Generic Decoding of Linear Codes

“Quantum Cryptanalysis” seminar
Dagstuhl, September 9-13, 2013

Nicolas Sendrier
Linear Codes for Telecommunication

Coding theory: find good codes (linear expansion) with
- good combinatorial properties (minimum distance)
- fast decoding procedures

Random linear codes have optimal minimum distance but cannot be decoded (decoding is more or less an exhaustive search)

Algebraic codes have fast decoding procedures up to the designed distance (smaller than information theoretic bounds)
Linear Codes for Cryptography

- If a random linear code is used, no one can decode efficiently
- If an algebraic code is used, anyone who knows the algebraic structure has access to a fast decoder

Assuming that the knowledge of the linear expansion does not reveal the algebraic structure:
- The decoder is known from the legitimate user
- For anyone else, the code looks random
Security Reduction

McEliece (78) suggests the use of binary Goppa codes for encryption.

A binary \((n, k)\) Goppa code corrects \(\frac{n-k}{\log_2 n}\) errors efficiently.

\((H\) is a parity check matrix, \(r = n - k\) the codimension\)

<table>
<thead>
<tr>
<th>Goppa code Distinguishing</th>
<th>NP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Instance:</strong> (H \in {0, 1}^{r \times n})</td>
<td></td>
</tr>
<tr>
<td><strong>Question:</strong> Is ({x \in {0, 1}^n \mid xH^T = 0}) a binary Goppa code?</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Goppa Bounded Decoding</th>
<th>NP-hard</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Instance:</strong> (H \in {0, 1}^{r \times n}, s \in {0, 1}^r)</td>
<td></td>
</tr>
<tr>
<td><strong>Output:</strong> (e \in {0, 1}^n) such that (\text{wt}(e) \leq \frac{r}{\log_2 n}) and (eH^T = s)</td>
<td></td>
</tr>
</tbody>
</table>

If both problems are hard on average then the McEliece encryption scheme is secure.
Syndrome Decoding – Problem Statement

Computational Syndrome Decoding

\[ \text{CSD}(n, r, w) \]

Given \( H \in \{0, 1\}^{r \times n} \) and \( s \in \{0, 1\}^r \), solve \( eH^T = s \) with \( \text{wt}(e) \leq w \)

\[ H = \]

\[ e = \text{Hamming weight } w \]

Find \( w \) columns of \( H \) adding to \( s \)

Very close to a subset sum problem

For instance:

\[ \begin{cases} n = 2048 \\ r = 352 & \rightarrow \text{computing effort} > 2^{80} \\ w = 32 \end{cases} \]
Algorithm 0

\[ H = \begin{bmatrix} n \end{bmatrix} \]

Compute every sum of \( w \) columns \( \rightarrow \) complexity \( \binom{n}{w} \) column ops.

1 column operation
\[
\begin{cases} 
1 \text{ read or write} \\
\text{and} \\
1 \text{ test} \\
\text{and} \\
1 \text{ addition or weight computation}
\end{cases}
\]
Algorithm 1: Birthday Decoding

\[ H = \begin{bmatrix} H_1 & H_2 \end{bmatrix} \]

\[ w/2 \quad w/2 \]

Compute \( \{H_1e \mid \text{wt}(e) = w/2\} \cap \{s + H_2e \mid \text{wt}(e) = w/2\} \)

Complexity \( 2^{\binom{n/2}{w/2}} \) and non-empty with probability \( \frac{\binom{n/2}{w/2}^2}{\binom{n}{w}} \)

\( \rightarrow \) average cost \( 2 \frac{\binom{n}{w}}{\binom{n/2}{w/2}} \approx 2 \frac{\sqrt{\binom{n}{w}}}{\sqrt[4]{w\pi}/2} \) can be reduced to \( O\left(\sqrt[4]{\binom{n}{w}}\right) \)
Algorithm 2: Information Set Decoding [Prange, 1962]

Big difference with subset sums: you can use linear algebra

\[ n = r + k \]

\[
\begin{bmatrix}
1 & 1 \\
\end{bmatrix}
\]

Repeat for several permutation matrices \( P \)

Claim: if \( \text{wt}(Us) \leq w \), I win!

Success probability: \( \binom{r}{w} / \binom{n}{w} \approx (r/n)^w \)

Total cost: \( \approx rn(n/r)^w \) column operations
Algorithm 2': ISD [Lee & Brickell, 1988]

**Idea:** amortize the Gaussian elimination

\[ n = r + k \]

\[ UHP = \begin{bmatrix}
1 & \quad & 0 \\
\downarrow & & \uparrow \\
1 & & H'
\end{bmatrix} \]

\[ Us = \begin{bmatrix}
1 \\
\downarrow \\
1 \\
\end{bmatrix} \]

Repeat for several permutation matrices \( P \)

**Claim:** if \( \exists e \) with \( \text{wt}(e) = p \) and \( \text{wt}(Us + H'e) = w - p \), I win!

Success probability: \( \frac{{n \choose w-p} \cdot {k \choose p}}{{n \choose w}} \)  
Iteration cost: \( rn + \frac{k}{p} \)

