LECTURE 3
INFORMATION SET DECODING ALGORITHMS

Summer School: *Introduction to Quantum-Safe Cryptography*

Thomas Debris-Alazard

July 03, 2024

Inria, École Polytechnique

Aim of Any Code-Based Cryptosystem:

Security relies on the hardness of the Decoding Problem (DP)

How to trust DP *hardness?*

$\longrightarrow$ By designing and studying algorithms solving DP!

An Old History (since 60 years):

Best algorithms: refinement of Prange's algorithm (1962)

Information Set Decoding (ISD) algorithms

$\longrightarrow$ Also a different and recent approach which turns out to be competitive: Dual Attacks

- Prange's Algorithm
- Find Collisions: Dumer's Algorithm
- Information Set Decoding Algorithms (ISD)
- Generalization of ISD to Reach Any Weights

PRANGE'S ALGORITHM

Our Aim:

Describing Prange's algorithm

Two points of view:

- Noisy codewords and generator matrices
- Syndromes and parity-check matrices

Our Aim:

Describing Prange's algorithm

Two points of view:

- Noisy codewords and generator matrices
- Syndromes and parity-check matrices

- Given: $\mathcal{C}$ be an $[n, k]_q$-code and $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{c} + \mathbf{e}$ where $\begin{cases} \mathbf{c} \in \mathcal{C} \\ |\mathbf{e}| = t \end{cases}$
- Recover: $\mathbf{e}$

Exhaustive Search: try all the $\mathbf{c}' \in \mathcal{C}$ until $|\mathbf{y} - \mathbf{c}'| = t$

$\longrightarrow$ If unicity of the solution: cost given by $\sharp\mathcal{C} = q^k$

Don't forget that $\mathcal{C}$ is a linear subset!

To fix the intuition: suppose t (Hamming weight of the error) being small

How could we use the "linearity" of $\mathcal{C}$ knowing that t is small?

First remark of Prange: use Information Sets!

Information Set:

$\mathcal{I} \subseteq \{1, \dots, n\}$ of size k is an information set of the $[n, k]_q$-$\mathcal{C}$ if:

$$\forall \mathbf{x} \in \mathbb{F}_q^k: \quad \exists(\text{unique})\ \mathbf{c} \in \mathcal{C} : \ \mathbf{c}_{\mathcal{I}} = \mathbf{x} \quad \left(\text{where } \mathbf{c}_{\mathcal{I}} = (c_i)_{i \in \mathcal{I}}\right)$$

Every codewords: uniquely determined by $k = \mathbf{dim}(\mathcal{C})$ coordinates given by $\mathcal{I}$

How to recover $\mathbf{c} \in \mathcal{C}$ from $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $|\mathbf{e}| = t$ by using information sets?

$\left(t \text{ can be supposed small}\right)$

First remark of Prange: use Information Sets!

Information Set:

$\mathcal{I} \subseteq \{1, \dots, n\}$ of size k is an information set of the $[n, k]_q$-$\mathcal{C}$ if:

$$\forall \mathbf{x} \in \mathbb{F}_q^k: \quad \exists(\text{unique})\ \mathbf{c} \in \mathcal{C} : \ \mathbf{c}_{\mathcal{I}} = \mathbf{x} \quad \left(\text{where } \mathbf{c}_{\mathcal{I}} = (c_i)_{i \in \mathcal{I}}\right)$$

Every codewords: uniquely determined by $k = \mathbf{dim}(\mathcal{C})$ coordinates given by $\mathcal{I}$

How to recover $\mathbf{c} \in \mathcal{C}$ from $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $|\mathbf{e}| = t$ by using information sets?

$\left(t \text{ can be supposed small}\right)$

$\longrightarrow$ If $\mathbf{e}_{\mathcal{I}} = \mathbf{0}$ $\left(\text{no errors on } \mathcal{I}\right)$,

then computing the unique $\mathbf{d} \in \mathcal{C}$ such that $\mathbf{d}_{\mathcal{I}} = \mathbf{y}_{\mathcal{I}}$ gives $\mathbf{c}$ as $\mathbf{c}_{\mathcal{I}} = \mathbf{y}_{\mathcal{I}}$!

