Decoding challenge

Assessing the practical hardness of syndrome decoding for code-based cryptography

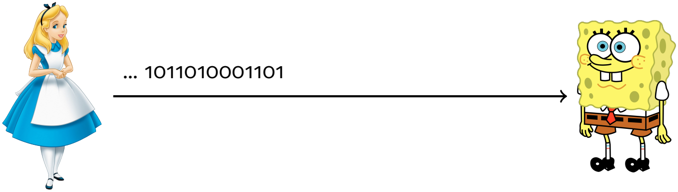
Matthieu Lequesne

Sorbonne Université
Inria Paris, team Cosmiq

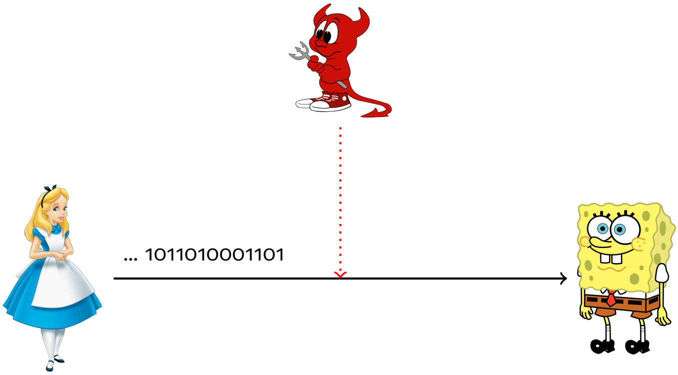
February 27, 2020

All you ever wanted to know about code-based crypto

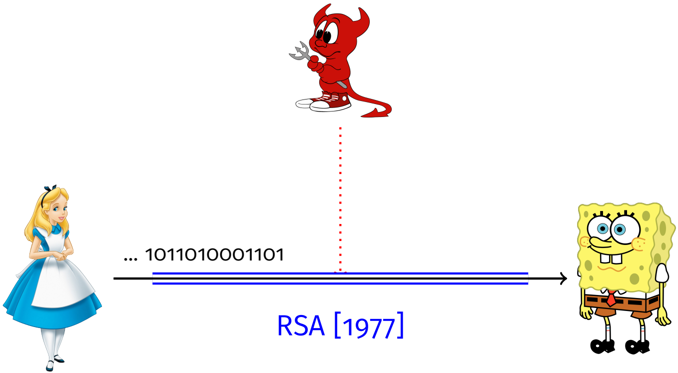
Public Key Cryptography



Public Key Cryptography



Public Key Cryptography



[1994]

+

+

$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

... 1011010001101

~~RSA [1977]~~

???

[1994]

+  +

$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$$

... 1011010001101

~~RSA [1977]~~

???

Post-Quantum Cryptography

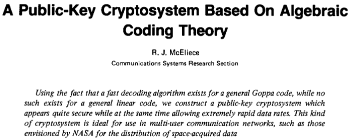
Post-Quantum Cryptography

Lattice | <u>Codes</u> | Hash | Multivariate | Isogenies

Post-Quantum Cryptography

Lattice | Codes | Hash | Multivariate | Isogenies

1978, Robert McEliece: [McE78]


A Public-Key Cryptosystem Based On Algebraic Coding Theory

R. J. McEliece
Communications Systems Research Section

Using the fact that a fast decoding algorithm exists for a general Goppa code, while no such exists for a general linear code, we construct a public-key cryptosystem which appears quite secure while at the same time allowing extremely rapid data rates. This kind of cryptosystem is ideal for use in multi-user communication networks, such as those envisioned by NASA for the distribution of space-acquired data

Definition (Code)

An $[n,k]_{\mathbb{F}_q}$ linear **code** $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$ of dimension k.

Definition (Decoder)

A **decoder** for the code $\mathcal{C}$ is a function

$$\Phi_{\mathcal{C}} : \mathbb{F}_q^n \to \mathcal{C} \cup \{?\}.$$

We say that $\Phi_{\mathcal{C}}$ can decode up to t errors if

$$\forall c \in \mathcal{C}, \forall e \in \mathbb{F}_q^n, \qquad |e| \leq t \quad \Rightarrow \quad \Phi_{\mathcal{C}}(c+e) = c.$$

Definition (Generator matrix)

A **generator matrix** of a code $\mathcal{C}$ is a matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ such that:

$$\mathcal{C} = \{x\mathbf{G} \mid x \in \mathbb{F}_q^k\}.$$

Definition (Parity-check matrix)

A **parity-check matrix** of a code $\mathcal{C}$ is a matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ such that:

$$\mathcal{C} = \{y \in \mathbb{F}_q^n \mid \mathbf{H}y^\intercal = 0\}.$$

Example (Repetition Code)

$$\begin{array}{ccc} \mathbb{F}_2 & \to & \mathbb{F}_2^3 \\ 0 & \mapsto & (0,0,0) \\ 1 & \mapsto & (1,1,1) \end{array}$$

Example (Decoder)

```
if |x| <= 1:
    return 0
else:
    return 1
```

Example (Repetition Code)

$$\begin{array}{ccc} \mathbb{F}_2 & \to & \mathbb{F}_2^3 \\ 0 & \mapsto & (0,0,0) \\ 1 & \mapsto & (1,1,1) \end{array}$$

Example (Decoder)

```
if |x| <= 1:
    return 0
else:
    return 1
```

$$G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Main idea: how hard is it to decode up to t errors?

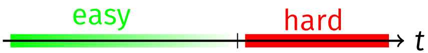
Main idea: how hard is it to decode up to t errors?

- For a random code

Main idea: how hard is it to decode up to t errors?

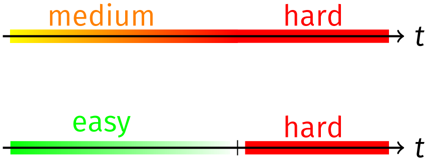
- For a random code

- For some special families of structured codes



Main idea: how hard is it to decode up to t errors?

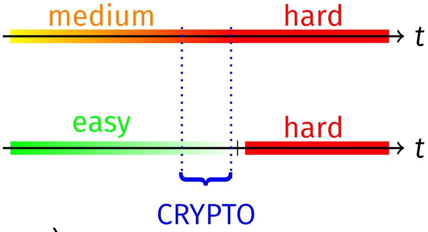
- For a random code
- For some special families of structured codes



easy = in polynomial time (with trap)
medium / hard = requires exponential time

Main idea: how hard is it to decode up to t errors?

- For a random code
- For some special families of structured codes



easy = in polynomial time (with trap)
medium / hard = requires exponential time

How to design a code-based cryptosystem?

Ingredients:

- a family $\mathcal{F}$ of structured codes;

Ingredients:

- a family $\mathcal{F}$ of structured codes;
- a decoder $\Phi_{\mathcal{F}}$ that can correct efficiently up to t errors;

Ingredients:

- a family $\mathcal{F}$ of structured codes;
- a decoder $\Phi_{\mathcal{F}}$ that can correct efficiently up to t errors;
- a shaker!

Ingredients:

- a family $\mathcal{F}$ of structured codes;
- a decoder $\Phi_{\mathcal{F}}$ that can correct efficiently up to t errors;
- a shaker!

Receipe:

KeyGen()	Enc(m)	Dec(c)
$\mathbf{G}_{\mathsf{sk}} \stackrel{\$}{\leftarrow} \mathcal{F}$ $\mathbf{G}_{\mathsf{pk}} \leftarrow \mathrm{Shake}(\mathbf{G}_{\mathsf{sk}})$	$e \stackrel{\$}{\leftarrow} \mathbb{F}_q^n$, s.t. $\lvert e\rvert = t$ $c \leftarrow m\mathbf{G}_{\mathsf{pk}} + e$	$m \leftarrow \Phi_{\mathcal{F}}(\mathbf{G}_{\mathsf{sk}}, c)$

Ingredients:

- a family $\mathcal{F}$ of structured codes;
- a decoder $\Phi_{\mathcal{F}}$ that can correct efficiently up to t errors;
- a shaker!

