Improved Information Set Decoding - Decoding Random Linear Codes in $O(2^{0.054n})$

Alexander May, Alexander Meurer, Enrico Thomae

ASIACRYPT 2011, Seoul
Recap Binary Linear Codes

A **binary linear code** $C$ is a $k$-dimensional subspace of $\mathbb{F}_2^n$

- Generator matrix
  
  $$G = \begin{bmatrix} 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix}_{k \times n}$$
  
  $C = \{ x^t \cdot G : x \in \mathbb{F}_2^k \}$

- Parity check matrix
  
  $$H = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{bmatrix}_{n-k \times n}$$
  
  $C = \{ H \cdot c = 0 : c \in \mathbb{F}_2^n \}$
Recap Binary Linear Codes

A binary linear code $C$ is a $k$-dimensional subspace of $\mathbb{F}_2^n$

- Generator matrix
  
  \[ G = \begin{bmatrix}
  \begin{array}{c}
  \vdots \\
  0/1
  \end{array}
  \end{bmatrix}^{n \times k} \quad C = \{ x^t \cdot G : x \in \mathbb{F}_2^k \} \]

- Parity check matrix
  
  \[ H = \begin{bmatrix}
  \begin{array}{c}
  \vdots \\
  0/1
  \end{array}
  \end{bmatrix}^{n \times (n-k)} \quad C = \{ H \cdot c = 0 : c \in \mathbb{F}_2^n \} \]

Minimum distance

\[ d = \min_{x \neq y \in C} \{ \text{wt}(x+y) \} \]

$C$ is called binary $[n,k,d]$ code.
Recap Binary Linear Codes

- **Running time** of decoding algorithms: $T(n,k,d)$
- In this talk: Analysis for **random** binary linear codes with constant rate $R = k/n$
- For $n \to \infty$, $k$ and $d$ are related via Gilbert-Varshamov bound, thus
  \[ T(n,k,d) = T(n,k) \]
- We compare algorithms by their **complexity coefficient** $F(k)$, i.e.
  \[ T(n,k) = O(2^{F(k)n}) \]
Scenario (Bounded Distance Decoding)

Obtain $x = c + e$ with $c \in C$ and $w := \text{wt}(e) = \left\lfloor \frac{d-1}{2} \right\rfloor$

Goal: Find $e$ and thus $c = x + e$

Consider syndrome $s := s(x) = H \cdot x = H \cdot (c + e) = H \cdot e$

→ Find linear combination of $w$ columns of $H$ matching $s$
Scenario (Bounded Distance Decoding)

Obtain $x = c + e$ with $c \in C$ and $w := \text{wt}(e) = \left\lceil \frac{d-1}{2} \right\rceil$

**Goal:** Find $e$ and thus $c = x + e$

Consider **syndrome** $s := s(x) = H \cdot x = H \cdot (c + e) = H \cdot e$

→ Find **linear combination** of $w$ columns of $H$ matching $s$

**Brute-Force complexity**

\[ T(n, k, d) = \binom{n}{w} \]
Scenario (Bounded Distance Decoding)

Obtain \( x = c + e \) with \( c \in C \) and \( w := \text{wt}(e) = \left\lfloor \frac{d-1}{2} \right\rfloor \)

**Goal:** Find \( e \) and thus \( c = x + e \)

Consider syndrome \( s := s(x) = H \cdot x = H \cdot (c + e) = H \cdot e \)

→ Find **linear combination** of \( w \) columns of \( H \) matching \( s \)

Complexity

\[ F(k) \leq 0.38677 \]
Some very basic observations

Allowed (linear algebra) transformations

- Permuting the columns of $H$ does not change the problem
Some very basic observations

Allowed (linear algebra) transformations

• Permuting the columns of \( H \) does not change the problem
Some very basic observations

Allowed (linear algebra) transformations

- Permuting the columns of $H$ does not change the problem
Some very basic observations

Allowed (linear algebra) transformations

- Permuting the columns of $\mathbf{H}$ does not change the problem
- Elementary row operations on $\mathbf{H}$ do not change the problem
Some very basic observations

Allowed (linear algebra) transformations

- Permuting the columns of $\mathbf{H}$ does not change the problem
- Elementary row operations on $\mathbf{H}$ do not change the problem

\[
\begin{align*}
\text{weight } w & \\
n & = \text{n-k} \\
\mathbf{H} & = \mathbf{s}
\end{align*}
\]
Some very basic observations

Allowed (linear algebra) transformations

- Permuting the columns of $H$ does not change the problem
- Elementary row operations on $H$ do not change the problem

$$\begin{align*}
\text{weight } w \\
U_G \\
H \\
= \\
U_G \\
\text{Invertible (n-k)x(n-k) matrix}
\end{align*}$$
Randomized systematic form

From now on, we work on a randomly column-permuted version of $H$ in systematic form.

