Statistical Decoding Revisited

R. Overbeck

GK Electronic Commerce,
TU-Darmstadt,
Department of Computer Science,
Cryptography and Computer Algebra Group
overbeck@cdc.informatik.tu-darmstadt.de

Abstract. In this paper we look at the statistical decoding attack on the McEliece cryptosystem from [4]. The statistical decoding algorithm is a probabilistic algorithm for correcting errors in random codes. It uses precomptuations to provide faster error correction than the classical general decoding algorithms. We analyze the success probability of the algorithm and show how to improve it. Further, we show that the algorithm may not be used to attack the McEliece cryptosystem, due to the large amount of precomputation needed.

Keywords: McEliece Cryptosystem, general decoding, coding theory, public key cryptography, code based cryptography.

1 Introduction

The security of cryptosystems based on error correcting codes is connected to the hardness of the general decoding problem. The first cryptosystem, which is based on that technique is the one presented by McEliece in 1978. McEliece's cryptosystem is very effective in en- and decryption, has a good information rate and we can even build a signature scheme from it. Furthermore, despite all effort, it remains unbroken for large public key sizes.

The statistical decoding attack on the McEliece PKC is a general decoding attack. It uses a precomputed alternative description of the public key, which has exponential space complexity. The author of [4] claims, that this alternative description can be computed in reasonable time. We show, that this is not possible employing the method proposed by Al Jabri. As a consequence, the attack fails even for the original parameter set of the McEliece cryptosystem, which is insecure against general decoding attacks [2].

However, statistical decoding can be used to correct errors in short random codes. After some precomputation, statistical decoding corrects errors more efficiently than the standard general decoding algorithms. Its mayor disadvantage is, that the algorithm is probabilistic and fails in some cases. We show how to improve the probability of correct decoding in that case.

The paper is structured as follows: In this section we give an introduction into the basic concepts of coding theory and the McEliece cryptosystem. In the

L. Batten and R. Safavi-Naini (Eds.): ACISP 2006, LNCS 4058, pp. 283–294, 2006.

second and third section we present the statistical decoding algorithm and show, how to improve it. In the fourth sections we analyze the precomputation phase of the statistical decoding algorithm.

1.1 Coding Theory and Problems

The security of the McEliece cryptosystem is based on the difficulty of some classical problems of coding theory. Here we give a short introduction into the topic of coding theory.

Definition 1.1. *An (n, k)-code $\mathcal{C}$ over a finite field $\mathbb{F}$ is a k-dimensional subvectorspace of the vector space $\mathbb{F}^n$. We call $\mathcal{C}$ an (n, k, d)-code if the minimum distance is $d = \min_{\mathbf{x},\mathbf{y}\in\mathcal{C}} \operatorname{dist}(\mathbf{x}, \mathbf{y})$, where "dist" denotes the Hamming distance. The distance of an element $\mathbf{x} \in \mathbb{F}^n$ to the null-vector $\operatorname{wt}(\mathbf{x}) := \operatorname{dist}(\mathbf{0}, \mathbf{x})$ is called weight of $\mathbf{x}$.*

Definition 1.2. *The matrix $\mathsf{C} \in \mathbb{F}^{k\times n}$ is a generator matrix for the (n, k) code $\mathcal{C}$ over $\mathbb{F}$, if the rows of C span $\mathcal{C}$ over $\mathbb{F}$. The matrix $\mathsf{H} \in \mathbb{F}^{(n-k)\times n}$ is called check matrix for the code $\mathcal{C}$ if $\mathsf{H}^\top$ is the right kernel of C. The code generated by H is called dual code of $\mathcal{C}$ and denoted by $\mathcal{C}^\perp$.*

With these definitions, we are able to define the problems of coding theory on which the security of the McEliece cryptosystem rely. The $\mathcal{NP}$-hardness of these problems is proven e.g. in [1].

