Explicit Bounds for Generic Decoding Algorithms for Code-Based Cryptography

Joint work with
Daniel J. Bernstein, Tanja Lange, and Henk van Tilborg

Christiane Peters

Technische Universiteit Eindhoven

WCC 2009
May 12, 2009

1. Introduction

2. Asymptotic complexity of the Lee–Brickell algorithm

3. Asymptotic complexity of Stern's algorithm

Decoding problem

We only consider binary codes, i.e., codes over $\mathbf{F}_2$. In particular, we consider codes with no obvious structure.

Classical decoding problem: find the closest codeword $\mathbf{x} \in C$ to a given $\mathbf{y} \in \mathbf{F}_2^n$, assuming that there is a unique closest codeword.

Berlekamp, McEliece, van Tilborg (1978) showed that the general decoding problem for linear codes is NP-complete.

McEliece PKC from an attacker's point of view

Given a $k \times n$ generator matrix G of a public code, and an error weight w.

To encrypt a message $\mathbf{m} \in \mathbf{F}_2^k$, the sender computes $\mathbf{m}G$, adds a random weight-w error vector $\mathbf{e}$, and sends $\mathbf{y} = \mathbf{m}G + \mathbf{e}$.

Not knowing the secret code and its decoding algorithm the attacker is faced with the problem of decoding $\mathbf{y}$ in a random-looking code.

McEliece proposed choosing random degree-t classical binary Goppa codes. The standard parameter choices are $k = n - t\lceil \lg n \rceil$ and $w = t$, typically with n a power of 2.

McEliece's original suggestion: $n = 1024$, $k = 524$, and $w = 50$.

Attacks on the McEliece PKC

Most effective attack against the McEliece cryptosystem is information-set decoding.

Many variants: McEliece (1978), Leon (1988), Lee and Brickell (1988), Stern (1989), van Tilburg (1990), Canteaut and Chabanne (1994), Canteaut and Chabaud (1998), and Canteaut and Sendrier (1998).

Bernstein, Lange, P. (PQCrypto 2008): improved Stern attack and broke original McEliece parameters

Note: some of the algorithms are used for decoding; some are minimum-weight-word-finding algorithms.
For comparison we rephrase all algorithms in terms of "fixed-distance decoding".

Fixed-distance decoding

A fixed-distance-decoding algorithm searches for a codeword at a fixed distance from a received vector.

Input: the received vector $\mathbf{y}$ and a generator matrix G for the code.

Output: a sequence of weight-w elements $\mathbf{e} \in \mathbf{y} - \mathbf{F}_2^k G$.

Note that the output consists of error vectors $\mathbf{e}$, rather than codewords $\mathbf{y} - \mathbf{e}$.

In the important special case $\mathbf{y} = 0$, a fixed-distance-decoding algorithm searches for codewords of weight w.

Information sets

Given a generator matrix G of an $[n, k]$ code.

An information set is a size-k subset $I \subseteq \{1, 2, \ldots, n\}$ such that the I-indexed columns of G are invertible.

Denote the matrix formed by the I-indexed columns of G by G_I.
The I-indexed columns of $G_I^{-1}G$ are the $k \times k$ identity matrix.

Let $\mathbf{y} \in \mathbf{F}_2^n$ have distance w to a codeword in $\mathbf{F}_2^k G$, i.e.,
$\mathbf{y} = \mathbf{c} + \mathbf{e}$ for a codeword $\mathbf{c} \in \mathbf{F}_2^k G$ and a vector $\mathbf{e}$ of weight w.

Denote the I-indexed positions of $\mathbf{y}$ by $\mathbf{y}_I$.

If $\mathbf{y}_I$ is error-free, $\mathbf{y}_I G_I^{-1}$ is the original message and
$\mathbf{c} = (\mathbf{y}_I G_I^{-1})G$. Thus, $\mathbf{e} = \mathbf{y} - (\mathbf{y}_I G_I^{-1})G$.

The Lee–Brickell algorithm

The algorithm consists of a series of independent iterations.
Each iteration contains the following steps.

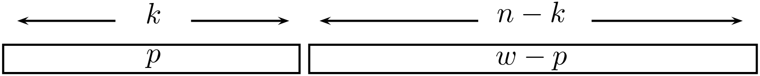
1. Select an information set $I \subseteq \{1, 2, \ldots, n\}$.
2. For each size-p subset $A \subseteq \{1, \ldots, k\}$: compute

$$\mathbf{e} = \mathbf{y} - (\mathbf{y}_I G_I^{-1})G - \sum_{a \in A} G_a,$$

   where G_a is the unique row of G in which column a has a 1; print $\mathbf{e}$ if it has weight w.