Given $\mathbf{x} \in \mathbb{F}_q^k$ and $\mathcal{I} \subseteq [1, n]$ an information set, how to compute

the unique $\mathbf{c} \in \mathcal{C}$ such that $\mathbf{c}_{\mathcal{I}} = \mathbf{x}$?

Information Set:

$\mathcal{I} \subseteq \{1, \dots, n\}$ of size k, information set of of the $[n, k]_q$-$\mathcal{C}$ if:

$$\forall \mathbf{x} \in \mathbb{F}_q^k: \quad \exists(\text{unique})\ \mathbf{c} \in \mathcal{C} : \ \mathbf{c}_{\mathcal{I}} = \mathbf{x} \quad \left(\text{where } \mathbf{c}_{\mathcal{I}} = (c_i)_{i \in \mathcal{I}}\right)$$

$$\begin{aligned} \mathcal{I} \text{ information set for } \mathcal{C} \quad & \iff \quad \forall \mathbf{G} \text{ generator matrix of } \mathcal{C}, \ \mathbf{G}_{\mathcal{I}} \in \mathbb{F}_q^{k \times k} \text{ has rank } k \\ & \iff \quad \forall \mathbf{G} \text{ generator matrix of } \mathcal{C}, \ \mathbf{G}_{\mathcal{I}} \text{ is invertible} \end{aligned}$$

Given an information set $\mathcal{I}$, suppose that $\mathcal{I} = [1, k]$, then, $\mathbf{G}_{[1,k]}$ has rank k. By Gaussian elimination:

$$\mathbf{S}\mathbf{G} = (\mathbf{I}_k \mid \mathbf{A}) \quad \left(\text{still generator matrix}\right)$$

Given $\mathbf{x} \in \mathbb{F}_q^k$,

$$\mathbf{c} \stackrel{\text{def}}{=} \mathbf{x}\mathbf{S}\mathbf{G} = (\mathbf{x} \mid \mathbf{x}\mathbf{A}) \text{ is the unique codeword such that } \mathbf{c}_{\mathcal{I}} = \mathbf{x}$$

- Given: $\mathcal{C}$ an $[n, k]_q$-code and $\mathbf{y} \stackrel{\text{def}}{=} \mathbf{c}^{\text{sol}} + \mathbf{e}^{\text{sol}}$ where $\begin{cases} \mathbf{c}^{\text{sol}} \in \mathcal{C} \\ \left|\mathbf{e}^{\text{sol}}\right| = t \end{cases}$
- Recover: $\mathbf{e}^{\text{sol}}$

1. Pick an information set $\mathcal{I}$,
2. Compute the **unique** $\mathbf{c} \in \mathcal{C}$ such that
$$\mathbf{c}_{\mathcal{I}} = \mathbf{y}_{\mathcal{I}}$$
3. You win if $|\mathbf{y} - \mathbf{c}| = t$, namely
$$\mathbf{y}_{\mathcal{I}} = \mathbf{c}^{\text{sol}}_{\mathcal{I}} \iff \mathbf{e}^{\text{sol}}_{\mathcal{I}} = \mathbf{0}$$
Otherwise, go back to 1.

Running time of the algorithm: number of times we pick $\mathcal{I}$ $\Big($times cost of Gaussian elimination$\Big)$

Our Aim:

Describing Prange's algorithm

Two points of view:

- Noisy codewords and generator matrices
- Syndromes and parity-check matrices

$$\text{Fixing } \left(\mathbf{H}, \mathbf{s} \stackrel{\text{def}}{=} \mathbf{H}\mathbf{e}^{\mathsf{T}}\right) \text{ where } |\mathbf{e}| = t.$$

$\longrightarrow$ Linear system: $n - k$ **equations** and n **unknowns**

$$\left(\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}\right)$$

But. . .

Fixing $\left(\mathbf{H}, \mathbf{s} \stackrel{\text{def}}{=} \mathbf{H}\mathbf{e}^{\intercal}\right)$ where $|\mathbf{e}| = t$.