Total cost: \( \frac{{n \choose w}}{r \choose w-p} \left( 1 + \frac{rn}{k \choose p} \right) \), only a polynomial gain
Generalized Information Set Decoding – Teaser

Stern, 1989 ; Dumer 1993

\[
UHP = \begin{bmatrix} 1 & k + \ell \\ 1 & H'' \\ 0 & H' \end{bmatrix}
\]

\[
Us = \begin{bmatrix} s'' \\ s' \end{bmatrix}
\]

Repeat:
1. Permutation + partial Gaussian elimination
2. Find many \( e' \) such that \( H'e' = s' \)
3. For all good \( e' \), test \( \text{wt}(s'' + e'H''T) \leq w - p \)

Step 3. is (a kind of) Lee & Brickell which embeds Step 2
Step 2. is Birthday Decoding (or whatever is best)

Total cost is minimized over \( \ell \) and \( p \)
Generalized Information Set Decoding – Teaser

Stern, 1989; Dumer 1993

\[ UHP = \begin{bmatrix}
1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 1 \\
\end{bmatrix}
\begin{bmatrix}
H'' \\
H' \\
\end{bmatrix}
\begin{bmatrix}
r - \ell \\
\ell \\
\end{bmatrix}
\begin{bmatrix}
s'' \\
s' \\
\end{bmatrix}
\]

Step 3

Repeat:
1. Permutation + partial Gaussian elimination
2. Find many \( e' \) such that \( H'e' = s' \)
3. For all good \( e' \), test \( \text{wt}(s'' + H''e') \leq w - p \)

Step 3. is (a kind of) Lee & Brickell which embeds Step 2

Step 2. is Birthday Decoding (or whatever is best)

Total cost is minimized over \( \ell \) and \( p \)
Information Set Decoding – Timeline

- Information Set Decoding: [Prange, 62]
- Relax the weight profile: [Lee & Brickell, 88]
- Compute sums on partial columns first: [Leon, 88]
- Use the birthday attack: [Stern, 89], [Dumer, 93]
- First “real” implementation: [Canteaut & Chabaud, 98]
- Initial McEliece parameters broken: [Bernstein, Lange, & Peters, 08]
- Lower bounds: [Finiasz & S., 09]
- Ball-collision decoding [Bernstein, Lange, & Peters, 11]
- Asymptotic exponent improved [May, Meurer, & Thomae, 11]
- Decoding one out of many [S., 11]
- Even better asymptotic exponent [Becker, Joux, May, & Meurer, 12]
Back to basics: which one is the best?

Answer: when there is a single solution, basic ISD (Prange or Lee & Brickell) always has a lower complexity than birthday decoding.

However, in the case of the embedded decoder the question is: what would be the cost per solution when there are many solutions to find?
When Do We Have Many Solutions?

\[ H = \begin{bmatrix} n \\ r \end{bmatrix}, \quad s = \begin{bmatrix} 1 \end{bmatrix} \]

\[ e = \begin{bmatrix} \text{Hamming weight } w \end{bmatrix} \]

When \( \binom{n}{w} > 2^r \), we expect \( N = \binom{n}{w}/2^r \) solutions
Fix the code size ($n$ and $r$) let the weight grow and observe the hardness of ISD (log scale) for one solution.

\[
\log_2(WF) = d_0 \log_2 \left( \frac{n}{r} \right)
\]

The maximum is reached for the Gilbert-Varshamov distance, that is when \(2^r \approx \binom{n}{d_0} \).
Birthday Decoding - Many Solutions Case

\[ H = \begin{pmatrix} H_1 & H_2 \end{pmatrix} \]

For a single solution (out many), we build 2 sets of size \(2^{r/2}\) instead of \(\sqrt{\binom{n}{w}}\). The cost drops to \(O(2^{r/2})\).

For many solutions, we keep lists of size \(\sqrt{\binom{n}{w}}\) and we obtain a constant proportion of the \(N = \binom{n}{w}/2^r\) solutions. The cost per solution is \(O\left(\frac{2^r}{\sqrt{\binom{n}{w}}}\right) = O\left(\frac{2^{r/2}}{\sqrt{N}}\right)\).
Let $N = \binom{n}{w}/2^r$. On iteration will cost $K = \binom{k}{p} + M(n)$. The probability to find any one solution is $\frac{\binom{r}{w-p}\binom{k}{p}}{2^r}$. The probability to find a particular solution is $\frac{\binom{r}{w-p}\binom{k}{p}}{\binom{n}{w}}$.

One solution will cost $\frac{2^r K}{\binom{r}{w-p}\binom{k}{p}}$ and $N$ solutions will cost $\frac{\binom{n}{w} K}{\binom{r}{w-p}\binom{k}{p}}$.

Finding $N$ solutions will cost $N$ times more. The cost per solution does not decrease.
Improving birthday decoding: Generalized Birthday Algorithm (GBA) [Wagner, 2002] and [Coron & Joux, 2004]

→ recent improvements [May, Meurer, & Thomae, 11] and [Becker, Joux, May, & Meurer, 12]
Can you use this to design better quantum decoding algorithm?
Thank you!