Receipe:

KeyGen()	Enc(m)	Dec(c)
$\mathbf{G}_{\text{sk}} \xleftarrow{\$} \mathcal{F}$ $\mathbf{G}_{\text{pk}} \leftarrow \mathrm{Shake}(\mathbf{G}_{\text{sk}})$	$e \xleftarrow{\$} \mathbb{F}_q^n$, s.t. $\lvert e \rvert = t$ $c \leftarrow m\mathbf{G}_{\text{pk}} + e$	$m \leftarrow \Phi_{\mathcal{F}}(\mathbf{G}_{\text{sk}}, c)$

The key to success:

- choose t s.t. it is hard to decode t errors for a random code;

Ingredients:

- a family $\mathcal{F}$ of structured codes;
- a decoder $\Phi_{\mathcal{F}}$ that can correct efficiently up to t errors;
- a shaker!

Receipe:

KeyGen()	Enc(m)	Dec(c)
$\mathbf{G}_{\text{sk}} \overset{\$}{\leftarrow} \mathcal{F}$ $\mathbf{G}_{\text{pk}} \leftarrow \mathrm{Shake}(\mathbf{G}_{\text{sk}})$	$e \overset{\$}{\leftarrow} \mathbb{F}_q^n$, s.t. $\lvert e \rvert = t$ $c \leftarrow m\mathbf{G}_{\text{pk}} + e$	$m \leftarrow \Phi_{\mathcal{F}}(\mathbf{G}_{\text{sk}}, c)$

The key to success:

- choose t s.t. it is hard to decode t errors for a random code;
- $\Phi_{\mathcal{F}}$ needs the structured version of the code to be efficient;

Ingredients:

- a family $\mathcal{F}$ of structured codes;
- a decoder $\Phi_{\mathcal{F}}$ that can correct efficiently up to t errors;
- a shaker!

Receipe:

KeyGen()	Enc(m)	Dec(c)
$\mathbf{G}_{\mathrm{sk}} \xleftarrow{\$} \mathcal{F}$ $\mathbf{G}_{\mathrm{pk}} \leftarrow \mathrm{Shake}(\mathbf{G}_{\mathrm{sk}})$	$e \xleftarrow{\$} \mathbb{F}_q^n$, s.t. $\lvert e \rvert = t$ $c \leftarrow m\mathbf{G}_{\mathrm{pk}} + e$	$m \leftarrow \Phi_{\mathcal{F}}(\mathbf{G}_{\mathrm{sk}}, c)$

The key to success:

- choose t s.t. it is hard to decode t errors for a random code;
- $\Phi_{\mathcal{F}}$ needs the structured version of the code to be efficient;
- the shaker shakes well enough!

How could Eve break this scheme? 2 possibilities:

1. Reconstruct $\mathbf{G}_{\mathrm{sk}}$ from $\mathbf{G}_{\mathrm{pk}}$ and then use $\Phi_{\mathcal{F}}$ to decode.

How could Eve break this scheme? 2 possibilities:

1. Reconstruct $\mathbf{G}_{\mathrm{sk}}$ from $\mathbf{G}_{\mathrm{pk}}$ and then use $\Phi_{\mathcal{F}}$ to decode.

Security hypothesis 1

$\mathbf{G}_{\mathrm{pk}}$ is indistinguishable from a random $k \times n$ matrix.

How could Eve break this scheme? 2 possibilities:

1. Reconstruct $\mathbf{G}_{\mathrm{sk}}$ from $\mathbf{G}_{\mathrm{pk}}$ and then use $\Phi_{\mathcal{F}}$ to decode.

Security hypothesis 1

$\mathbf{G}_{\mathrm{pk}}$ is indistinguishable from a random $k \times n$ matrix.

2. Decode using $\mathbf{G}_{\mathrm{pk}}$.

How could Eve break this scheme? 2 possibilities:

1. Reconstruct $\mathbf{G}_{\mathrm{sk}}$ from $\mathbf{G}_{\mathrm{pk}}$ and then use $\Phi_{\mathcal{F}}$ to decode.

Security hypothesis 1

$\mathbf{G}_{\mathrm{pk}}$ is indistinguishable from a random $k \times n$ matrix.

2. Decode using $\mathbf{G}_{\mathrm{pk}}$.

Security hypothesis 2

Decoding t errors in a random $[n, k]$-code is hard.

How could Eve break this scheme? 2 possibilities:

1. Reconstruct $\mathbf{G}_{\mathrm{sk}}$ from $\mathbf{G}_{\mathrm{pk}}$ and then use $\Phi_{\mathcal{F}}$ to decode.

Security hypothesis 1

$\mathbf{G}_{\mathrm{pk}}$ is indistinguishable from a random $k \times n$ matrix.

2. Decode using $\mathbf{G}_{\mathrm{pk}}$.

Security hypothesis 2

Decoding t errors in a random $[n, k]$-code is hard.

Remark: Hypothesis 1 depends on the choice of the family of codes $\mathcal{F}$ and the shaker, while Hypothesis 2 is generic!

Some examples

- Examples of choices of $\mathcal{F}$:
  - Goppa codes [Original McEliece];
  - Reed Solomon codes [Nie86] (broken by [SS92]);
  - QC-MDPC codes [BIKE];
  - Rank-based codes [ROLLO].

- Examples of choices of $\mathcal{F}$:
  - Goppa codes [Original McEliece];
  - Reed Solomon codes [Nie86] (broken by [SS92]);
  - QC-MDPC codes [BIKE];
  - Rank-based codes [ROLLO].

- Examples of shakers:
  - row scrambler;
  - columns isometry (permutation);
  - subfield subcode;
  - adding random columns...

Syndrome Decoding

Let $\mathcal{C}$ be an $[n, k]$ linear code of parity-check matrix $\mathbf{H}$.
Let $y \in \mathbb{F}_q^n$ and $s = y\mathbf{H}^\intercal \in \mathbb{F}_q^k$ (the **syndrome** of y).
The following problems are equivalent.

Let $\mathcal{C}$ be an $[n, k]$ linear code of parity-check matrix $\mathbf{H}$.
Let $y \in \mathbb{F}_q^n$ and $s = y\mathbf{H}^\intercal \in \mathbb{F}_q^k$ (the **syndrome** of y).
The following problems are equivalent.

1. Find a codeword $x \in \mathcal{C}$ such that $|y - x| \leq t$.

Let $\mathcal{C}$ be an $[n, k]$ linear code of parity-check matrix $\mathbf{H}$.
Let $y \in \mathbb{F}_q^n$ and $s = y\mathbf{H}^\intercal \in \mathbb{F}_q^k$ (the **syndrome** of y).
The following problems are equivalent.

1. Find a codeword $x \in \mathcal{C}$ such that $|y - x| \leq t$.
2. Find an error $e \in y + \mathcal{C}$ such that $|e| \leq t$.

Let $\mathcal{C}$ be an $[n, k]$ linear code of parity-check matrix $\mathbf{H}$.
Let $y \in \mathbb{F}_q^n$ and $s = y\mathbf{H}^\intercal \in \mathbb{F}_q^k$ (the **syndrome** of y).
The following problems are equivalent.

1. Find a codeword $x \in \mathcal{C}$ such that $|y - x| \leq t$.
2. Find an error $e \in y + \mathcal{C}$ such that $|e| \leq t$.
3. Find an error e such that $e\mathbf{H}^\intercal = s$ and $|e| \leq t$.

Let $\mathcal{C}$ be an $[n, k]$ linear code of parity-check matrix $\mathbf{H}$.
Let $y \in \mathbb{F}_q^n$ and $s = y\mathbf{H}^\mathsf{T} \in \mathbb{F}_q^k$ (the **syndrome** of y).
The following problems are equivalent.

1. Find a codeword $x \in \mathcal{C}$ such that $|y - x| \leq t$.
2. Find an error $e \in y + \mathcal{C}$ such that $|e| \leq t$.
3. Find an error e such that $e\mathbf{H}^\mathsf{T} = s$ and $|e| \leq t$.