A weight-w error vector $\mathbf{e} \in \mathbf{y} - \mathbf{F}_2^k G$ is found by an information set I if and only if the I-indexed components of $\mathbf{e}$ have weight p, and the remaining components of $\mathbf{e}$ have weight $w - p$.

Model of the number of iterations (Lee–Brickell)



If $\mathbf{e}$ is a uniform random weight-w element of $\mathbf{F}_2^n$, and I is a size-k subset of $\{1,\ldots,n\}$, then $\mathbf{e}$ has probability exactly

$$\mathrm{LBPr}(n,k,w,p) = \frac{\binom{n-k}{w-p}\binom{k}{p}}{\binom{n}{w}}$$

of having weight exactly p on I.

Consequently the Lee–Brickell algorithm, given $\mathbf{c}+\mathbf{e}$ as input for some codeword $\mathbf{c}$, has probability exactly $\mathrm{LBPr}(n,k,w,p)$ of printing $\mathbf{e}$ in the first iteration.

Model of the total cost (Lee–Brickell)

The function LBCost defined as

$$\mathrm{LBCost}(n,k,w,p) = \frac{\frac{1}{2}(n-k)^2(n+k) + \binom{k}{p}p(n-k)}{\mathrm{LBPr}(n,k,w,p)}.$$

is a model of the average time used by the Lee–Brickell algorithm.

- The term $\frac{1}{2}(n-k)^2(n+k)$ is a model of row-reduction time;
- $\binom{k}{p}$ is the number of size-p subsets A of $\{1,2,\ldots,k\}$;
- and $p(n-k)$ is a model of the cost of computing $\mathbf{y} - \sum_{a \in A} G_a$.

Asymptotic analysis

Let R be the code rate and S the error fraction S; i.e., $k = Rn$ and $w = Sn$.

Goal: Measure the scalability of the information-set algorithm.

The simplest form of information-set decoding takes time $2^{(\alpha(R,S)+o(1))n}$ to find Sn errors in a dimension-Rn length-n binary code if R and S are fixed while $n \to \infty$; here

$$\alpha(R,S) = (1-R-S)\lg(1-R-S) - (1-R)\lg(1-R) - (1-S)\lg(1-S)$$

and $\lg$ means the logarithm base 2.

Stirling revisited

We assume that the code rate $R = k/n$ and error fraction $S = w/n$ satisfy $0 < S < 1 - R < 1$.

We put bounds on binomial coefficients as follows. Define $\epsilon(m)$ for each integer $m \geq 1$ by the formula

$$m! = \sqrt{2\pi}\ m^{m+1/2}\ e^{-m+\epsilon(m)}.$$

The classic Stirling approximation is $\epsilon(m) \approx 0$. Robbins showed that

$$\frac{1}{12m+1} < \epsilon(m) < \frac{1}{12m}. \tag{1}$$

Define $\mathrm{LBErr}(n, k, w, p)$ as

$$\frac{k!}{(k-p)!k^p} \frac{w!}{(w-p)!w^p} \frac{(n-k-w)!(n-k-w)^p}{(n-k-w+p)!} \frac{e^{\epsilon(n-k)+\epsilon(n-w)}}{e^{\epsilon(n-k-w)+\epsilon(n)}}.$$

Putting upper and lower bounds on $\mathrm{LBPr}(n, k, w, p)$

Define $\beta(R, S) = \sqrt{(1 - R - S)/((1 - R)(1 - S))}$.

Lemma

$\mathrm{LBPr}(n, k, w, p)$ *equals*

$$2^{-\alpha(R,S)n} \frac{1}{p!} \left(\frac{RSn}{1 - R - S}\right)^p \frac{1}{\beta(R, S)} \mathrm{LBErr}(n, k, w, p).$$

Furthermore

$$\frac{(1-\frac{p}{k})^p(1-\frac{p}{w})^p}{(1+\frac{p}{n-k-w})^p} e^{-\frac{1}{12n}(1 + \frac{1}{1-R-S})} < \mathrm{LBErr}(n, k, w, p) < e^{\frac{1}{12n}(\frac{1}{1-R} + \frac{1}{1-S})}.$$

Note that for fixed rate R, fixed error fraction S, and fixed p the error factor $\mathrm{LBErr}(n, nR, nS, p)$ is close to 1 as n tends to infinity.