Definition 1.3. *The general decoding problem for linear codes is defined as follows: For a given (n, k) linear code $\mathcal{C}$ over $\mathbb{F}$ and a vector $\mathbf{y} \in \mathbb{F}^n$ find $\mathbf{x} \in \mathcal{C}$, where $\operatorname{dist}(\mathbf{y}, \mathbf{x})$ is minimal.*

Let $\mathbf{e}$ be a vector of weight $\leq t := \left\lfloor \frac{d-1}{2} \right\rfloor$ and $\mathbf{x} \in \mathcal{C}$. Then there is a unique solution to the general decoding problem for $\mathbf{y} = \mathbf{x} + \mathbf{e}$. The code $\mathcal{C}$ is said to be an t-error correcting code.

Definition 1.4. *The problem of finding weights of a linear code is defined as follows: For a given linear code $\mathcal{C}$ over $\mathbb{F}$ and $w \in \mathbb{N}$ find a vector $\mathbf{x} \in \mathcal{C}$ with weight w.*

Throughout this paper, we will use the following notation. We write $\mathcal{G} = \langle \mathsf{G} \rangle$ if the linear (n, k)-code $\mathcal{G}$ over $\mathbb{F}$ has the generator matrix G. We can write $\mathbf{x} \in \mathcal{G}$ as $(x_1, \cdots, x_n) \in \mathbb{K}^n$. For any (ordered) subset $\{j_1, \cdots j_m\} = J \subseteq \{1, \cdots n\}$ we denote the vector $(x_{j_1}, \cdots, x_{j_m}) \in \mathbb{K}^m$ with $\mathbf{x}_J$. Similarly we denote by $\mathsf{M}_{\cdot J}$ the submatrix of a $k \times n$ matrix M consisting of the columns corresponding to the indices of J and $\mathsf{M}_{J'\cdot} = \left(\mathsf{M}^\top\right)_{\cdot J'}$ for any (ordered) subset J' of $\{1, \cdots, k\}$.

1.2 The McEliece PKC

This cryptosystem was proposed by McEliece [5] and is the first, which uses error correcting codes as a trapdoor. It remains unbroken in its original version, which uses irreducible binary Goppa codes. There exist efficient algorithms to correct errors up to half of the designed minimum distance of the Goppa code. We briefly describe the cryptosystem:

- **System Parameters:** $n = 2^m$, $t \in \mathbb{N}$, where $t \ll n$.
- **Key Generation:** Given the parameters n, t generate the following matrices:
  G' : $k \times n$ generator matrix of a binary irreducible
  $(n, k = 2^m - mt, 2t + 1)$ Goppa code $\mathcal{G}$
  S : $k \times k$ random binary non-singular matrix
  P : $n \times n$ random permutation matrix
  Then, compute the $k \times n$ matrix $\mathsf{G} = \mathsf{S}\mathsf{G}'P$.
- **Public Key:** (G, t)
- **Private Key:** $(\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$, where $D_{\mathcal{G}}$ is an efficient decoding algorithm for $\mathcal{G}$.
- **Encryption:** To encrypt a plaintext $\mathbf{m} \in \{0,1\}^k$ choose a vector $\mathbf{z} \in \{0,1\}^n$ of weight t randomly and compute the ciphertext $\mathbf{c}$ as follows:

$$\mathbf{c} = \mathbf{m}\mathsf{G} \oplus \mathbf{z} \ .$$

- **Decryption:** To decrypt a ciphertext $\mathbf{c}$ calculate

$$\mathbf{c}\mathsf{P}^{-1} = (\mathbf{m}\mathsf{S})\,\mathsf{G}' \oplus \mathbf{z}\mathsf{P}^{-1}$$

first, and apply the decoding algorithm $\mathcal{D}_{\mathsf{G}}$ for $\mathcal{G}$ to it. Since $\mathbf{c}\mathsf{P}^{-1}$ has a hamming distance of t to the Goppa code we obtain the codeword

$$\mathbf{m}\mathsf{S}\mathsf{G}' = \mathcal{D}_{\mathcal{G}}\left(\mathbf{c}\mathsf{P}^{-1}\right) \ .$$

