\}}_{\text{code: len. } n-s\text{, dim. } n-k} = \{\mathbf{h}' \in \mathbb{F}_2^{n-s} : \mathbf{H}'\mathbf{h}' = \mathbf{0}\},\ \mathbf{H}' \in \mathbb{F}_2^{(k-s)\times(n-s)}$$

Dumer idea: Collision

Split $\mathbf{H}'$ in two equal parts: $\mathbf{H}' = (\mathbf{H}'_1 \mid \mathbf{H}'_2)$

If $\mathbf{h}'_1, \mathbf{h}'_2$ of weight $\frac{w}{2}$ s.t $\mathbf{H}'_1\mathbf{h}'_1 = \mathbf{H}'_2\mathbf{h}'_2$

$\Downarrow$

$\mathbf{h}' = (\mathbf{h}'_1||\mathbf{h}'_2)$ of weight w s.t $\mathbf{H}'\mathbf{h}' = \mathbf{0}$

$$\rightarrow T_N^{(Dumer86)} = \underbrace{\binom{\frac{n-s}{2}}{\frac{w}{2}}}_{\text{Cost enumeration}} + \underbrace{\frac{\binom{\frac{n-s}{2}}{\frac{w}{2}}^2}{2^{k-s}}}_{\text{Number equations produced}}$$

Complexity of RLPN when R small

Choose $\binom{\frac{n-s}{2}}{\frac{w}{2}} \approx 2^{k-s}$ and $s \approx \frac{k}{2} \Rightarrow T_{RLPN} = O\left(\underbrace{2^{\frac{k}{2}}}_{FFT} + \underbrace{2^{\frac{k}{2}}}_{Dumer}\right) = O\left(\sqrt{2^{Rn}}\right)$

Table of Contents

To conclude

- First algorithm in 60 years beat ISD for a significant range: $R < 0.3$
- RLPN uses more advanced techniques to compute parity checks [BJMM12]
- Similar techniques are used in dual attacks in lattices

ISD

$\rightarrow$ Compute low weight codewords

- $R \approx 0 \rightarrow T \approx$ Naive
- $R \approx 1 \rightarrow T \approx \sqrt{\text{Naive}}$

RLPN

$\rightarrow$ Compute low weight parity-checks

- $R \approx 0.02 \rightarrow T \approx \sqrt{\text{Naive}}$
- $R \approx 1 \quad \rightarrow T \approx$ Naive

RLPN "dualize" the ISD