Generalized Inverse Based Decoding

Vlad-Florin Dragoi, Ferucio Laurentiu Tiplea
vlad.dragoi@uav.ro

Aurel Vlaicu Univ. of Arad, Romania
Alexandru Ioan Cuza University of Iasi, Romania
LITIS EA 4108 Université de Rouen Normandie, France

Project No:PN-III-P1-1.1-PD-2019-0285

Post-quantum code-based encryption schemes

Quantum computing industry [1]

1. Silicon Foundry Jul 14, 2020, "Near Future :Quantum Computing"

Post-quantum cryptography

- A polynomial time quantum algorithm for solving number theoretic problems used in publick key cryptography (P. Shor 1994).

Post-quantum cryptography

- A polynomial time quantum algorithm for solving number theoretic problems used in publick key cryptography (P. Shor 1994).
- NIST – post-quantum cryptography standardization process [2] (2016–)

2. http ://csrc.nist.gov/groups/ST/post-quantum-crypto/

Post-quantum cryptography

- A polynomial time quantum algorithm for solving number theoretic problems used in publick key cryptography (P. Shor 1994).
- NIST – post-quantum cryptography standardization process [2] (2016–)
- Round 3 finalists in the KEM section are *Classic McEliece* (code-based), Crystal-Kyber, NTRU, Saber (lattice-based)

Public-key encryption from codes

- Generate a linear code ($\boldsymbol{G}$) with an efficient decoding algorithm and mask its structure – **Key generation**

$$\boldsymbol{G_{pub}} = \boldsymbol{SGP} \qquad , \qquad \boldsymbol{H_{pub}} = \boldsymbol{SHP}$$

Public-key encryption from codes

- Generate a linear code ($\boldsymbol{G}$) with an efficient decoding algorithm and mask its structure – **Key generation**

$$\boldsymbol{G_{pub}} = \boldsymbol{SGP} \qquad , \qquad \boldsymbol{H_{pub}} = \boldsymbol{SHP}$$

- Erroneous codeword/ syndrome – **Encrypted data**

(McEliece) $\boldsymbol{z} = \boldsymbol{mG} + \boldsymbol{e}$ or $\boldsymbol{m} \rightarrow \boldsymbol{e}\ ,\ \boldsymbol{z} = \boldsymbol{He}^t$ (Neiderreiter)

Public-key encryption from codes

- Generate a linear code ($\boldsymbol{G}$) with an efficient decoding algorithm and mask its structure – **Key generation**

$$\boldsymbol{G_{pub}} = \boldsymbol{SGP} \qquad , \qquad \boldsymbol{H_{pub}} = \boldsymbol{SHP}$$

- Erroneous codeword/ syndrome – **Encrypted data**

(McEliece) $\boldsymbol{z} = \boldsymbol{mG} + \boldsymbol{e}$ or $\boldsymbol{m} \to \boldsymbol{e}\ ,\ \boldsymbol{z} = \boldsymbol{He}^t$ (Neiderreiter)

- Security of the cipher

Generic decoding
$$Alg(\boldsymbol{mG_{pub}} + \boldsymbol{e}, \boldsymbol{G_{pub}}) = \boldsymbol{m}$$

Syndrome decoding
$$Alg(\boldsymbol{H_{pub}e}, \boldsymbol{H_{pub}}) = \boldsymbol{e}$$

Syndrome decoding problem

Binary SDP Berlekamp, McEliece, van Tilborg (1978)

Input : $\boldsymbol{H} \in \mathcal{M}_{n-k,n}(\mathbb{F}_2)$ of rank $n-k$, a vector $\boldsymbol{s} \in \mathbb{F}_2^{n-k}$ and $t \in \mathbb{Z}^+$.
Output : $\boldsymbol{e} \in \mathbb{F}_2^n$, with $\mathrm{wt}(\boldsymbol{e}) \leqslant t$, such that $\boldsymbol{He} = \boldsymbol{s}$.

Syndrome decoding problem

Binary SDP Berlekamp, McEliece, van Tilborg (1978)

Input : $\boldsymbol{H} \in \mathcal{M}_{n-k,n}(\mathbb{F}_2)$ of rank $n-k$, a vector $\boldsymbol{s} \in \mathbb{F}_2^{n-k}$ and $t \in \mathbb{Z}^+$.
Output : $\boldsymbol{e} \in \mathbb{F}_2^n$, with $\mathrm{wt}(\boldsymbol{e}) \leqslant t$, such that $\boldsymbol{He} = \boldsymbol{s}$.