Comparing Lee–Brickell for various p

Corollary

$\mathrm{LBCost}(n, Rn, Sn, 0) = (c_0 + O(1/n))2^{\alpha(R,S)n}n^3$ *as* $n \to \infty$ *where* $c_0 = (1/2)(1 - R)(1 - R^2)\beta(R, S)$.

Corollary

$\mathrm{LBCost}(n, Rn, Sn, 1) = (c_1 + O(1/n))2^{\alpha(R,S)n}n^2$ *as* $n \to \infty$ *where* $c_1 = (1/2)(1 - R)(1 - R^2)(1 - R - S)(1/RS)\beta(R, S)$.

Corollary

$\mathrm{LBCost}(n, Rn, Sn, 2) = (c_2 + O(1/n))2^{\alpha(R,S)n}n$ *as* $n \to \infty$ *where* $c_2 = (1 - R)(1 + R^2)(1 - R - S)^2(1/RS)^2\beta(R, S)$.

Corollary

$\mathrm{LBCost}(n, Rn, Sn, 3) = (c_3 + O(1/n))2^{\alpha(R,S)n}n$ *as* $n \to \infty$ *where* $c_3 = 3(1 - R)(1 - R - S)^3(1/S)^3\beta(R, S)$.

1. Introduction

2. Asymptotic complexity of the Lee–Brickell algorithm

3. Asymptotic complexity of Stern's algorithm

Stern's algorithm

Each iteration of Stern's algorithm contains the following steps.
Fix parameters ℓ and p. Typically ℓ is chosen close to $\binom{\lfloor k/2 \rfloor}{p}$.

1. Select an information set $I \subseteq \{1, 2, \ldots, n\}$.
2. Eliminate the I-indexed entries from $\mathbf{y}$, i.e., replace $\mathbf{y}$ by $\mathbf{y} - \mathbf{y}_I G_I^{-1} G$.
3. Select a uniform random size-$\lfloor k/2 \rfloor$ subset $X \subseteq I$.
4. Define $Y = I \setminus X$.
5. Select a uniform random size-ℓ subset $Z \subseteq \{1, 2, \ldots, n\} \setminus I$.
6. For each size-p subset $A \subseteq X$: Compute $\varphi(A) \in \mathbf{F}_2^\ell$, the Z-indexed entries of $\mathbf{y} - \sum_{a \in A} G_a$.
7. For each size-p subset $B \subseteq Y$: Compute $\psi(B) \in \mathbf{F}_2^\ell$, the Z-indexed entries of $\sum_{b \in B} G_b$.
8. For each pair (A, B) such that $\varphi(A) = \psi(B)$: Compute $\mathbf{e} = \mathbf{y} - \sum_{a \in A} G_a - \sum_{b \in B} G_b$; print $\mathbf{e}$ if it has weight w.

A weight-w error vector $\mathbf{e} \in \mathbf{y} + \mathbf{F}_2^k G$ is found by an information set Iif and only if it has weight p in the part corresponding to X, weight p in the part corresponding to Y, and weight 0 in the part corr. to Z.

Model of the number of iterations (Stern)

$\longleftarrow k \longrightarrow$		$\leftarrow \ell \rightarrow$	$\longleftarrow n-k-\ell \longrightarrow$
p	p	0	$w-2p$

If $\mathbf{e}$ is a uniform random weight-w element of $\mathbf{F}_2^n$, I is a size-k subset of $\{1, \ldots, n\}$, X is a size-$k/2$ subset of I, and Z is a size-ℓ subset of $\{1, 2, \ldots, n\} \setminus I$, then $\mathbf{e}$ has probability exactly

$$\mathrm{STPr}(n, k, w, \ell, p) = \binom{k/2}{p}^2 \binom{n-k-\ell}{w-2p} \bigg/ \binom{n}{w}$$

of having weights exactly p on X, p on $I \setminus X$, and 0 on Z.

Consequently the Stern algorithm, given $\mathbf{c} + \mathbf{e}$ as input for some code word $\mathbf{c}$, has probability exactly $\mathrm{STPr}(n, k, w, \ell, p)$ of printing $\mathbf{e}$ in the first iteration.