$\longrightarrow$ Linear system: $n-k$ **equations** and n **unknowns**

$$\left(\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}\right)$$

But. . .

with a **non-linear constraint** $\left(|\mathbf{e}| = t\right)$

Fixing $\left(\mathbf{H}, \mathbf{s} \stackrel{\text{def}}{=} \mathbf{H}\mathbf{e}^{\mathsf{T}}\right)$ where $|\mathbf{e}| = t$.

$\longrightarrow$ Linear system: $n - k$ **equations** and n **unknowns**

$$\left(\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}\right)$$

But. . .

with a **non-linear constraint** $\left(|\mathbf{e}| = t\right)$

Prange's Algorithm:

1. Fixing **a random set of k unknowns to** 0
2. Solving a square $(n - k) \times (n - k)$ linear system
3. Hoping the solution has the good Hamming weight otherwise repeat by fixing other k coordinates to 0

Pick a set of k coordinates $\mathcal{I}$ randomly

$\longrightarrow$ Suppose for the sake of simplicity that $\mathcal{I} = [n-k+1, n]$

1. Perform a Gaussian elimination,
$$\mathbf{S}\mathbf{H} = (\mathbf{I}_{n-k} \mid \mathbf{A})$$
2. Compute, $\mathbf{e}^\top = \begin{pmatrix} \mathbf{S}\mathbf{s}^\top \\ \mathbf{0} \end{pmatrix}$
3. If $|\mathbf{e}| \neq t$, then return to step 1 by choosing another set of $n-k$ coordinates where performing Gaussian elimination

- If unicity of the solution, probability of success

$$p = \frac{\binom{n-k}{t}(q-1)^t}{\binom{n}{t}(q-1)^t}$$

- If N solutions, probability of success

$$p \approx N \times \frac{\binom{n-k}{t}(q-1)^t}{\binom{n}{t}(q-1)^t}$$

$\longrightarrow$ But the number of solutions is $N = \max\left(1, \frac{\binom{n}{t}(q-1)^t}{q^{n-k}}\right)$

Conclusion:

Running time of Prange's algorithm (times the cost of Gaussian elimination),

$$\frac{1}{p} \quad \text{where} \quad p = \frac{\binom{n-k}{t}(q-1)^t}{\min\left(q^{n-k}, \binom{n}{t}(q-1)^t\right)} \quad \text{probability of success of one iteration}$$

PRANGE'S ALGORITHM: WHAT ELSE?

Prange's algorithm: pick $\mathcal{I}$ of size k and hope that $\mathbf{e}_{\mathcal{I}} = \mathbf{0}$

Is it not too strong to suppose that there are no errors on $\mathcal{I}$,

i.e., $\mathbf{e}_{\mathcal{I}} = \mathbf{0}$?

Natural idea: suppose there are p errors on $\mathcal{I}$, *i.e.*, $|\mathbf{e}_{\mathcal{I}}| = p$

$\longrightarrow$ Compute all the codewords $\mathbf{c} \in \mathcal{C}$ such that $|\mathbf{c}_{\mathcal{I}} - \mathbf{y}_{\mathcal{I}}| = p$

Better probability of success, but a cost $\binom{k}{p}(q-1)^p$ per iteration (exponential)

$\left(\text{test all the } \mathbf{z} \text{ with } |\mathbf{z}| = p \text{ as } \left(\mathbf{y}_{[1,k]} + \mathbf{z}\right)\left(\mathbf{I}_k, \mathbf{A}\right) \right)$

This algorithm is known as Lee-Brickell

FIND COLLISION: DUMER’S ALGORITHM

To understand how has been improved we need to backtrack!

Come Back to the Exhaustive Search:

$$\text{Given } \left(\mathbf{H}, \mathbf{s}^\top \stackrel{\text{def}}{=} \mathbf{H}\mathbf{x}^\top\right) \text{ with } |\mathbf{x}| = t$$

$$\longrightarrow \text{Try all the } \mathbf{e} \text{ with } |\mathbf{e}| \text{ and verify } \mathbf{H}\mathbf{e}^\top \stackrel{?}{=} \mathbf{s}^\top$$

Dumer's Idea:

Take advantage of the birthday paradox by looking for columns collision!