- When $\boldsymbol{s} = \boldsymbol{0}$ we talk about Low weight codewords problem (A. Vardy 1997)

Syndrome decoding problem

Binary SDP Berlekamp, McEliece, van Tilborg (1978)

Input : $\boldsymbol{H} \in \mathcal{M}_{n-k,n}(\mathbb{F}_2)$ of rank $n-k$, a vector $\boldsymbol{s} \in \mathbb{F}_2^{n-k}$ and $t \in \mathbb{Z}^+$.
Output : $\boldsymbol{e} \in \mathbb{F}_2^n$, with $\mathrm{wt}(\boldsymbol{e}) \leqslant t$, such that $\boldsymbol{He} = \boldsymbol{s}$.

- When $\boldsymbol{s} = \mathbf{0}$ we talk about Low weight codewords problem (A. Vardy 1997)
- Security of the McEliece/Niederreiter stands on SDP/LWC - find solution of linear systems of equations with a particular feature (weight condition)

Syndrome decoding problem

Binary SDP Berlekamp, McEliece, van Tilborg (1978)

Input : $\boldsymbol{H} \in \mathcal{M}_{n-k,n}(\mathbb{F}_2)$ of rank $n-k$, a vector $\boldsymbol{s} \in \mathbb{F}_2^{n-k}$ and $t \in \mathbb{Z}^+$.
Output : $\boldsymbol{e} \in \mathbb{F}_2^n$, with $\mathrm{wt}(\boldsymbol{e}) \leqslant t$, such that $\boldsymbol{He} = \boldsymbol{s}$.

- When $\boldsymbol{s} = \mathbf{0}$ we talk about Low weight codewords problem (A. Vardy 1997)
- Security of the McEliece/Niederreiter stands on SDP/LWC - find solution of linear systems of equations with a particular feature (weight condition)

Algorithms for solving these two problems explore "in a clever way" the set of solutions/null space

Generalized Inverse of a matrix over finite fields

Generalized Inverse

1. GI - generalize the concept of inverse of non-square matrices (Moore, Penrose, Rao, Mitra, Greville, Ben Israel, Fullton, Duffin, etc.)
2. Several types of such inverses exists, depending on some constraints : reflexive, normalized, pseudoinverse (Moore-Penrose)
3. Are used to solve linear system of equation
   – when the system is defined over $\mathbb{R}$ or $\mathbb{C}$, variants of GIs are used to find min. weight solutions
4. What happens over finite fields ? Can GIs be used for SDP/LWC ? [3]

3. M. Finiasz in 2005 : *pseudo-inverse can be used for computing solutions of weight* $t > n/2$

Generalized Inverse

- Let $\boldsymbol{A} \in \mathcal{M}_{m,n}(\mathbb{F})$. A *generalized inverse of* $\boldsymbol{A}$ is any $\boldsymbol{X} \in \mathcal{M}_{n,m}(\mathbb{F})$ s.t.

$$\boldsymbol{AXA} = \boldsymbol{A}.$$

  The set of GIs of a matrix is $\mathcal{GI}(\boldsymbol{A})$

Generalized Inverse

- Let $\boldsymbol{A} \in \mathcal{M}_{m,n}(\mathbb{F})$. A *generalized inverse of* $\boldsymbol{A}$ is any $\boldsymbol{X} \in \mathcal{M}_{n,m}(\mathbb{F})$ s.t.

$$\boldsymbol{AXA} = \boldsymbol{A}.$$

The set of GIs of a matrix is $\mathcal{GI}(\boldsymbol{A})$

Thm. Ben-Israel and Greville

Let $\boldsymbol{A} \in \mathcal{M}_{m,n}(\mathbb{F})$, $\boldsymbol{b} \in \mathcal{R}(\boldsymbol{A})$, and $\boldsymbol{X} \in \mathcal{GI}(\boldsymbol{A})$. Then, $\boldsymbol{x}$ is a solution to $\boldsymbol{Ax} = \boldsymbol{b}$ if and only if $\boldsymbol{x} = \boldsymbol{Xb} + (\boldsymbol{I} - \boldsymbol{XA})\boldsymbol{c}$, for some $\boldsymbol{c} \in \mathbb{F}^n$.

Thm.