The Syndrome Decoding Problem - $\mathrm{SD}(q, n, R, W)$

Instance: $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$,
$s \in \mathbb{F}_q^{n-k}$.

Output: $e \in \mathbb{F}_q^n$ such that $|e| = w$ and $e\mathbf{H}^\mathsf{T} = s$,

where $k \triangleq \lceil Rn \rceil$, $w \triangleq \lceil Wn \rceil$.

Theorem (NP-completeness)

The Syndrome Decoding problem is NP-complete. [BMvT78]

Theorem (NP-completeness)

The Syndrome Decoding problem is NP-complete. [BMvT78]

Conjecture (average case)

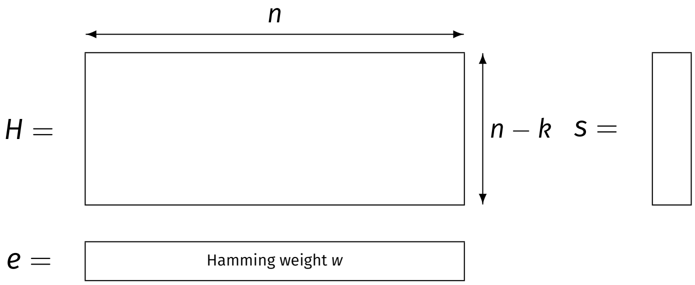
Decoding n^{ε} errors is hard on average $\forall \varepsilon > 0$. [Ale11]

Binary Syndrome Decoding Problem

From now on, we focus on the binary case $q = 2$.

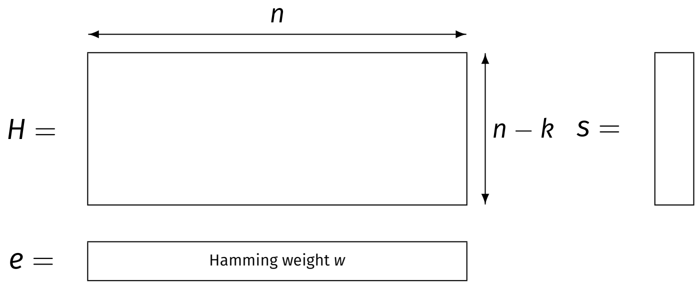
Binary Syndrome Decoding Problem

From now on, we focus on the binary case $q = 2$.



Find w columns of H adding to s

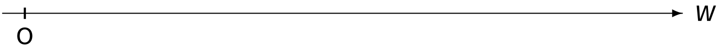
From now on, we focus on the binary case $q = 2$.



Find w columns of H adding to s

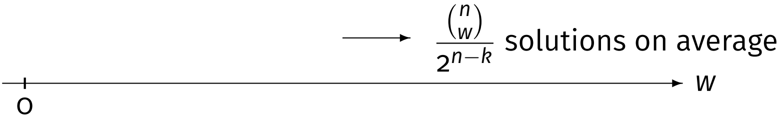
The next slides of this section are reproduced from Nicolas Sendrier's MOOC "Code Based Cryptography" with his authorization.

Fix n and k, let w grow:



Number of solutions

Fix n and k, let w grow:



Number of solutions

Fix n and k, let w grow:

at most one solution $\longleftarrow$ $\longrightarrow$ $\frac{\binom{n}{w}}{2^{n-k}}$ solutions on average

0 $\longrightarrow$ w

Fix n and k, let w grow:

at most one solution $\longleftarrow$ $\longrightarrow$ $\frac{\binom{n}{w}}{2^{n-k}}$ solutions on average

0 — d_{GV} — w

$d_{GV} \triangleq$ Gilbert-Varshamov radius, s.t. $\binom{n}{d_{GV}} = 2^{n-k}$.

Number of solutions

Fix n and k, let w grow:

exactly
~~at most~~ one solution $\longleftarrow$ $\longrightarrow$ $\frac{\binom{n}{w}}{2^{n-k}}$ solutions on average

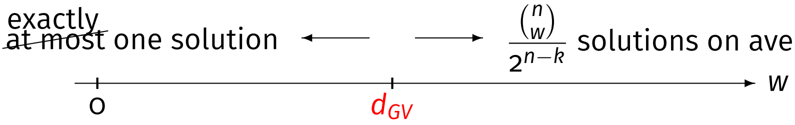
0 $\quad\quad$ d_{GV} $\quad\quad$ $\longrightarrow w$

$d_{GV} \triangleq$ Gilbert-Varshamov radius, s.t. $\binom{n}{d_{GV}} = 2^{n-k}$.

In cryptanalysis, we only consider situations where there is a solution.

Number of solutions

Fix n and k, let w grow:



$d_{GV} \triangleq$ Gilbert-Varshamov radius, s.t. $\binom{n}{d_{GV}} = 2^{n-k}$.

In cryptanalysis, we only consider situations where there is a solution.

We expect $\approx \max\left(1, \binom{n}{w}/2^{n-k}\right)$ solutions.

Problem:
find w columns of H adding to s (modulo 2)

$$H = \overbrace{\boxed{\begin{matrix} h_1 & h_2 & \cdots & h_n \end{matrix}}}^{n} \; n-k \qquad s = \boxed{}$$

Exhaustive search

Problem:
find w columns of H adding to s (modulo 2)

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \; n-k \qquad s = \begin{bmatrix} \; \end{bmatrix}$$

Enumerate all w-tuples $(j_1, j_2, \cdots, j_w)$ such that

$$1 \leq j_1 < j_2 < \ldots < j_w \leq n.$$

Check whether

$$s + h_{j_1} + h_{j_2} \cdots + h_{j_w} = 0.$$

Exhaustive search

Problem:
find w columns of H adding to s (modulo 2)

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \Big\updownarrow n-k \qquad s = \begin{bmatrix} \ \end{bmatrix}$$

Enumerate all w-tuples $(j_1, j_2, \cdots, j_w)$ such that

$$1 \leq j_1 < j_2 < \ldots < j_w \leq n.$$

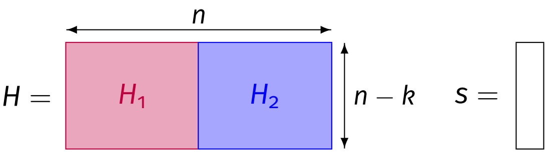
Check whether

$$s + h_{j_1} + h_{j_2} \cdots + h_{j_w} = 0.$$

Cost: *about* $\binom{n}{w}$ column operations.

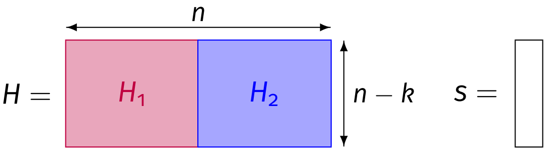
Remark: we obtain all solutions.

Problem:
find w columns of H adding to s (modulo 2)



BIRTHDAY ALGORITHM

Problem:
find w columns of H adding to s (modulo 2)

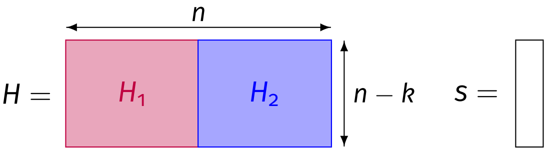


Idea: Split H into two equal parts and enumerate the two following sets

$$\mathcal{L}_1 = \left\{ e_1 H_1^T, |e_1| = \frac{w}{2} \right\} \text{ and } \mathcal{L}_2 = \left\{ s + e_2 H_2^T, |e_2| = \frac{w}{2} \right\}$$

If $\mathcal{L}_1 \cap \mathcal{L}_2 \neq \emptyset$, we have solution(s): $s + e_1 H_1^T + e_2 H_2^T = 0$

Problem:
find w columns of H adding to s (modulo 2)



Idea: Split H into two equal parts and enumerate the two following sets

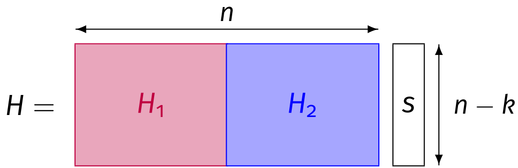
$$\mathcal{L}_1 = \left\{ e_1 H_1^T, |e_1| = \frac{w}{2} \right\} \text{ and } \mathcal{L}_2 = \left\{ s + e_2 H_2^T, |e_2| = \frac{w}{2} \right\}$$