$H = U_G \ast U_P \ast I_{(n-k)}$

$(n-k) \times k$ matrix

$(n-k)$-dimensional identity matrix
Basic Information Set Decoding

"Reducing the brute-force search space by linear algebra."
Basic Information Set Decoding

Fundamental principle:

- **Randomization**: transform $H$ into column-permuted systematic form

  \[
  H \rightarrow Q_{\text{I}_{n-k}}
  \]

- **Search phase**: Try to find solution $e = \text{weight } w$, i.e.

  \[
  Q_{\text{I}_{n-k}} = s
  \]

- If no solution found: Rerandomize and try again!
Basic Information Set Decoding

Fundamental principle:

- **Randomization**: transform $H$ into column-permutated systematic form

$$T(n,k,d) = \Pr["good" \text{ randomization}]^{-1} \times C[\text{Search}]$$

- If no solution found: Rerandomize and try again!
Classical ISD variants

Lee-Brickel (1988)

- Brute force on $Q$, i.e. search $e' = \begin{bmatrix} p \\ \end{bmatrix}$ with $\text{wt}(Qe' + s) = w - p$
- If so, fill up $e'' = \begin{bmatrix} w - p \\ \end{bmatrix}$ with remaining $(w-p)$ 1's
Classical ISD variants

Lee-Brickel (1988)

- Brute force on $Q$, i.e. search $e' = \begin{bmatrix} p \\ n-k \end{bmatrix}$ with $\text{wt}(Qe' + s) = w-p$
- If so, fill up $e'' = \begin{bmatrix} w-p \\ n-k \end{bmatrix}$ with remaining $(w-p)$ 1's
Classical ISD variants

Lee-Brickel (1988)

\[ T(n,k,d) = \Pr["good" \text{ randomization}]^{-1} \times C[\text{Brute force}] \]

\[ = \left( \binom{k}{p} \binom{n-k}{w-p} \right)^{-1} \times \binom{k}{p} \]
Classical ISD variants

Lee-Brickel (1988)

\[ Q_k \oplus I_{n-k} = s \quad \iff \quad \text{wt}(Qe' + s) = w - p \]

Complexity (bounded distance decoding)

\[ F(k) \leq 0.05751 \]
Classical ISD variants

Stern (1989)

- Meet-in-the-middle on $Q$

\[
\begin{bmatrix}
p/2 & p/2 & 0 & w - p \\
k/2 & k/2 & l & n - k - l
\end{bmatrix} = S
\]

\[
q_1, \ldots, q_k
\]

$Q'$
Classical ISD variants

Stern (1989)

- **Meet-in-the-middle** on \( Q \)

\[
\begin{array}{cccc}
p/2 & p/2 & 0 & w - p \\
k/2 & k/2 & l & n-k-l \\
q_1, \ldots, q_k & Q' \\
\end{array}
\]

Find \( e' = \begin{bmatrix} p/2 \\ p/2 \end{bmatrix} \) with \( Qe' = s' \) on first \( l \) coordinates, i.e.

via standard sort & match

\[
\sum_{i \in I_1} q_i = s' \quad \sum_{i \in I_2} q_i + s'
\]

\( I_1 \subseteq [1, \ldots, \frac{k}{2}] \)

\( I_2 \subseteq [\frac{k}{2} + 1, \ldots, k] \)
Classical ISD variants

Stern (1989)

- Meet-in-the-middle on $Q$

\[
p/2 \quad p/2 \quad 0 \quad w - p
\]

\[
l \quad \frac{k}{2} \quad n - k - l \quad \frac{k}{2}
\]

\[
q_1, \ldots, q_k
\]

\[
Q' = s \iff I_{n-k}
\]

\[
p/2 \quad p/2
\]

\[
q_1, \ldots, q_k
\]

\[
Q' + s = I_{n-k}
\]
Classical ISD variants

Stern (1989)

- Meet-in-the-middle on $Q$

\[
\begin{align*}
&\frac{p}{2} & \frac{p}{2} & 0 & w - p \\
&\frac{k}{2} & \frac{k}{2} & l & n - k - l
\end{align*}
\]

\[
\begin{array}{c}
\begin{array}{c}
q_1, \ldots, q_k \\
\end{array}
\end{array}
\quad = \quad
\begin{array}{c}
\begin{array}{c}
I_{n-k}
\end{array}
\end{array}
\quad = \quad
\begin{array}{c}
\begin{array}{c}
I_{n-k}
\end{array}
\end{array}
\quad + \quad
\begin{array}{c}
\begin{array}{c}
s'
\end{array}
\end{array}
\quad + \quad \begin{array}{c}
\begin{array}{c}
s
\end{array}
\end{array}
\end{array}
\]
Classical ISD variants