How large should be a group of people for two of them to be born the same day?

How large should be a group of people for two of them to be born the same day?

$\longrightarrow$ 23 $\approx \sqrt{365}$ is basically enough for this to be true with probability $\approx 1/2$

$\left(\text{number of pairs with 23 people, } \frac{23 \times 22}{2} \approx 365\right)$

How large should be a group of people for two of them to be born the same day?

$\longrightarrow$ 23 $\approx \sqrt{365}$ is basically enough for this to be true with probability $\approx 1/2$

$\left(\text{number of pairs with 23 people, } \frac{23\times 22}{2} \approx 365\right)$

Birthday Paradox in Computer Science:

Generate lists $\mathcal{L}_1, \mathcal{L}_2 \subseteq \{0,1\}^\ell$ of size L with elements independently picked uniformly at random

How many elements do we expect in $\mathcal{L}_1 \cap \mathcal{L}_2$?

$$\mathbb{E}\left(\sharp\ \mathcal{L}_1 \cap \mathcal{L}_2\right) = \frac{L^2}{2^\ell}$$

$\longrightarrow$ With $L = \sqrt{2^\ell}$ we expect one element in the intersection!

Proof.

$\mathcal{L}_1 = (X_1, \ldots, X_L)$ and $\mathcal{L}_2 = (Y_1, \ldots, Y_L)$, then

$$\sharp\ \mathcal{L}_1 \cap \mathcal{L}_2 = \sum_{i,j=1}^{L} 1_{\{X_i = Y_j\}}, \quad \text{then} \quad \mathbb{E}\left(\sharp\ \mathcal{L}_1 \cap \mathcal{L}_2\right) = \sum_{i,j=1}^{L} \mathbb{P}(X_i = Y_j) = \sum_{i,j=1}^{L} \frac{1}{2^\ell} \qquad \square$$

Dumer's Idea: given $\left(\mathbf{H}, \mathbf{H}\mathbf{x}^{\top}\right)$

1. Split $\mathbf{H}$ in two, *i.e.*, $\mathbf{H} = (\mathbf{H}_1, \mathbf{H}_2)$
2. Compute the lists
$$\mathcal{L}_1 = \left\{\mathbf{H}_1\mathbf{e}_1^{\top} \; : \; |\mathbf{e}_1| = \tfrac{t}{2}\right\} \quad \text{and} \quad \mathcal{L}_2 = \left\{\mathbf{s}^{\top} - \mathbf{H}_2\mathbf{e}_2^{\top} \; : \; |\mathbf{e}_2| = \tfrac{t}{2}\right\}$$
3. Compute $\mathcal{L}_1 \cap \mathcal{L}_2$, if it is non-empty it gives a solutions $(\mathbf{e}_1, \mathbf{e}_2)$

if the solution $\mathbf{x}$ splits as $(\mathbf{x}_1, \mathbf{x}_2)$ with $|\mathbf{x}_1| = |\mathbf{x}_2| = t/2$, then Dumer's algorithm finds it

$$\longrightarrow \text{It happens with probability} \quad \frac{\binom{n}{t/2}(q-1)^{t/2} \times \binom{n}{t/2}(q-1)^{t/2}}{\binom{n}{t}(q-1)^{t}} \approx 1$$

Dumer's Idea: given $\left(\mathbf{H}, \mathbf{H}\mathbf{x}^\top\right)$

1. Split $\mathbf{H}$ in two, *i.e.*, $\mathbf{H} = (\mathbf{H}_1, \mathbf{H}_2)$
2. Compute the lists
$$\mathcal{L}_1 = \left\{\mathbf{H}_1\mathbf{e}_1^\top \;:\; |\mathbf{e}_1| = \tfrac{t}{2}\right\} \quad \text{and} \quad \mathcal{L}_2 = \left\{\mathbf{s}^\top - \mathbf{H}_2\mathbf{e}_2^\top \;:\; |\mathbf{e}_2| = \tfrac{t}{2}\right\}$$
3. Compute $\mathcal{L}_1 \cap \mathcal{L}_2$, if it is non-empty it gives solutions $(\mathbf{e}_1, \mathbf{e}_2)$