Let $J \subseteq \{1, \cdots, n\}$ be a set, such that $\mathsf{G}_{\cdot J}$ is invertible, then we can compute the plaintext $\mathbf{m} = (\mathbf{m}\mathsf{S}\mathsf{G}')_J\,(\mathsf{G}'_{\cdot J})^{-1}\,S^{-1}$

In its initial version from 1978, McEliece proposed to choose $m = 10$ and $t = 50$, i.e. using a $(1024, 524, 101)$ Goppa code. After the proposal of several general decoding attacks the parameters had to be modified. The fastest of these attacks was proposed in [2], compare section 4 and 5. Today parameter sets with $m = 11$ and $40 \leq t \leq 93$ are considered to be secure. There exists also a signature scheme based on the McEliece PKC (CFS, see [3]), which is as secure as the McEliece PKC with the same parameters. For CFS it is proposed to choose e.g. $m = 16$ and $t = 9$.

2 Statistical Decoding

This general decoding algorithm was presented by A Kh. Al Jabri in [4]. The idea of statistical decoding may be described as follows: Given an (n, k, d) code $\mathcal{G}$, we first compute a sufficiently large set $\mathcal{H}_w$ of dual vectors of weight w (i.e. an alternative description of $\mathcal{G} = \mathcal{H}_w^{\perp}$). In the following we assume that $w < n/2$. However all observations are analogous for $w > n/2$. Given a word $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where $\mathbf{x} \in \mathcal{G}$ and $\mathrm{wt}(\mathbf{e})$ is small, we take a vector $\mathbf{h} \in \mathcal{H}_w$, where $\mathbf{y}\mathbf{h}^\top \neq 0$. As $\mathbf{x}\mathbf{h}^\top = 0$, the non-zero positions of $\mathbf{h}$ reveals some information about $\mathbf{e}$. (Let e.g. $\mathrm{wt}(\mathbf{e}) = 4$, then either one or three non-zero entries of $\mathbf{e}$ correspond to non-zero entries of $\mathbf{h}$). Collecting the information each of the different vectors $\mathbf{h} \in \mathcal{H}_w$ reveals, we are able to find $\mathbf{e}$ in some cases.

There are three major questions regarding this technique, which we will address in the following sections: “How to compute the set $\mathcal{H}_w$?” (section 4), “How to combine the information the vectors of $\mathcal{H}_w$ reveal about $\mathbf{e}$?” (this section) and “What is the probability of identifying $\mathbf{e}$?” (section 3). In section 3.2 we show how to improve the success probability of correct decoding.

Let $\mathcal{H}_w$ be a set of vectors of weight w of the dual space of the $(n, k, 2t+1)$ linear binary code $\mathcal{G}$ with generator Matrix G. Let $\mathbf{y}$ be the sum of a codeword $\mathbf{u}\mathsf{G} \in \mathcal{G}$ and a error vector $\mathbf{e}$ with weight at most t. A Kh. Al Jabri points out, that for randomly generated codes the probability that a value of 1 appears in the i-th position of $\mathbf{h} \in \mathcal{H}_w$ with $\mathbf{y}\mathbf{h}^T = 1$ depends on i being a erroneous position in the vector $\mathbf{y}$. We have an *odd error detection* in i if $\mathbf{y}\mathbf{h}^T = 1$ and $\mathbf{h}_i = 1$. Assume that we have an odd error detection in i, then let p_w^+ be the probability that i is a erroneous position and q_w^+ be the probability that i is a non-erroneous position. We can compute these probabilities as

$$p_w^+ = \frac{\sum_{m \text{ odd}}^{m \le t} \binom{n-t}{w-m}\binom{t-1}{m-1}}{\sum_{m \text{ odd}}^{m \le t} \binom{t}{m}\binom{n-t}{w-m}}, \; q_w^+ = \frac{\sum_{m \text{ odd}}^{m \le t} \binom{n-t-1}{w-m-1}\binom{t}{m}}{\sum_{m \text{ odd}}^{m \le t} \binom{t}{m}\binom{n-t}{w-m}}.$$