Stern (1989)

- Meet-in-the-middle on $Q$

\[
\begin{align*}
\begin{bmatrix}
p/2 & p/2 & 0 & w - p \\
k/2 & k/2 & l & n - k \end{bmatrix}
& \begin{bmatrix}
q_1, \ldots, q_k \\
Q' \end{bmatrix}

& = 
\begin{bmatrix}
I_{n-k} \\
s \end{bmatrix}

\overset{\leftrightarrow}{\Rightarrow}

\begin{bmatrix}
0 & w - p \\
0 & 0 \end{bmatrix}
& \begin{bmatrix}
s' \\
s \end{bmatrix}

& = 
\begin{bmatrix}
I_{n-k} \\
e'' \end{bmatrix}
\end{align*}
\]
Classical ISD variants

Stern (1989)

- Meet-in-the-middle on $Q$

$$T(n,k,d) = \Pr["good" \ \text{randomization}]^{-1} \times C[M\text{itM}]$$

$$= \frac{(k/2)^2 (n-k-l) - 1}{(p/2)(w-p)} \times \binom{k/2}{n/w}$$
Classical ISD variants

Stern (1989)

- Meet-in-the-middle on $Q$

$\begin{bmatrix} p/2 & p/2 & 0 & w - p \\ k/2 & k/2 & l & n-k-l \end{bmatrix} = s$

Complexity (bounded distance decoding)

$F(k) \leq 0.05563$
Wrapping up so far

- Different methods search for different error-distributions
  - Lee-Brickel (1988)
  - Stern (1989)
Wrapping up so far

- Different methods search for different error-distributions
  - Lee-Brickel (1988)
  - Stern (1989)

- Recently: **Ball-collision decoding** (CRYPTO11) by Bernstein et al.
  - Improving $\Pr[\text{good rand.}]$ by allowing $q$ extra 1's within the length $l$ zero-window
  - $F(k) \leq 0.05558$
Wrapping up so far

- Different methods search for different error-distributions
  - Lee-Brickel (1988)
  - Stern (1989)

- Recently: Ball-collision decoding (CRYPTO11) by Bernstein et al.
  - Improving $\Pr[\text{good rand.}]$ by allowing $q$ extra 1's within the length $l$ zero-window
  - $F(k) \leq 0.05558$
  - Both Stern and BallColl improve runtime at the cost of increased space consumption (→ see paper for details)
Wrapping up so far

Asymptotic complexity (Bounded Distance Decoding)

<table>
<thead>
<tr>
<th>Method</th>
<th>F(k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee-Brickel</td>
<td>0.05751</td>
</tr>
<tr>
<td>Stern</td>
<td>0.05563</td>
</tr>
<tr>
<td>BallColl</td>
<td>0.05558</td>
</tr>
</tbody>
</table>
Our New Algorithm
Generalized ISD framework

- **Starting point:** Generalized ISD framework of Finiasz and Sendrier (ASIACRYPT 2009)

- Observation: Since first \( l \) columns of \( \mathbf{I}_{n-k} \) can not be used in Stern's algorithm, we can simply "shift" them to the \( \mathbf{Q} \)-part of \( \mathbf{H} \).
Generalized ISD framework

- **Starting point:** Generalized ISD framework of Finiasz and Sendrier (ASIACRYPT 2009)
- Observation: Since first l columns of $I_{n-k}$ can not be used in Stern's algorithm, we can simply „shift“ them to the $Q$-part of $H$
Generalized ISD framework

- **Starting point**: Generalized ISD framework of Finiasz and Sendrier (ASIACRYPT 2009)

- Observation: Since first l columns of $I_{n-k}$ can not be used in Stern's algorithm, we can simply "shift" them to the $Q$-part of $H$.
### Generalized ISD framework

- **Major subproblem:** Finding exactly $p$ columns amongst $q_1, \ldots, q_{k+l}$ matching $s$ on its first $l$ coordinates $s'$

$$H = \begin{bmatrix} q_1, \ldots, q_{k+l} & 0 \\ Q' & I_{n-k-l} \end{bmatrix}$$
Generalized ISD framework

- **Major subproblem:** Finding exactly $p$ columns amongst $q_1, \ldots, q_{k+l}$ matching $s$ on its first $l$ coordinates $s'$