Let $\boldsymbol{A} \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ with full row rank and $\boldsymbol{b} \in \mathcal{R}(\boldsymbol{A})$ with $\boldsymbol{b} \neq 0$. Then,

$$\{\boldsymbol{x} \in \mathbb{F}_q^n \mid \boldsymbol{Ax} = \boldsymbol{b}\} = \{\boldsymbol{Xb} \mid \boldsymbol{X} \in \mathcal{GI}(\boldsymbol{A})\}.$$

Construction of GIs

- Finding a solution to $\boldsymbol{Ax} = \boldsymbol{b}$ with a weight restriction resumes to exploring the set $\mathcal{GI}(\boldsymbol{A})$ until a solution $|\boldsymbol{Xb}| = t$ is found

Construction of GIs

- Finding a solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with a weight restriction resumes to exploring the set $\mathcal{GI}(\boldsymbol{A})$ until a solution $|\boldsymbol{X}\boldsymbol{b}| = t$ is found
- How to sample from $\mathcal{GI}(\boldsymbol{A})$?

Construction of GIs

- Finding a solution to $\boldsymbol{Ax} = \boldsymbol{b}$ with a weight restriction resumes to exploring the set $\mathcal{GI}(\boldsymbol{A})$ until a solution $|\boldsymbol{Xb}| = t$ is found
- How to sample from $\mathcal{GI}(\boldsymbol{A})$?

Thm. Ben-Israel and Greville

Let $\boldsymbol{A} \in \mathcal{M}_{m,n}(\mathbb{F})$, $\boldsymbol{P} \in \mathrm{GL}_m(\mathbb{F})$, and $\boldsymbol{Q} \in \mathrm{GL}_n(\mathbb{F})$. Then, the function $f : \mathcal{GI}(\boldsymbol{A}) \to \mathcal{GI}(\boldsymbol{PAQ})$ given by $f(\boldsymbol{X}) = \boldsymbol{Q}^{-1}\boldsymbol{X}\boldsymbol{P}^{-1}$, for any $\boldsymbol{X} \in \mathcal{GI}(\boldsymbol{A})$, is a bijection.

Construction of GIs

- "Nice" forms are thus prefered, e.g.,
  - canonical form ($\boldsymbol{Q}$ is an invertible matrix)

$$\boldsymbol{PAQ} = \begin{pmatrix} \boldsymbol{I}_r & \boldsymbol{0} \end{pmatrix} \Rightarrow \mathcal{GI}(\boldsymbol{A}) = \left\{ \boldsymbol{Q} \begin{pmatrix} \boldsymbol{I}_r \\ \boldsymbol{X}_2 \end{pmatrix} \boldsymbol{P} \mid \boldsymbol{X}_2 \right\} \Rightarrow \boldsymbol{Xb} = \boldsymbol{Q} \begin{pmatrix} \boldsymbol{Pb} \\ \boldsymbol{X}_2 \boldsymbol{Pb} \end{pmatrix}$$

  - standard form ($\boldsymbol{Q}$ could be a permutation)

$$\boldsymbol{PAQ} = \begin{pmatrix} \boldsymbol{I}_r & \boldsymbol{V} \end{pmatrix} \Rightarrow \mathcal{GI}(\boldsymbol{A}) = \left\{ \boldsymbol{Q} \begin{pmatrix} \boldsymbol{I}_r - \boldsymbol{VX}_2 \\ \boldsymbol{X}_2 \end{pmatrix} \boldsymbol{P} \mid \boldsymbol{X}_2 \right\} \Rightarrow \boldsymbol{Xb} = \boldsymbol{Q} \begin{pmatrix} \boldsymbol{Pb} - \boldsymbol{VX}_2 \boldsymbol{Pb} \\ \boldsymbol{X}_2 \boldsymbol{Pb} \end{pmatrix}$$

- A single transformation $(\boldsymbol{P}, \boldsymbol{Q})$ suffices to compute $\mathcal{GI}(\boldsymbol{A})$

GI based solver for SDP

- A generic solver fixes a transformation $(\boldsymbol{P}, \boldsymbol{Q})$ for which $\mathcal{GI}(\boldsymbol{A})$ is known and samples $\boldsymbol{X} \in \mathcal{GI}(\boldsymbol{A})$ until $|\boldsymbol{X}\boldsymbol{b}| = t$.

GI based solver for SDP

- A generic solver fixes a transformation $(\boldsymbol{P}, \boldsymbol{Q})$ for which $\mathcal{GI}(\boldsymbol{A})$ is known and samples $\boldsymbol{X} \in \mathcal{GI}(\boldsymbol{A})$ until $|\boldsymbol{X}\boldsymbol{b}| = t$.
- Remark : Many inverses give the same solution
  Optimize the sampling !

GI based solver for SDP

- A generic solver fixes a transformation $(\boldsymbol{P}, \boldsymbol{Q})$ for which $\mathcal{GI}(\boldsymbol{A})$ is known and samples $\boldsymbol{X} \in \mathcal{GI}(\boldsymbol{A})$ until $|\boldsymbol{Xb}| = t$.
- Remark : Many inverses give the same solution
  Optimize the sampling !
- Randomization : changing the transformation $(\boldsymbol{P}, \boldsymbol{Q})$ and allowing a fixed number of samples for each transformation decreases the overall time complexity of the decoder (in simulations).