Since $w < n/2$ the inequation $p_w^+ > q_w^+$ holds, although for large w the difference is small. We define $v_{\mathbf{y},w}^+ := \sum_{\mathbf{h} \in \mathcal{H}_w} \left(\mathbf{y}\mathbf{h}^T \mod 2\right)$. Then, for $i \in \{1, \cdots, n\}$ an (non-)error position the random variable

$$\frac{1}{v_{\mathbf{y},w}^+} \sum_{\mathbf{h} \in \mathcal{H}_w} \left(\mathbf{y}\mathbf{h}^T \mod 2\right) \mathbf{h}_i$$

is the relative frequency estimate for p_w^+ (q_w^+ respectively). Its variance is $\left(\sigma_w^+\right)^2 = p_w^+(p_w^+ - 1)/v_{\mathbf{y},w}^+$. Thus, we can recover $\mathbf{u}$ using algorithm 2.1 if $\mathcal{H}_w$ is chosen in a way so that we can distinguish between p_w^+ and q_w^+.

Algorithm 2.1. StatDec

```
Input: $\mathcal{H}_w$, $\mathbf{y}$.
Output: $\mathbf{u}$, the information vector.

$\mathbf{v} = \sum_{\mathbf{h} \in \mathcal{H}_w} \left(\mathbf{y}\mathbf{h}^\top \mod 2\right) \mathbf{h} \in \mathbb{Z}^n$.

choose $I = \{\text{positions of the } k \text{ smallest entries of } \mathbf{v}\}$ s.t. $\mathsf{G}_{\cdot I}$ is invertible.

$\mathbf{u} = \mathbf{y}_I \mathsf{G}_{\cdot I}^{-1}$
```

Al Jabri claims, that precomputing a set $\mathcal{H}_w$ with

$$|\mathcal{H}_w| = 625 \cdot 10^{-6} \cdot p_w^+ \left(1 - p_w^+\right) \epsilon^{-2} \tag{1}$$

vectors is sufficient for correct decoding [4]. The work factor for algorithm 2.1 is

$$\mathcal{O}\left(n \cdot |\mathcal{H}_w| + 2k^3 + kn\right)$$

binary operations having computed the set $\mathcal{H}_w$ in advance. The author of [4] claims that the latter can be done e.g. by the methods of [2]. However, computing the set $\mathcal{H}_w$ is solving problem 1.4, which is a $\mathcal{NP}$-hard problem in general. In addition, a set $\mathcal{H}_w$ of the desired size will not even exist if w is chosen too small. Goppa codes, as BCH codes and GRS codes have a weight distribution "close" to the expected weight distribution of random code, which is the binomial distribution [4]. Consequently, we get the following condition for $\mathcal{H}_w$:

$$|\mathcal{H}_w| \leq \binom{n}{w} 2^{-k}, \tag{2}$$

if we want to decode e.g. a random code or a Goppa code. We come back to this problem in section 4, but first we want to analyze the success probability of StatDec.

3 The Success Probability of Statistical Decoding

The first point of critique on the statistical decoding is its success probability. In our experiments for small parameter sets we had difficulties, to correct errors with a set $\mathcal{H}_w$ of size given in equation (1). It seems, that the set has to be about 2^{13} times larger than claimed by Al Jabri to allow correct decoding in most cases. We give a brief example: For a $(2^6, 40, 9)$ Goppa code (or a $(2^6, 40, 9)$ random code), Al Jabri's bound for $|\mathcal{H}_{17}|$ is $1 \leq |\mathcal{H}_{17}| \leq \binom{64}{17} 2^{-40} \approx 2^{10}$. However, one vector of the dual code can not be sufficient for correct decoding in most cases. Therefore we want to take a closer look at the success probability of statistical decoding. Later we show how to improve StatDec and give a small example.

In the following, we assume, that every set $\mathcal{H}_w$ consists of random vectors of weight w. If the vectors in $\mathcal{H}_w$ are somehow related, the probability for finding the correct error vector decreases.