If $\mathcal{L}_1 \cap \mathcal{L}_2 \neq \emptyset$, we have solution(s): $s + e_1 H_1^T + e_2 H_2^T = 0$

Cost: Requires *about* $2L + L^2/2^{n-k}$ column operations, where $L = |\mathcal{L}_1| = |\mathcal{L}_2|$

Birthday algorithm

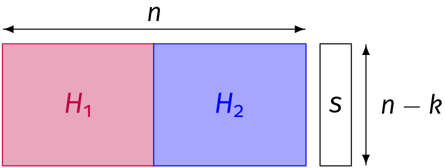
$$\boxed{\text{Compute } \mathcal{L}_1 \cap \mathcal{L}_2 = \left\{ e_1 H_1^T \mid |(|e_1) = \tfrac{w}{2} \right\} \cap \left\{ s + e_2 H_2^T \mid |(|e_2) = \tfrac{w}{2} \right\}}$$



$$\boxed{\text{Compute } \mathcal{L}_1 \cap \mathcal{L}_2 = \left\{ e_1 H_1^T \mid |(|e_1) = \tfrac{w}{2} \right\} \cap \left\{ s + e_2 H_2^T \mid |(|e_2) = \tfrac{w}{2} \right\}}$$

`for all` e_1 of weight $w/2$

$\quad x \leftarrow e_1 H_1^T \; ; T[x] \leftarrow T[x] \cup \{e_1\}$

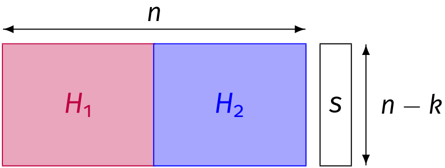
$H =$ 

Total cost: $\binom{n/2}{w/2}$

$|\mathcal{L}_1|$

$$\boxed{\text{Compute } \mathcal{L}_1 \cap \mathcal{L}_2 = \{e_1H_1^T \mid |(|e_1) = \tfrac{w}{2}\} \cap \{s + e_2H_2^T \mid |(|e_2) = \tfrac{w}{2}\}}$$

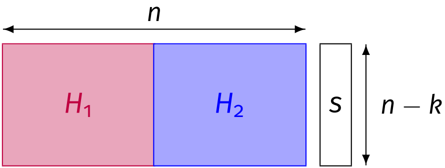
`for all` e_1 of weight $w/2$
  $x \leftarrow e_1H_1^T$; $T[x] \leftarrow T[x] \cup \{e_1\}$
`for all` e_2 of weight $w/2$
  $x \leftarrow s + e_2H_2^T$

$H =$ 

Total cost: $\underbrace{\binom{n/2}{w/2}}_{|\mathcal{L}_1|} + \underbrace{\binom{n/2}{w/2}}_{|\mathcal{L}_2|}$

$$\boxed{\text{Compute } \mathcal{L}_1 \cap \mathcal{L}_2 = \left\{ e_1 H_1^T \mid |(|e_1) = \tfrac{w}{2} \right\} \cap \left\{ s + e_2 H_2^T \mid |(|e_2) = \tfrac{w}{2} \right\}}$$

```
for all e_1 of weight w/2
    x ← e_1 H_1^T ; T[x] ← T[x] ∪ {e_1}
for all e_2 of weight w/2
    x ← s + e_2 H_2^T
    for all e_1 ∈ T[x]
        I ← I ∪ {(e_1, e_2)}
```

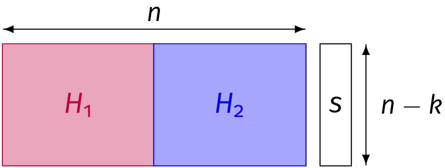
$H =$ 

$$\text{Total cost: } \underbrace{\binom{n/2}{w/2}}_{|\mathcal{L}_1|} + \underbrace{\binom{n/2}{w/2}}_{|\mathcal{L}_2|} + \underbrace{\frac{\binom{n/2}{w/2}^2}{2^{n-k}}}_{\frac{|\mathcal{L}_1|\cdot|\mathcal{L}_2|}{2^{n-k}}}$$

$$\boxed{\text{Compute } \mathcal{L}_1 \cap \mathcal{L}_2 = \left\{ e_1 H_1^T \mid |(|e_1) = \tfrac{w}{2} \right\} \cap \left\{ s + e_2 H_2^T \mid |(|e_2) = \tfrac{w}{2} \right\}}$$

```
for all e_1 of weight w/2
    x ← e_1 H_1^T ; T[x] ← T[x] ∪ {e_1}
for all e_2 of weight w/2
    x ← s + e_2 H_2^T
    for all e_1 ∈ T[x]
        I ← I ∪ {(e_1, e_2)}
return I
```

$H =$



Total cost: $\binom{n/2}{w/2} + \binom{n/2}{w/2} + \dfrac{\binom{n/2}{w/2}^2}{2^{n-k}}$

$|\mathcal{L}_1| \qquad |\mathcal{L}_2| \qquad \dfrac{|\mathcal{L}_1| \cdot |\mathcal{L}_2|}{2^{n-k}}$

Birthday algorithm

One particular error of Hamming weight w splits evenly with probability

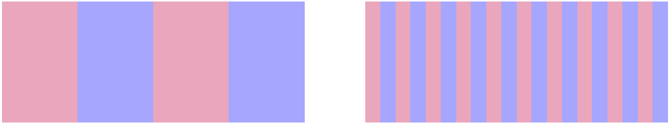
$$\mathcal{P} = \frac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$$

Birthday algorithm

One particular error of Hamming weight w splits evenly with probability

$$\mathcal{P} = \frac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$$

We may have to repeat with H divided in several different ways



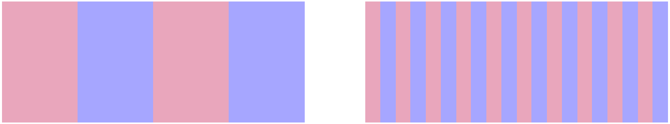
or more generally by picking the two halves randomly

Birthday algorithm

One particular error of Hamming weight w splits evenly with probability

$$\mathcal{P} = \frac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$$

We may have to repeat with H divided in several different ways



or more generally by picking the two halves randomly

Repeat $1/\mathcal{P}$ times to get most solutions. **Cost:** $O\left(\sqrt{\binom{n}{w}}\right)$.

Until here, we have not used linear algebra!

Until here, we have not used linear algebra!

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n\times n}$

$$\left(eH^T = s\right) \Leftrightarrow \left(e'H'^T = s'\right) \text{ where } \left\{ \begin{array}{ccc} H' & \leftarrow & UHP \\ s' & \leftarrow & sU^T \\ e' & \leftarrow & eP. \end{array} \right.$$

Until here, we have not used linear algebra!