Information Set Decoding and GID

Information Set Decoding (ISD) [4]

- Prange's decoding technique :

$$\boldsymbol{He} = \boldsymbol{s} \qquad |\boldsymbol{S}, \Pi$$

$$\boldsymbol{SH}\Pi\left(\Pi^{-1}\boldsymbol{e}\right) = \boldsymbol{Ss}$$

$$\begin{pmatrix}\boldsymbol{I}_r & \boldsymbol{V}\end{pmatrix}\boldsymbol{e}^* = \boldsymbol{s}^*$$

If $\boldsymbol{e}^* = (\boldsymbol{e}_1, \boldsymbol{0}_{n-r})$ then $\|\boldsymbol{s}^*\| = t$

4. Prange(1957), Lee-Brickell (1988), Stern(1988), Dumer (1991), Canteaut et Chabaud (1998), May, Meurer, Thomae (2011), Becker, Joux, May, Meurer (2012), May, Ozerov (2015)

Information Set Decoding (ISD) [4]

- Prange's decoding technique :

$$\begin{aligned} \boldsymbol{He} &= \boldsymbol{s} & &\Big| \boldsymbol{S}, \Pi \\ \boldsymbol{SH}\Pi\left(\Pi^{-1}\boldsymbol{e}\right) &= \boldsymbol{Ss} \\ \begin{pmatrix} \boldsymbol{I}_r & \boldsymbol{V} \end{pmatrix} \boldsymbol{e}^* &= \boldsymbol{s}^* \end{aligned}$$

  If $\boldsymbol{e}^* = (\boldsymbol{e}_1, \boldsymbol{0}_{n-r})$ then $\|\boldsymbol{s}^*\| = t$
- Other variants allow a small weight (p) on the support of $\boldsymbol{V}$. In the asymptotic, for $t = o(n)$ the time complexity of all variants converge to that of Prange's algorithm (Canto-Torres and Sendrier 2016).

4. Prange(1957), Lee-Brickell (1988), Stern(1988), Dumer (1991), Canteaut et Chabaud (1998), May, Meurer, Thomae (2011), Becker, Joux, May, Meurer (2012), May, Ozerov (2015)

ISD and GID

- Prange's algorithm generates solutions to $\boldsymbol{He} = \boldsymbol{s}$ of the form $\boldsymbol{Xs}$ with

$$\boldsymbol{X} \in \left\{ \boldsymbol{Q} \begin{pmatrix} \boldsymbol{I}_r \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{P} \mid (\boldsymbol{P}, \boldsymbol{Q}) \in \mathrm{GL}_r(\mathbb{F}) \times \mathrm{S}_n(\mathbb{F}),\ (\exists \boldsymbol{V} : \boldsymbol{PHQ} = (\boldsymbol{I}_r \quad \boldsymbol{V})) \right\}.$$

- The set of all solutions can be generated :
  - *By fixing a transformation* : for a given transformation $(\boldsymbol{P}, \boldsymbol{Q}) \in \mathrm{GL}_r(\mathbb{F}) \times \mathrm{S}_n(\mathbb{F})$ with $\boldsymbol{PHQ} = \begin{pmatrix} \boldsymbol{I}_r & \boldsymbol{V} \end{pmatrix}$ for some $\boldsymbol{V}$, we have

$$\mathcal{GI}(\boldsymbol{H}) = \left\{ \boldsymbol{Q} \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{pmatrix} \boldsymbol{P} \mid \boldsymbol{VX}_2 + \boldsymbol{X}_1 = \boldsymbol{I}_r \right\};$$

  - *By fixing a GI* :

$$\mathcal{GI}(\boldsymbol{H}) = \left\{ \boldsymbol{Q} \begin{pmatrix} \boldsymbol{I}_r \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{P} \mid (\boldsymbol{P}, \boldsymbol{Q}) \in \mathrm{GL}_r(\mathbb{F}) \times \mathrm{GL}_n(\mathbb{F}),\ (\exists \boldsymbol{V} : \boldsymbol{PHQ} = (\boldsymbol{I}_r \quad \boldsymbol{V})) \right\}.$$

ISD and GID

- Prange, Lee-Brickell, Stern, Leon, Finiasz-Sendrier and multiple decompositions techniques

$$\boldsymbol{PHQ} = \begin{pmatrix} \boldsymbol{V}_1 & \boldsymbol{I}_\ell & \boldsymbol{0} \\ \boldsymbol{V}_2 & \boldsymbol{0} & \boldsymbol{I}_{r-\ell} \end{pmatrix},$$

do not run through the entire set of solutions.