Model of the total cost

This function STCost is a model of the average time used by Stern's algorithm.

$$\mathrm{STCost}(n,k,w,\ell,p) = \frac{\frac{1}{2}(n-k)^2(n+k)+2\binom{k/2}{p}p\ell+2\binom{k/2}{p}^2 p(n-k)/2^\ell}{\mathrm{STPr}(n,k,w,\ell,p)}.$$

- the term $\frac{1}{2}(n-k)^2(n+k)$ is a model of row-reduction time;
- $\binom{k/2}{p}$ is the number of size-p subsets A of X;
- $p\ell$ is a model of the cost of computing $\varphi(A)$; $\binom{k/2}{p}$ is the number of size-p subsets B of Y;
- $p\ell$ is a model of the cost of computing $\psi(B)$.
- we use $\binom{k/2}{p}^2/2^\ell$ as a model for the number of colliding pairs (A,B).
- For each collision $2p(n-k)$ is a model of the cost of computing $\mathbf{y}-\sum_{a\in A} G_a - \sum_{b\in B} G_b$.

Bounds on $\mathrm{STPr}(n, k, w, \ell, p)$

Define error term as $\mathrm{STErr}(n, k, w, \ell, p) = \frac{e^{\epsilon(n-k)+\epsilon(n-w)}}{e^{\epsilon(n-k-w)+\epsilon(n)}}$

$$\cdot \left(\frac{(k/2)!}{(k/2-p)!(k/2)^p}\right)^2 \frac{w!}{(w-2p)!w^{2p}} \frac{(n-k-\ell)!(n-k)^\ell}{(n-k)!} \frac{(n-k-w)!}{(n-k-\ell-w+2p)!(n-k-w)^{\ell-2p}}.$$

Lemma

$\mathrm{STPr}(n, k, w, \ell, p)$ *equals*

$$2^{-\alpha(R,S)n} \frac{1}{(p!)^2} \left(\frac{RSV}{2(1-R-S)}\right)^{2p} \left(\frac{1-R-S}{1-R}\right)^{\ell} \frac{1}{\beta(R,S)} \mathrm{STErr}(n, k, w, \ell, p).$$

Furthermore

$$(1 - \tfrac{2p}{k})^{2p}(1 - \tfrac{2p}{w})^{2p}(1 - \tfrac{n-k-\ell-w+2p}{n-k-w})^p e^{-\frac{1}{12n}\left(1+\frac{1}{1-R-S}\right)} <$$

$$\mathrm{STErr}(n, k, w, \ell, p) < (1 + \tfrac{\ell-1}{n-k-\ell-1})^p e^{\frac{1}{12n}\left(\frac{1}{1-R}+\frac{1}{1-S}\right)}.$$

Choice of parameters (1)

- Most papers choose ℓ as $\lg \binom{k/2}{p}$ in order to balance $\binom{k/2}{p}$ with $\binom{k/2}{p}^2/2^\ell$.

- However, starting from this balance, increasing ℓ by 1 produces better results:
  it chops $2\binom{k/2}{p}^2 p(n-k)/2^\ell$ in half without seriously affecting $2\binom{k/2}{p}p\ell$ or $\mathrm{STPr}(n,k,w,\ell,p)$.

- Choosing ℓ close to $\lg \binom{k/2}{p} + \lg(n-k)$ would ensure that $2\binom{k/2}{p}p\ell$ dominates but would also significantly hurt STPr.

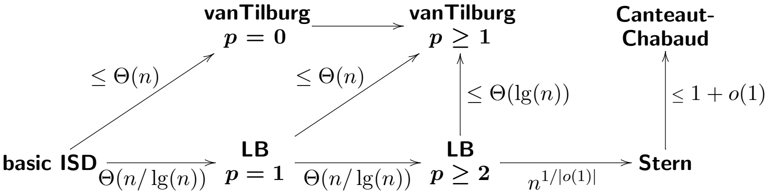
Choice of parameters (2)

- (With any reasonable choice of ℓ), increasing p by 1 means that the dominating term $2\binom{k/2}{p}p\ell$ increases by a factor of approximately $k/(2p)$ while the denominator $\mathrm{STPr}(n, k, w, \ell, p)$ increases by a factor of approximately $(k/2p)^2 w^2/(n - k - w)^2$.

- Overall $\mathrm{STCost}(n, k, w, \ell, p)$ decreases by a factor of approximately $(k/2p)w^2/(n - k - w)^2 = (R/2)(S/(1 - R - S))^2(n/p)$.

- The improvement from Lee–Brickell to Stern is therefore, for fixed R and S, more than any constant power of n.

Decoding complexity comparison



- There are several variants of information-set decoding designed to reduce the cost of row reduction, sometimes at the expense of success probability.
- These variants save a non-constant factor for Lee–Brickell (LB) but save at most a factor $1 + o(1)$ for Stern. The critical point is that row reduction takes negligible time inside Stern's algorithm, since p is large.

Thank you for your attention!