3.1 The Initial Algorithm

We return to the notations of the previous section. On input $\mathcal{H}_w$ and $\mathbf{y}$ StatDec does only return the correct error vector if for some δ with $p_w^+ - 1 < \delta < p_w^+$ the following two conditions hold:

(i) For every error position i:

$$\mathbf{v}_i > (p_w^+ - \delta) v_{\mathbf{y},w}^+.$$

(ii) There are at least k non-error positions j, such that

$$\mathbf{v}_j < (p_w^+ - \delta) v_{\mathbf{y},w}^+.$$

We may assume, that $v^+_{\mathbf{y},w} \approx \frac{1}{2}\left|\mathcal{H}_w\right|$, and thus the probability, that a certain δ fulfills the first condition is smaller than

$$\mathcal{P} := \Phi\left(\delta/\sigma^+_w\right)^t = \Phi\left(\delta\sqrt{\frac{\frac{1}{2}\left|\mathcal{H}_w\right|}{p^+_w(p^+_w-1)}}\right)^t, \tag{3}$$

where Φ refers to the distribution function of the standardized normal distribution. Thus, we have to choose

$$2\left(\Phi^{-1}\left(\mathcal{P}^{1/t}\right)\right)^2 p^+_w(1-p^+_w)\delta^{-2} \leq \left|\mathcal{H}_w\right| \leq \binom{n}{w}2^{-k}. \tag{4}$$

Assume $k \approx (n-t)/2$, then it is probable, that half of the values $\mathbf{v}_j$ for non error positions j will be below their mean value $p^+_w v^+_{\mathbf{y},w}$. Thus, if there exists an δ for a given ciphertext y, such that the two conditions above are fulfilled, then it will probably be smaller than $|p^+_w - q^+_w|$. Since $\Phi^{-1}(0.95) = 1.65$ we conclude, that with a set of size

$$\left|\mathcal{H}_w\right| \approx 5.4 p^+_w(1-p^+_w)\frac{1}{(p^+_w-q^+_w)^2}. \tag{5}$$

we can correct errors with a probability about 0.95^t. Note, that this number is a factor 2^{13} larger than the one given by Al Jabri. We expect that with a set of size given in equation (1) we could correct errors with a probability about $1/2^t$, only.

3.2 An Improved Version

To improve the probability of correct error correction, we want to include *even error detection.* Let $\mathbf{y}$ be the sum of a codeword $\mathbf{u}\mathsf{G} \in \mathcal{G}$ and a error vector $\mathbf{e}$ with weight at most t. We observe, that for randomly generated codes the probability that a value of 1 appears in the i-th position of $\mathbf{h} \in \mathcal{H}_w$ with $\mathbf{y}\mathbf{h}^T = 0$ depends on i being a erroneous position in the vector $\mathbf{y}$. Thus, we have an even error detection if $\mathbf{y}\mathbf{h}^T = 0$ and $\mathbf{h}_i = 1$. Let p^-_w be the probability that i is a erroneous position and q^-_w be the probability that i is a non-erroneous position in the case of an even error detection. These probabilities can be computed as follows:

$$p^-_w = \frac{\sum_{2\leq m \text{ even}}^{m\leq t}\binom{n-t}{w-m}\binom{t-1}{m-1}}{\sum_{m \text{ even}}^{m\leq t}\binom{t}{m}\binom{n-t}{w-m}}, \quad q^-_w = \frac{\sum_{m \text{ even}}^{m\leq t}\binom{n-t-1}{w-m-1}\binom{t}{m}}{\sum_{m \text{ even}}^{m\leq t}\binom{t}{m}\binom{n-t}{w-m}}.$$

We define $v^-_{\mathbf{y},w} := \sum_{\mathbf{h}\in\mathcal{H}_w}\left(1 - \mathbf{y}\mathbf{h}^T \mod 2\right)$. Then, for an (non-)error position i the value

$$\frac{1}{v^-_{\mathbf{y},w}}\sum_{\mathbf{h}\in\mathcal{H}_w}\left(1 - \mathbf{y}\mathbf{h}^T \mod 2\right)\mathbf{h}_i$$

is the relative frequency estimate for p^-_w (q^-_w respectively). We observe, that if $p^+_w > q^+_w$, then $p^-_w < q^-_w$.