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n\times n}$

$$\left(eH^T = s\right) \Leftrightarrow \left(e'H'^T = s'\right) \text{ where } \begin{cases} H' & \leftarrow & UHP \\ s' & \leftarrow & sU^T \\ e' & \leftarrow & eP. \end{cases}$$

Proof:
$$\begin{aligned} e'H'^T &= (eP)(UHP)^T \\ &= (eP)P^T H^T U^T \\ &= eH^T U^T \\ &= sU^T \\ &= s'. \end{aligned}$$

Idea: Perform a Gaussian Elimination and hope that all the errors are in positions corresponding to the identity part!

$$H' = UHP = \left[\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right] \quad \text{and } s' = sU^T = \left[\begin{array}{c} \\ \\ \\ \end{array}\right]$$

Idea: Perform a Gaussian Elimination and hope that all the errors are in positions corresponding to the identity part!

$$H' = UHP = \left[\begin{array}{c|c} \underbrace{\begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix}}_{n-k} & \phantom{\begin{matrix} 0 & 0 & 0 & 0 \end{matrix}} \end{array}\right] \quad \text{and } s' = sU^T = \left[\phantom{\begin{matrix} 0 \\ 0 \\ 0 \end{matrix}}\right]$$

possible if the first $n-k$ columns of HP are independent

Idea: Perform a Gaussian Elimination and hope that all the errors are in positions corresponding to the identity part!

$$H' = UHP = \left[\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right] \quad \text{and } s' = sU^T = \left[\begin{array}{c} \\ \\ \\ \end{array}\right]$$

$$e' = eP = \left[\begin{array}{c|ccc} \text{weight } w & 0 & \text{—} & 0 \end{array}\right]$$

Idea: Perform a Gaussian Elimination and hope that all the errors are in positions corresponding to the identity part!

$$H' = UHP = \left[\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right] \quad \text{and } s' = sU^T = \left[\begin{array}{c} \\ \\ \\ \end{array}\right]$$

$$e' = eP = \left[\begin{array}{c|c} s' & 0 \text{ —— } 0 \end{array}\right]$$

REPEAT:

1- Pick a permutation matrix P

Prange's algorithm

REPEAT:

1- Pick a permutation matrix P

2- Compute $UHP = \left[\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right]$

REPEAT:

1- Pick a permutation matrix P

2- Compute $UHP = \left[\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right]$

3- If $wt(sU^T) = w$ then return $(sU^T, 0)P^{-1}$

REPEAT:

1- Pick a permutation matrix P

2- Compute $UHP = \left[\begin{array}{c|c} \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & \quad\quad\quad \end{array}\right]$

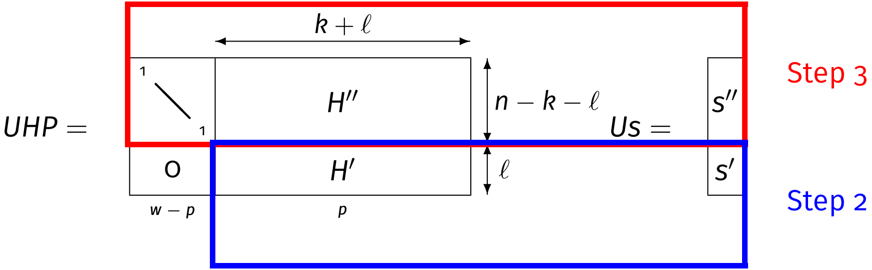
3- If $wt(sU^T) = w$ then return $(sU^T, 0)P^{-1}$

Cost of one iteration: $\mathcal{K} = n(n-k)$ column operations.
Success probability: $\mathcal{P} = \binom{n-k}{w} / \binom{n}{w}$.

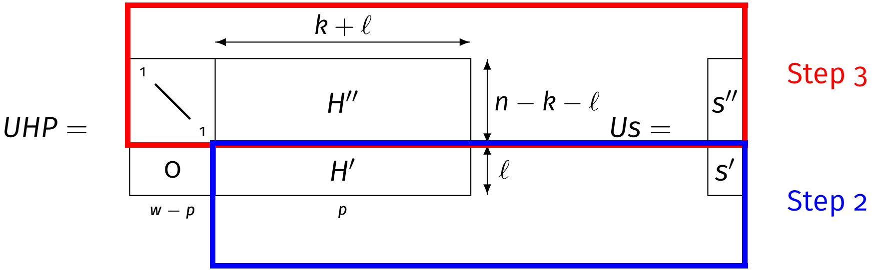
Total cost = $\mathcal{K}/\mathcal{P}$.

Stern and Dumer's algorithm

$$
UHP = \begin{array}{cc}
 & \overleftrightarrow{\quad k+\ell \quad} \\
\left[\begin{array}{c|c} \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & H'' \\ \hline 0 & H' \end{array}\right] & \begin{array}{l} n-k-\ell \\ \\ \ell \end{array} \\
\begin{array}{cc} w-p & \qquad\qquad p \end{array} &
\end{array}
\qquad
Us = \begin{bmatrix} s'' \\ \hline s' \end{bmatrix}
$$


$UHP =$
$k + \ell$
1
1
H''
$n - k - \ell$
$Us =$
s''
Step 3
0
H'
ℓ
s'
$w - p$
p
Step 2

Stern and Dumer's algorithm



Repeat:
1. Permutation + partial Gaussian elimination
2. Find many e' such that $|e'| = p$ and $H'e' = s'$
3. For all good e', test $|s'' + H''e'| \leq w - p$

Step 2 is Birthday Decoding (or whatever is best);
Step 3 is (a kind of) Prange;
Total cost is minimized over ℓ and p.

Iteration cost: $\mathcal{K} = n(n-k-\ell) + 2\sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell} + \frac{\binom{k+\ell}{p}}{2^\ell}$

Iteration cost: $\mathcal{K} = \underbrace{n(n-k-\ell)}_{\text{Gaussian elimination}} + 2\sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell} + \frac{\binom{k+\ell}{p}}{2^\ell}$

Iteration cost: $\mathcal{K} = \underbrace{n(n-k-\ell)}_{\text{Gaussian elimination}} + \underbrace{2\sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell}}_{\text{Birthday decoding}} + \frac{\binom{k+\ell}{p}}{2^\ell}$

$$\textbf{Iteration cost: } \mathcal{K} = \underbrace{n(n-k-\ell)}_{\text{Gaussian elimination}} + \underbrace{2\sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell}}_{\text{Birthday decoding}} + \underbrace{\frac{\binom{k+\ell}{p}}{2^\ell}}_{\text{Final check}}$$

Iteration cost: $\mathcal{K} = n(n-k-\ell) + 2\sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell} + \frac{\binom{k+\ell}{p}}{2^\ell}$

Gaussian elimination

Birthday decoding

Final check

Success probability: $\mathcal{P} = \frac{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}{\binom{n}{w}}$.

Total cost = $\mathcal{K}/\mathcal{P}$, minimized over p and ℓ.

More advanced algorithms

- Improved Birthday Decoding: overlapping support.
- Representations.
- Recursive Birthday Decoding.
- Decoding One Out of Many.
- Nearest Neighbour approach.

- **Theoretical** asymptotic exponent

  Best algorithm solves $\mathrm{SD}(n, W, R)$ in $2^{c \cdot n}$ operations with

1962	c = 0.121	[Pra62]
1988	c = 0.117	[Ste88, Dum89]
2011	c = 0.112	[MMT11]
2012	c = 0.102	[BJMM12]
2017	c = 0.095	[MO15, BM17]
2018	c = 0.089	[BM18]

  for $w = \mathrm{d}_{\mathrm{GV}}$ and worst choice of k.

- **Theoretical** asymptotic exponent

  Best algorithm solves $\mathrm{SD}(n, W, R)$ in $2^{c \cdot n}$ operations with

1962	c = 0.121	[Pra62]
1988	c = 0.117	[Ste88, Dum89]
2011	c = 0.112	[MMT11]
2012	c = 0.102	[BJMM12]
2017	c = 0.095	[MO15, BM17]
2018	c = 0.089	[BM18]

  for $w = \mathrm{d}_{\mathrm{GV}}$ and worst choice of k.

- **Practical** complexity?

The Decoding Challenge

decodingchallenge.org

Fichier Édition Affichage Historique Marque-pages Outils Aide

hallenges for code-based

decodingchallenge.org

Home Generic problems Documentation Contact NIST-like problems

Welcome to the code-based challenges webpage!

The purpose of this webpage is to assess the **practical hardness** of problems in coding theory.

The Challenges

The following challenges are currently available.

Generic problems. First we consider the two main problems on which Hamming-weight code-based cryptography mainly relies.

> Challenge: Syndrome Decoding

Given a matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$, a vector $\mathbf{s} \in \mathbb{F}_2^{n-k}$ and an integer $w \leq n$, find a vector $\mathbf{e} \in \mathbb{F}_2^n$ of Hamming weight $|\mathbf{e}| \leq w$ such that $\mathbf{H}\mathbf{e}^\top = \mathbf{s}^\top$. Here we will focus especially on the case with rate $R = 0.5$ and w close to the Gilbert-Varshamov bound.