ISD and GID

- Prange, Lee-Brickell, Stern, Leon, Finiasz-Sendrier and multiple decompositions techniques

$$\boldsymbol{PHQ} = \begin{pmatrix} \boldsymbol{V}_1 & \boldsymbol{I}_\ell & \boldsymbol{0} \\ \boldsymbol{V}_2 & \boldsymbol{0} & \boldsymbol{I}_{r-\ell} \end{pmatrix},$$

  do not run through the entire set of solutions.

- All these variants of ISD are particular GI based decoders.

ISD AND GID

- Prange, Lee-Brickell, Stern, Leon, Finiasz-Sendrier and multiple decompositions techniques

$$\boldsymbol{PHQ} = \begin{pmatrix} \boldsymbol{V}_1 & \boldsymbol{I}_\ell & \boldsymbol{0} \\ \boldsymbol{V}_2 & \boldsymbol{0} & \boldsymbol{I}_{r-\ell} \end{pmatrix},$$

  do not run through the entire set of solutions.

- All these variants of ISD are particular GI based decoders.

- ISD algorithms for low-weight codewords are particular GI based decoders.

Perspectives on GID

MIN-CWP

- Look at SDP as an optimization problem : MIN-CWP

MIN-CWP

- Look at SDP as an optimization problem : MIN-CWP
- There is a sharp reduction between MIN-CWP over $\mathbb{F}_2$ and MIN-SAT affine (use GI for the proof)

MIN-CWP

- Look at SDP as an optimization problem : MIN-CWP
- There is a sharp reduction between MIN-CWP over $\mathbb{F}_2$ and MIN-SAT affine (use GI for the proof)
- We propose a SAT solver for MIN-CWP

Simulations on GID

- Experiments with $n \leqslant 2000$ show that using a polynomial set of samples from $\mathcal{GI}(\boldsymbol{H})$, the solution have weight in the interval

$$\left[r\frac{q-1}{q} - \sqrt{n}, r\frac{q-1}{q} + n - r + \sqrt{n}\right].$$

Simulations on GID

- Experiments with $n \leqslant 2000$ show that using a polynomial set of samples from $\mathcal{GI}(\boldsymbol{H})$, the solution have weight in the interval

$$\left[r\frac{q-1}{q} - \sqrt{n}, r\frac{q-1}{q} + n - r + \sqrt{n}\right].$$

- Is this always true ? What happens when n goes to infinity ? (theoretical evidence)

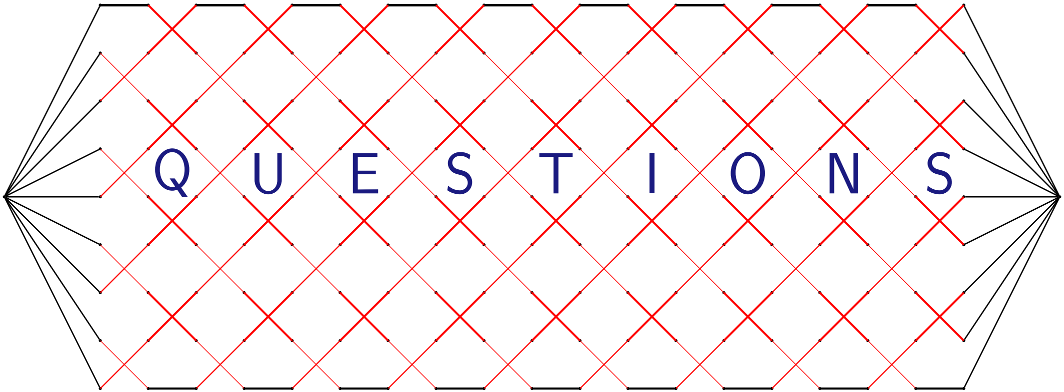
Simulations on GID

- Experiments with $n \leqslant 2000$ show that using a polynomial set of samples from $\mathcal{GI}(\boldsymbol{H})$, the solution have weight in the interval

$$\left[r\frac{q-1}{q} - \sqrt{n}, r\frac{q-1}{q} + n - r + \sqrt{n}\right].$$

- Is this always true ? What happens when n goes to infinity ? (theoretical evidence)

- Solutions of weight $n/2$ were easily retrieved by the GI based Decoder – In average this problem might be easy even though there are intractable instances (Graham and Diaconis (1985)).


QUESTIONS