For all possible weights, the relative frequency estimates of p_w^+ and p_w^- are approximately normal distributed if $|\mathcal{H}_w|$ is large enough. Therefore we can use the standard transformation, s.t. all the relative frequency estimates are $\mathcal{N}(0,1)$ distributed. It follows, that one can sum the scaled relative frequency estimates obtained by several sets containing dual vectors of different weights. As a consequence, we consider $\mathcal{H}$ as the set of all dual vectors of weight w satisfying $b \leq w \leq B < n/2$, i.e. $\mathcal{H} = \bigcup_{w=b}^{B} \mathcal{H}_w$. All in all, we get the modified algorithm 3.1. With the notation of StatDec+: If i is an error position, then for all $\mathbf{v}$, $(\mathbf{v})_i$ has mean value 0. For an implementation one should omit the previous computation of σ_w^+ and σ_w^-. and compute these values while computing $\mathbf{v}_w$.

Algorithm 3.1. StatDec+

```
Input: $\mathcal{H} = \bigcup_{w=b}^{B} \mathcal{H}_w$, $\mathbf{y}$.
Output: $\mathbf{u}$, the information vector.

for $w = b$ to $B$ do
    $\left(\sigma_w^+\right)^2 = p_w^+ \cdot (1 - p_w^+) \cdot v_{\mathbf{y},w}^+$.
    $\left(\sigma_w^-\right)^2 = p_w^- \cdot (1 - p_w^-) \cdot v_{\mathbf{y},w}^-$.

$\mathbf{1} = (1, 1, \cdots, 1) \in \{0,1\}^n$.
for $w = b$ to $B$ do
    $\mathbf{v}_w = \sum_{\mathbf{h} \in \mathcal{H}_w} \left(\mathbf{y}\mathbf{h}^\top \mod 2\right) (\mathbf{h} - p_w^+ \mathbf{1})/\sigma_w^+ \in \mathbb{R}^n$.
    $\mathbf{v}_{w+B} = -\sum_{\mathbf{h} \in \mathcal{H}_w} \left(1 - \mathbf{y}\mathbf{h}^\top \mod 2\right) (\mathbf{h} - p_w^- \mathbf{1})/\sigma_w^- \in \mathbb{R}^n$.

for all binary combinations $\mathbf{v}$ of the different $\mathbf{v}_l$ do
    choose $I$ = {positions of the $k$ smalles entries of $\mathbf{v}$} s.t. $\mathsf{G}_{\cdot I}$ is invertible.
    $\mathbf{u} = \mathbf{y}_I \mathsf{G}_{\cdot I}^{-1}$
    if weight$(\mathbf{u}\mathsf{G} \oplus \mathbf{y}) \leq t$ then
        return $\mathbf{u} = \mathbf{u}$
```

Let us assume, that the different relative frequency estimates are independent. We define $\mathbf{v} = \sum_{w=b}^{B} e_w \mathbf{v}_w + \sum_{w=b}^{B} e_{w+B} \mathbf{v}_{w+B}$, where each $e_i \in \{0,1\}$. Then for an error position j, $(\mathbf{v})_j$ is normal distributed with mean value 0 and variance σ^2 equal to the number of $e_w \neq 0$. If j is a non-error position, then $(\mathbf{v})_j$ is normal distributed with mean value

$$E := \sum_{w=b}^{B} e_w \left(\frac{|q_w^+ - p_w^+|}{\sigma_w^+} v_{\mathbf{y},w}^+ \right) + \sum_{w=b}^{B} e_{w+B} \left(\frac{|q_w^- - p_w^-|}{\sigma_w^-} v_{\mathbf{y},w}^- \right)$$

and variance

$$S^2 = \sum_{w=b}^{B} w_w \left(\frac{q_w^+(1 - q_w^+)}{\left(\sigma_w^+\right)^2} v_{\mathbf{y},w}^+ \right) + \sum_{w=b}^{B} w_{w+B} \left(\frac{q_w^-(1 - q_w^-)}{\left(\sigma_w^-\right)^2} v_{\mathbf{y},w}^- \right)$$