> Challenge: Finding Low Weight Codewords

In a random linear code of size $n = 1280$ and rate $R = 0.5$, there exists on average a unique codeword of weight 144 (the Gilbert-Varshamov bound) and finding it should requires at least 2^{128} operations with the best known algorithms. Finding words of higher weight is easier. The goal of this challenge is to find codewords with a weight as close as possible to the GV-bound.

NIST-like problems. We propose challenges with the same parameter settings as the main cryptographic schemes proposed for the NIST standardization process for post-quantum cryptography. For now, we propose two such challenges in Hamming metric. In both cases, the goal is to assess the hardness of generic decoding, not to find distinguishers on the codes. Therefore we propose random linear codes with the same rate and error weight as the corresponding NIST candidates.

Latest news

20-08-2019. Announcement of the challenge at Crypto 2019 conference

12-08-2019. All challenges are now online

21-07-2019. Website is online.

12-07-2019. Contact email is active.

Launched in August 2019 by Aragon, Lavauzelle and L.

- **Goal:**
  - assess the practical complexity of problems in coding theory;
  - motivate the implementation of ISD algorithms;
  - increase the confidence in code-based crypto.

The Decoding Challenge

Launched in August 2019 by Aragon, Lavauzelle and L.

- **Goal:**
  - assess the practical complexity of problems in coding theory;
  - motivate the implementation of ISD algorithms;
  - increase the confidence in code-based crypto.

- **Concept:**
  - 4 categories of challenges;
  - instances of increasing size;
  - a hall of fame.

- 2 generic problems
  - Syndrome Decoding $k/n = 0.5$ and $w = \mathrm{d}_{\mathrm{GV}}$
  - Finding the Lowest Codeword
    for $k/n = 0.5$ and n of cryptographic size
- 2 problems based on schemes in the NIST competition
  - Goppa-McEliece $k/n = 0.8$ and $w = (n-k)/\log_2(n)$
  - QC-MDPC $k/n = 0.5$ and $w = \sqrt{n}$

Questions raised by implementation

Based on previous work from Landais, Sendrier, Meurer and Hochbach,

and recent work from Vasseur, Couvreur, Kunz and L.

- Choice of parameters p, ℓ, ε ... must be integers!
- Random shuffle vs. Canteaut-Chabaud.
- Birthday algorithm: sort vs. hash table.
- Allowing overlap?
- Early abort?
- ...

It's not just about asymptotic exponents anymore!

decodingchallenge.org

- **How to contribute?**
  - Solve some challenges!
  - Talk about the project to other people.
  - Propose this as a student project.
  - Contact us if you want to help.

decodingchallenge.org

- **How to contribute?**
  - Solve some challenges!
  - Talk about the project to other people.
  - Propose this as a student project.
  - Contact us if you want to help.

Current leader of the Hall of Fame:
Valentin Vasseur, $n = 450$ (for SD)
$\simeq 2^{47}$ operations (Dumer).

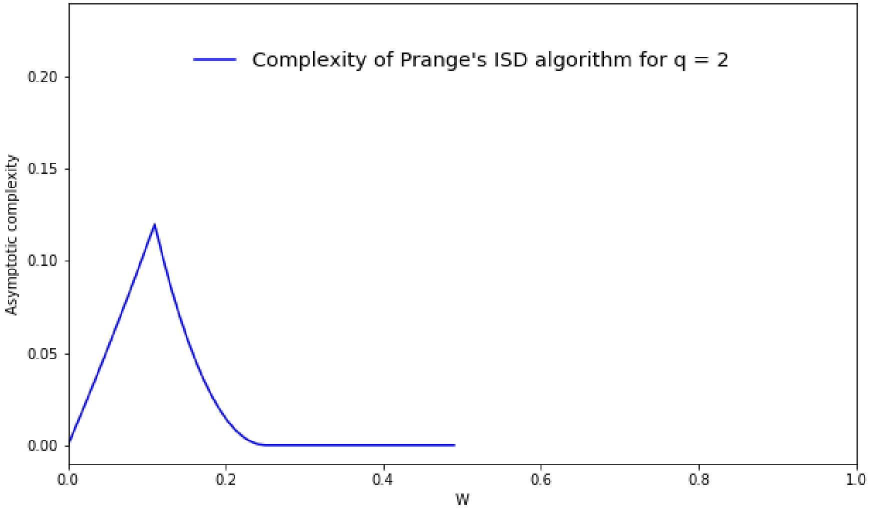
You dream to read your name in a Hall of Fame?
This is the chance of a lifetime!

Future challenges

- We intend to propose other categories of challenges
  - rank-metric Syndrome decoding;
  - q-ary Syndrome Decoding in Hamming metric;
  - q-ary Syndrome Decoding in Hamming metric with large weight.

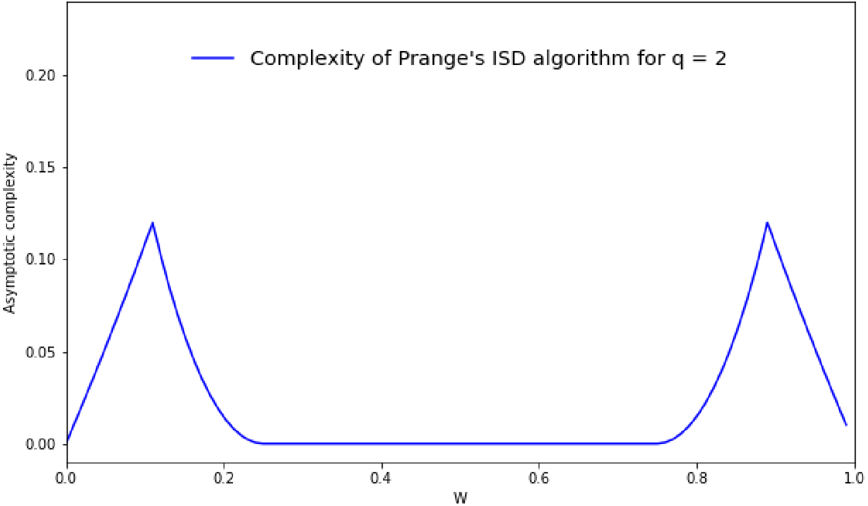
q-ary Syndrome Decoding

for $R = 1/2$:


Complexity of Prange's ISD algorithm for q = 2
Asymptotic complexity
0.00
0.05
0.10
0.15
0.20
0.0
0.2
0.4
0.6
0.8
1.0
W

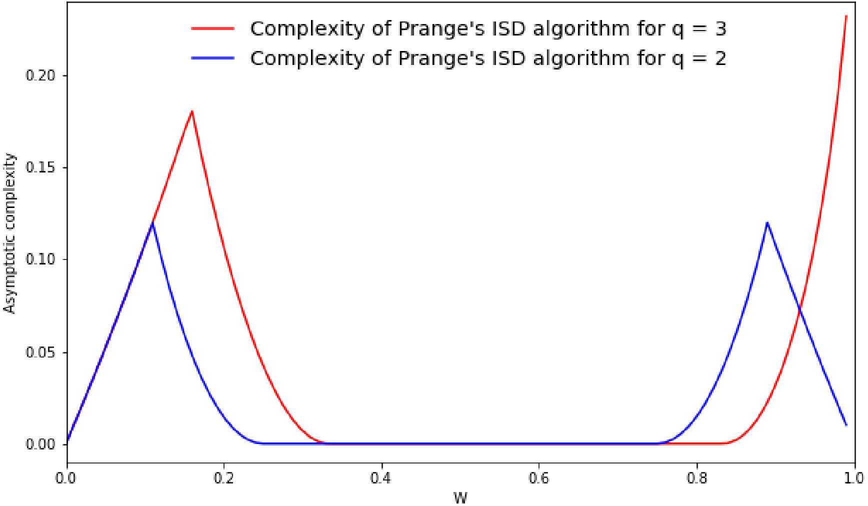
Binary vs. ternary Decoding Challenge

for $R = 1/2$:

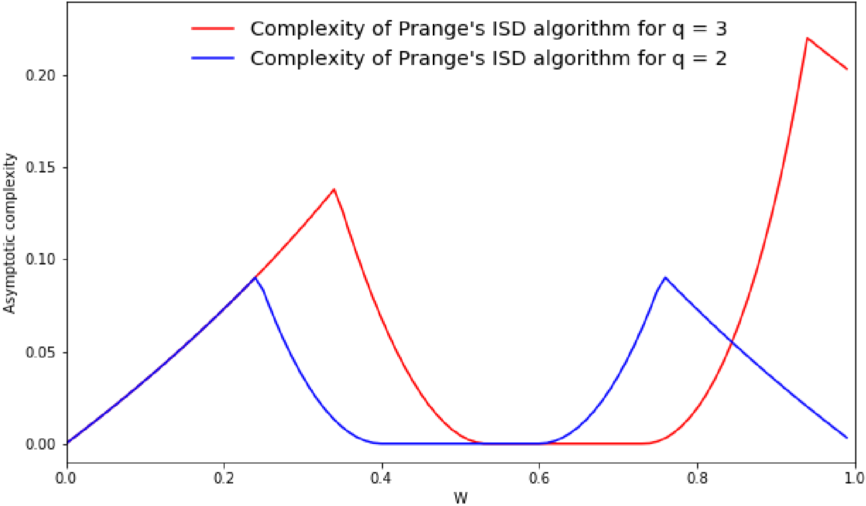


Binary vs. ternary Decoding Challenge

for $R = 1/2$:



for $R = 1/5$:


Complexity of Prange's ISD algorithm for q = 3
Complexity of Prange's ISD algorithm for q = 2
Asymptotic complexity
0.00
0.05
0.10
0.15
0.20
0.0
0.2
0.4
0.6
0.8
1.0
w

Some observations

- Asymetry

Some observations

- Asymetry
- Prange's algorithm works in polynomial time if

$$w \in \left[\!\left[\frac{q-1}{q}(n-k), k + \frac{q-1}{q}(n-k) \right]\!\right].$$

- Asymetry
- Prange's algorithm works in polynomial time if

$$w \in [\![\frac{q-1}{q}(n-k), k + \frac{q-1}{q}(n-k)]\!].$$

- For some values of R, there exists an equivalent of d_{GV} for large weight:

$$\binom{n}{d}(q-1)^d = q^{n-k}.$$

- Asymetry
- Prange's algorithm works in polynomial time if

$$w \in \llbracket \frac{q-1}{q}(n-k), k + \frac{q-1}{q}(n-k) \rrbracket.$$

- For some values of R, there exists an equivalent of d_{GV} for large weight:

$$\binom{n}{d}(q-1)^d = q^{n-k}.$$

- Worst case complexity for Prange's algorithm is reached for

$$R = 1 - \log_q(q-1) \text{ and } W = 1.$$

for $q = 3$ this is $R = 0.369$.

"Ternary Syndrome Decoding with Large Weight",
Bricout, Chailloux, Debris-Alazard and L., SAC 2019

- Motivation: Wave signature scheme [DST19].

Doing better than Prange?

"Ternary Syndrome Decoding with Large Weight",
Bricout, Chailloux, Debris-Alazard and L., SAC 2019

- Motivation: Wave signature scheme [DST19].
- What would an equivalent of Dumer's algorithm be?

Doing better than Prange?

“Ternary Syndrome Decoding with Large Weight”,
Bricout, Chailloux, Debris-Alazard and L., SAC 2019

- Motivation: Wave signature scheme [DST19].

- What would an equivalent of Dumer’s algorithm be?
- $W = 1$: we look for a solution containing no zeros.

"Ternary Syndrome Decoding with Large Weight",
Bricout, Chailloux, Debris-Alazard and L., SAC 2019

- Motivation: Wave signature scheme [DST19].

- What would an equivalent of Dumer's algorithm be?
- $W = 1$: we look for a solution containing no zeros.
- Up to a small transform, 1's and 2's become 0's and 1's.

"Ternary Syndrome Decoding with Large Weight",
Bricout, Chailloux, Debris-Alazard and L., SAC 2019

- Motivation: Wave signature scheme [DST19].

- What would an equivalent of Dumer's algorithm be?
- $W = 1$: we look for a solution containing no zeros.
- Up to a small transform, 1's and 2's become 0's and 1's.

Our problem is now the modular knapsack problem!

Given $k + \ell$ vectors $\mathbf{h}_i \in \mathbb{F}_3^\ell$ and a target vector $s \in \mathbb{F}_3^\ell$,
find L solutions of the form $(b_1, \ldots, b_{k+\ell}) \in \{0, 1\}^{k+\ell}$
such that $\sum_{i=1}^{k+\ell} b_i \mathbf{h}_i = s$.

Doing better than Prange?

"Ternary Syndrome Decoding with Large Weight",
Bricout, Chailloux, Debris-Alazard and L., SAC 2019

- Motivation: Wave signature scheme [DST19].

- What would an equivalent of Dumer's algorithm be?
- $W = 1$: we look for a solution containing no zeros.
- Up to a small transform, 1's and 2's become 0's and 1's.

Our problem is now the modular knapsack problem!

Given $k+\ell$ vectors $\mathbf{h}_i \in \mathbb{F}_3^\ell$ and a target vector $s \in \mathbb{F}_3^\ell$, find L solutions of the form $(b_1, \dots, b_{k+\ell}) \in \{0,1\}^{k+\ell}$ such that $\sum_{i=1}^{k+\ell} b_i \mathbf{h}_i = s$.

This can be solved using Wagner's algorithm [Wag02].

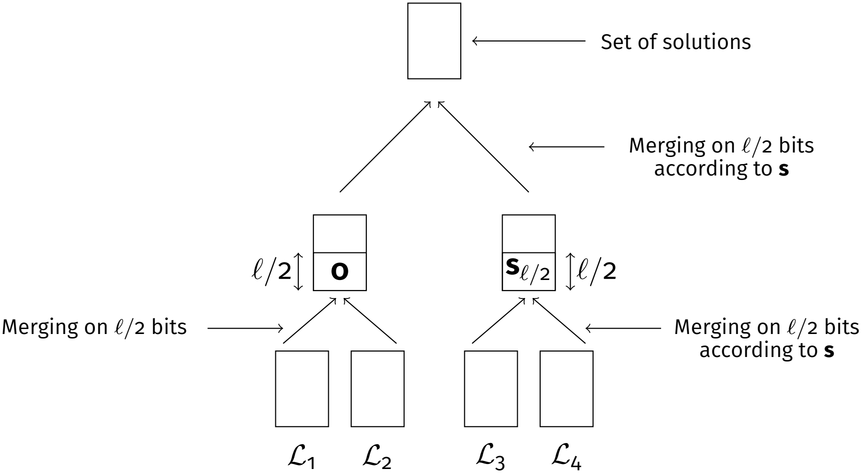


Figure: Wagner's algorithm with $a = 2$.

- Using Wagner's algorithm with a floors and $L = 3^{\ell/a}$ solutions can be solved in amortize time $O\left(3^{\ell/a}\right)$.

- Using Wagner's algorithm with a floors and $L = 3^{\ell/a}$ solutions can be solved in amortize time $O\left(3^{\ell/a}\right)$.

- Smoothing of the algorithm.

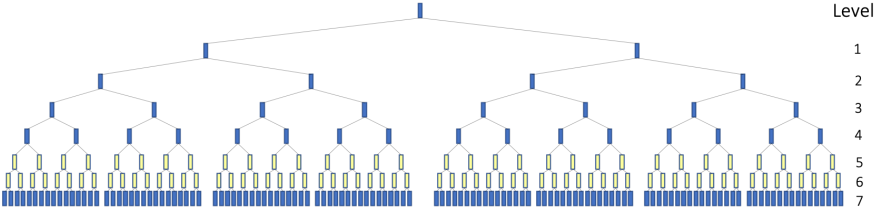
- Using Wagner's algorithm with a floors and $L = 3^{\ell/a}$ solutions can be solved in amortize time $O\left(3^{\ell/a}\right)$.
- Smoothing of the algorithm.
- Using representations (as in [BJMM12]).
- Using partial representations.