- **Stern** divides the $q_i$ into disjoint sets, i.e., searches index sets $I_1 \subset \left[ 1, \ldots, \frac{k+l}{2} \right]$ and $I_2 \subset \left[ \frac{k+l}{2} + 1, \ldots, k+l \right]$ with $\sum_{i \in I_1} q_i + \sum_{i \in I_2} q_i = s'$

$$H = \begin{pmatrix}
q_1, \ldots, q_{k+l} \\
Q' \\
\mathbf{0}
\end{pmatrix}$$

$l$ rows
Our New Algorithm

- **Main contribution:** More efficient *non-disjoint* matching inspired by *representation technique* of Howgrave-Graham and Joux (in the context of subset sum algorithms), i.e. pick $I_1, I_2 \subseteq [1, \ldots, k+l]$ with $|I_j| = \frac{p}{2}$ and $\sum_{i \in I_1} q_i + \sum_{i \in I_2} q_i = s'$

\[
H = \begin{bmatrix} q_1, \ldots, q_{k+l} & 0 \\ Q' & I_{n-k-l} \end{bmatrix} \text{ with } l \text{ rows}
\]
The representation trick in detail

- **Aim**: Find a representation of weight \( p \) such that

\[
\begin{bmatrix}
q_1, \ldots, q_{k+l}
\end{bmatrix}
= \begin{bmatrix}
s_1' \\
\vdots \\
s_l'
\end{bmatrix}
\]

- There are \( \binom{p}{p/2} \) ways of decomposing \( \begin{bmatrix} p \end{bmatrix} \), e.g.

\[
\begin{bmatrix}
p/2 \\
p/2 \\
p/2
\end{bmatrix}
, \begin{bmatrix}
p/2 \\
p/2
\end{bmatrix}
, \ldots
\]

→ It suffices to find one of them!
How can we profit from representations?

<table>
<thead>
<tr>
<th>p</th>
<th>w-p</th>
</tr>
</thead>
<tbody>
<tr>
<td>k+l</td>
<td>n-k-l</td>
</tr>
</tbody>
</table>

- **Advantage**: Every solution expands into \( \binom{p}{p/2} \) equivalent solutions
- **Disadvantage**: Search space grows from \( \binom{(k+l)/2}{p/2} \) to \( \binom{k+l}{p/2} \)
- **Goal**: Find one solution via Divide & Conquer strategy
- **Rule of thumb**: We get better if \( \binom{k+l}{p/2} < \binom{(k+l)/2}{p/2} \)
How can we profit from representations?

- **Advantage:** Every solution expands into equivalent solutions
- **Disadvantage:** Search space grows from \( p \) to \( w - p \)

**Goal:** Find one solution via Generalized Birthday attack (Wagner)

**Rule of thumb:** We get better if

\[
\frac{k + l}{p^2} \cdot \binom{n - k - l}{w - p} \cdot \binom{n}{w}^{-1} \cdot \binom{k + l}{p/2} \cdot \binom{p}{p/2} = 0
\]

**Complexity**

\[
T(n,k,d) = \Pr[\text{“good” randomization}]^{-1} \times C[\text{RepTrick}]
\]

\[
= \left( \frac{k + l}{p^2} \cdot \binom{n - k - l}{w - p} \cdot \binom{n}{w}^{-1} \right) \cdot \binom{k + l}{p/2} \cdot \binom{p}{p/2}
\]
How can we profit from representations?

<table>
<thead>
<tr>
<th>p</th>
<th>w-p</th>
</tr>
</thead>
<tbody>
<tr>
<td>k+l</td>
<td>n-k-l</td>
</tr>
</tbody>
</table>

\[
q_1, \ldots, q_{k+l} \quad 0
\]

\[
Q' \quad 1_{n-k-l}
\]

Complexity (bounded distance decoding)

\[ F(k) \leq 0.05364 \]
Complexity comparison

Asymptotic complexity (Bounded Distance Decoding)

- Lee-Brickel: $F(k) = 0.05751$
- BallColl: $F(k) = 0.05558$
- MMT: $F(k) = 0.05364$
Summary

- Asymptotically fastest generic decoding algorithm
- Not a mere time-memory trade-off (→ see paper)
- First application of HGJ representation trick to a different scenario

Open questions

- Precise analysis w.r.t. polynomial factors (e.g. linear algebra)
- Transfer Bernstein's speed-up tricks for Stern's algorithm
- What is the real break even point compared to other methods?
Summary

- Asymptotically fastest generic decoding algorithm
- Not a mere time-memory trade-off (→ see paper)
- First application of HGJ representation trick to a different scenario

Open questions

- Precise analysis w.r.t. polynomial factors (e.g. linear algebra)
- Transfer Bernstein's speed-up tricks for Stern's algorithm
- What is the real break even point compared to other methods?

Thank you!