- Lists $\mathcal{L}_1$ and $\mathcal{L}_2$ have size
$$\binom{n/2}{t/2}(q-1)^{t/2} \approx \sqrt{\binom{n}{t}(q-1)^t} \qquad \left(\text{use that } \binom{n}{u}(q-1)^u \approx q^{n \cdot h_q(u/n)}\right)$$
- Intersection of lists $\mathcal{L}_1 \cap \mathcal{L}_2$ have size $\dfrac{\sqrt{\binom{n}{t}(q-1)^t}^{\,2}}{q^{n-k}} = \dfrac{\sqrt{\binom{n}{t}(q-1)^t}}{q^{n-k}}$
- Running time of Dumer's algorithm:
$$\underbrace{\sqrt{\binom{n}{t}(q-1)^t}}_{\text{cost to builds lists}} + \underbrace{\frac{\binom{n}{t}(q-1)^t}{q^{n-k}}}_{\text{cost to build intersections}}$$
- Dumer's Algorithm returns $\max\left(1, \frac{\binom{n}{t}(q-1)^t}{q^{n-k}}\right)$ solutions of the decoding problem!

1. It returns all solutions of decoding problem

2. When decoding at distance t_{GV} for codes of rate $k/n \to 1$,

$$\text{Prange running time: } q^{n-k} \quad ; \quad \text{Dumer running time: } \sqrt{q^{n-k}}$$

$\longrightarrow$ Quadratic improvement over Prange's algorithm for these parameters!

3. Dumer's algorithm returns solutions in amortized time one if

$$\sqrt{\binom{n}{t}(q-1)^t} = \frac{\binom{n}{t}(q-1)^t}{q^{n-k}} \iff \binom{n}{t} = \left(q^{n-k}\right)^2$$

Would it be possible to combine both Prange and Dumer's approach?

$\longrightarrow$ Yes! It corresponds to the birth of Information Set Decoding (ISD) algorithms

INFORMATION SET DECODING ALGORITHMS

Combination of Ideas:

- We want to keep the Prange bet
- We want to use the fact that we can decode codes of rate k/n close to 1 with quadratic gain over Prange

$$\mathbf{SH} = \begin{pmatrix} \mathbf{1}_{n-k-\ell} & \mathbf{H}' \\ \mathbf{0} & \mathbf{H}'' \end{pmatrix} \quad \text{where} \quad \mathbf{H}'' \in \mathbb{F}_q^{\ell \times (k+\ell)}$$

With this partial Gaussian elimination,

$$\begin{aligned} \mathbf{He}^\top = \mathbf{s}^\top &\iff \mathbf{SHe}^\top = \mathbf{Ss}^\top \\ &\iff \begin{pmatrix} \mathbf{1}_{n-k-\ell} & \mathbf{H}' \\ \mathbf{0} & \mathbf{H}'' \end{pmatrix} \begin{pmatrix} \mathbf{e}'^\top \\ \mathbf{e}''^\top \end{pmatrix} \\ &\iff \begin{cases} \mathbf{e}'^\top + \mathbf{H}'\mathbf{e}''^\top = \mathbf{s}'^\top \\ \mathbf{H}''\mathbf{e}''^\top = \mathbf{s}''^\top \end{cases} \end{aligned}$$

The Algorithm:

1. Solve the decoding problem at distance p by computing all the solutions:
$$\mathbf{H}''\mathbf{e}''^\top = \mathbf{s}''^\top$$
$\longrightarrow$ It corresponds to decode a code of dimension k and length $k+\ell$ at distance p

2. Deduce a solutions $(\mathbf{e}', \mathbf{e}'')$

$\longrightarrow$ It will succeed if there are p errors on the window of size $k+\ell$

Two parameters in ISD: p and ℓ

Information Set Decoding:

1. Select randomly a window of size $k + \ell$
2. Solve a decoding problem at distance p for a code of dimension k and length $k + \ell$ but compute all solutions. Deduce potential solutions
3. If a solution has an Hamming weight p on the window of size $k + \ell$ will obtain it. Otherwise we repeat Step 1

- Prange's bet is step 1
- Use Dumer's algorithm to solve step 2: nice approach as we can compute (for well-chosen p and ℓ) all solutions in amortized time 1

It Interpolates Prange and Dumer' Algorithm:

Prange's algorithm: $\ell = p = 0$; Dumer's algorithm: $\ell = n - k$ and $p = t$

To improve the previous algorithm:

Use “better” algorithm than Dumer to solve the sub-decoding problem at distance p

PRANGE'S ALGORITHM FOR ANY WEIGHTS

Prange's Algorithm

1. Perform a Gaussian elimination,
$$\mathbf{S}\mathbf{H} = (\mathbf{I}_{n-k} \mid \mathbf{A})$$
2. Compute, $\mathbf{e}^{\top} = \begin{pmatrix} \mathbf{S}\mathbf{s}^{\top} \\ \mathbf{0} \end{pmatrix}$
3. If $|\mathbf{e}| \neq t$, then return to step 1 by choosing another set of $n-k$ coordinates where performing Gaussian elimination

By supposing that $\mathbf{s}$ is uniform, what is the typical weight of $\mathbf{e}$ after one iteration?

Prange's Algorithm

1. Perform a Gaussian elimination,
$$\mathbf{S}\mathbf{H} = (\mathbf{I}_{n-k} \mid \mathbf{A})$$
2. Compute, $\mathbf{e}^\top = \begin{pmatrix} \mathbf{S}\mathbf{s}^\top \\ \mathbf{0} \end{pmatrix}$
3. If $|\mathbf{e}| \neq t$, then return to step 1 by choosing another set of $n-k$ coordinates where performing Gaussian elimination

By supposing that $\mathbf{s}$ is uniform, what is the typical weight of $\mathbf{e}$ after one iteration?

$$|\mathbf{e}| = \frac{q-1}{q}(n-k)$$

$\longrightarrow$ The Hamming $\frac{q-1}{q}(n-k)$ can easily be reached in Prange's algorithm!

How could we reach larger weights easily?

Don't fix k unknowns to 0!

Generalized Prange's Algorithm

1. Perform a Gaussian elimination,
$$\mathbf{S}\mathbf{H} = (\mathbf{I}_{n-k} \mid \mathbf{A})$$
2. Compute, $\mathbf{e}^{\top} = \begin{pmatrix} \mathbf{S}\mathbf{s}^{\top} \\ \mathbf{x} \end{pmatrix}$
3. If $|\mathbf{e}| \neq t$, then return to step 1 by choosing another set of $n-k$ coordinates where performing Gaussian elimination

By supposing that $\mathbf{s}$ is uniform, what is the typical weight of $\mathbf{e}$ after one iteration?

Don't fix k unknowns to 0!

Generalized Prange's Algorithm

1. Perform a Gaussian elimination,
$$\mathbf{SH} = (\mathbf{I}_{n-k} \mid \mathbf{A})$$
2. Compute, $\mathbf{e}^\top = \begin{pmatrix} \mathbf{Ss}^\top \\ \mathbf{x} \end{pmatrix}$
3. If $|\mathbf{e}| \neq t$, then return to step 1 by choosing another set of $n-k$ coordinates where performing Gaussian elimination

By supposing that $\mathbf{s}$ is uniform, what is the typical weight of $\mathbf{e}$ after one iteration?

$$|\mathbf{e}| = |\mathbf{x}| + \tfrac{q-1}{q}(n-k)$$

$\longrightarrow$ $\mathbf{x} \in \mathbb{F}_q^k$, by carefully choosing $|\mathbf{x}| \in [1,k]$ we can reach easily any weight in the interval

$$\left[\tfrac{q-1}{q}(n-k), k + \tfrac{q-1}{q}(n-k)\right]$$

$$\left(R = k/n \quad \text{and} \quad \tau = t/n\right)$$