- Using Wagner's algorithm with a floors and $L = 3^{\ell/a}$ solutions can be solved in amortize time $O\left(3^{\ell/a}\right)$.
- Smoothing of the algorithm.
- Using representations (as in [BJMM12]).
- Using partial representations.

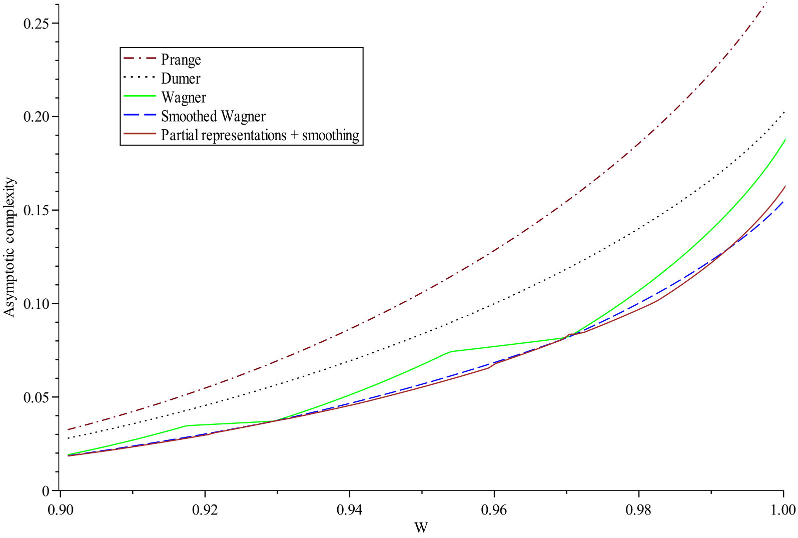
Remark: When $q \to \infty$, all ISD algorithm become equivalent to Prange's algorithm [Can17].

Our algorithm [BCDL19]

- 7 floors
- Blue = “left-right” splits (no representations)
- Yellow = representations
- Badly-formed elements at floor 4 and 5



Results (R = 0.5) [BCDL19]



HARDEST INSTANCES FOR $q = 3$ [BCDL19]

Algorithm	$q = 2$	$q = 3$ and $W > 0.5$
Prange	0.121 (R = 0.454)	0.369 (R = 0.369)
Dumer/Wagner	0.116 (R = 0.447)	0.269 (R = 0.369)
BJMM/our algorithm	0.102 (R = 0.427)	0.247 (R = 0.369)

Table: Best exponents with associated rates.

Hardest instances for $q = 3$ [BCDL19]

Algorithm	$q = 2$	$q = 3$ and $W > 0.5$
Prange	0.121 (R = 0.454)	0.369 (R = 0.369)
Dumer/Wagner	0.116 (R = 0.447)	0.269 (R = 0.369)
BJMM/our algorithm	0.102 (R = 0.427)	0.247 (R = 0.369)

Table: Best exponents with associated rates.

Algorithm	$q = 2$	$q = 3$ and $W > 0.5$
Prange	275	44
Dumer/Wagner	295	83
BJMM/Our algorithm	374	99

Table: Minimum input sizes (in kbits) for a time complexity of 2^{128}.

Concluding remarks

Conclusion

- Syndrome decoding is an old problem but still needs to be studied.

Conclusion

- Syndrome decoding is an old problem but still needs to be studied.
- Case $q \geq 3$ behaves very differently from $q = 2$.

Conclusion

- Syndrome decoding is an old problem but still needs to be studied.
- Case $q \geq 3$ behaves very differently from $q = 2$.
  - New problem: syndrome decoding in large weight;

- Syndrome decoding is an old problem but still needs to be studied.
- Case $q \geq 3$ behaves very differently from $q = 2$.
  - New problem: syndrome decoding in large weight;
  - Worst case complexity seems higher than in small weight;

Conclusion

- Syndrome decoding is an old problem but still needs to be studied.
- Case $q \geq 3$ behaves very differently from $q = 2$.
  - New problem: syndrome decoding in large weight;
  - Worst case complexity seems higher than in small weight;
  - New cryptographic schemes with shorter key size relying on this problem?

Conclusion

- Syndrome decoding is an old problem but still needs to be studied.
- Case $q \geq 3$ behaves very differently from $q = 2$.
  - New problem: syndrome decoding in large weight;
  - Worst case complexity seems higher than in small weight;
  - New cryptographic schemes with shorter key size relying on this problem?
  - Requires further study.

Conclusion

- Syndrome decoding is an old problem but still needs to be studied.
- Case $q \geq 3$ behaves very differently from $q = 2$.
  - New problem: syndrome decoding in large weight;
  - Worst case complexity seems higher than in small weight;
  - New cryptographic schemes with shorter key size relying on this problem?
  - Requires further study.
- Solve the challenges!

- Syndrome decoding is an old problem but still needs to be studied.
- Case $q \geq 3$ behaves very differently from $q = 2$.
  - New problem: syndrome decoding in large weight;
  - Worst case complexity seems higher than in small weight;
  - New cryptographic schemes with shorter key size relying on this problem?
  - Requires further study.
- Solve the challenges!

Thank you for your attention!

Michael Alekhnovich.
More on average case vs approximation complexity.
Computational Complexity, 20(4):755–786, 2011.

Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer.
Decoding random binary linear codes in $2^{n/20}$: How $1+1=0$ improves information set decoding.
In *Advances in Cryptology - EUROCRYPT 2012*, LNCS. Springer, 2012.

Leif Both and Alexander May.
Optimizing BJMM with Nearest Neighbors: Full Decoding in $2^{2/21n}$ and McEliece Security.
In *WCC Workshop on Coding and Cryptography*, September 2017.
on line proceedings, see
`http://wcc2017.suai.ru/Proceedings_WCC2017.zip`.

Leif Both and Alexander May.
Decoding linear codes with high error rate and its impact for LPN security.
In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography 2018*, volume 10786 of *LNCS*, pages 25–46, Fort Lauderdale, FL, USA, April 2018. Springer.

Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg.

On the inherent intractability of certain coding problems.
IEEE Trans. Inform. Theory, 24(3):384–386, May 1978.

Rodolfo Canto Torres.
Asymptotic analysis of ISD algorithms for the $q-$ary case.
In *Proceedings of the Tenth International Workshop on Coding and Cryptography WCC 2017*, September 2017.

Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich.
Wave: A new family of trapdoor one-way preimage sampleable functions based on codes.
Cryptology ePrint Archive, Report 2018/996, May 2019.
`https://eprint.iacr.org/2018/996`.

Il'ya Dumer.
Two decoding algorithms for linear codes.
Probl. Inf. Transm., 25(1):17–23, 1989.

Robert J. McEliece.
***A Public-Key System Based on Algebraic Coding Theory*, pages 114–116.**
Jet Propulsion Lab, 1978.
DSN Progress Report 44.

Alexander May, Alexander Meurer, and Enrico Thomae.
Decoding random linear codes in $O(2^{0.054n})$.

In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 107–124. Springer, 2011.

Alexander May and Ilya Ozerov.
On computing nearest neighbors with applications to decoding of binary linear codes.
In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 203–228. Springer, 2015.

Harald Niederreiter.
Knapsack-type cryptosystems and algebraic coding theory.
Problems of Control and Information Theory, 15(2):159–166, 1986.

Eugene Prange.
The use of information sets in decoding cyclic codes.
IRE Transactions on Information Theory, 8(5):5–9, 1962.

Vladimir Michilovich Sidelnikov and S.O. Shestakov.
On the insecurity of cryptosystems based on generalized Reed-Solomon codes.
Discrete Math. Appl., 1(4):439–444, 1992.

Jacques Stern.
A method for finding codewords of small weight.

In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *LNCS*, pages 106–113. Springer, 1988.

DAVID WAGNER.
A GENERALIZED BIRTHDAY PROBLEM.
In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, 2002.