In most cases we will have $2v_{\mathbf{y},w}^+ \approx 2v_{\mathbf{y},w}^- \approx |\mathcal{H}_w|$. To distinguish between error and non-error positions by $\mathbf{v}$, we get the following conditions: There exists an

$\delta \in \mathbb{R}$ such, that for every error position i the inequation $|\mathbf{v}_i| < \delta$ holds and there are at least k non-error positions j, such that $|\mathbf{v}_j| > \delta$. The probability, that a certain δ fulfills this conditions is smaller than $\Phi\left(\delta/\sigma\right)^t$. Again, we expect, that the condition $\delta \leq E$ has to be true in most cases, and thus we get

$$\mathcal{P} \approx \Phi\left(\frac{1}{\sigma}\left(\sum_{w=b}^{B} e_w \sqrt{\frac{\left|q_w^+ - p_w^+\right|^2 |\mathcal{H}_w|}{2p_w^+(1-p_w^+)}} + \sum_{w=b}^{B} e_{w+B} \sqrt{\frac{\left|q_w^- - p_w^-\right|^2 |\mathcal{H}_w|}{2p_w^-(1-p_w^-)}}\right)\right)^t$$

as a suitable estimate for the probability of correct decoding with StatDec+. However we are not able to prove, that the different relative frequency estimates for p_w^+ and q_w^+ are independent. Nevertheless, for an implementation it seems recommendable, to start with the vectors $\mathbf{v}$ where $|\{e_i \neq 0\}|$ is large.

3.3 Experimental Results

We made several experiments with codes of small length. As expected, the proposed variant StatDec+ of the initial algorithm allows error correction in a significant larger number of cases than StatDec, especially when the size of the sets $\mathcal{H}_w$ is small. Further, it seems recommendable to include sets $\mathcal{H}_w$ with small w, even if their size is smaller than desired (e.g. up to a factor 4).

In the following we present three examples of our experiments. Note that for all our examples the bound for $|\mathcal{H}_w|$ given by equation (1) is useless, as it is smaller than 0. Further, the precomputation to find the sets $\mathcal{H}_w$ was quite time-consuming and an exhaustive search in some cases. The time needed to perform the precomputation for StatDec+ is the same as for StatDec.

In our first example we considered a $(2^6, 40, 9)$ Goppa code. For this code the relative frequency estimates and the desired sizes of each $\mathcal{H}_w$ are given by table 3.1. We computed a set $\mathcal{H} = \{\mathcal{H}_{16}, \mathcal{H}_{17}, \mathcal{H}_{18}\}$, where each of the sets $\mathcal{H}_w$ consisted of 100 random vectors. With StatDec+ we were able to correct errors of weight 4 in 93.2% of the cases. With the original algorithm, called with each set $\mathcal{H}_w$, correct error correction was possible in 17.5% of the cases, only.

Table 3.1. Correcting errors of weight 4 in a $(64, 40)$ code

w	p_w^+	q_w^+	p_w^-	q_w^-	$\lvert\mathcal{H}_w\rvert$
16	0.295	0.248	0.210	0.263	1433
17	0.302	0.263	0.232	0.268	2160
18	0.311	0.280	0.254	0.284	3393

In the second example, we looked at the same code as in the first example, but chose each $\mathcal{H}_w$ to be the set of all vectors of weight w. For our particular Goppa code, we got: $|\mathcal{H}_{16}| = 345$, $|\mathcal{H}_{17}| = 1234$ and $|\mathcal{H}_{18}| = 3149$. In this case, error correction was possible with StatDec and StatDec+ in all cases. An correct error correction with StatDec would not have been possible in all cases, if only one of the sets $\mathcal{H}_w$ would have been used.

Table 3.2. Correcting errors of weight 6 in a $(64, 22)$ code

w	p_w^+	q_w^+	p_w^-	q_w^-	$\lvert\mathcal{H}_w\rvert$	StatDec success rate
8	0.183	0.119	0.082	0.129	562	95.0%
9	0.189	0.136	0.102	0.145	835	79.4%
10	0.196	0.152	0.122	0.160	1283	73.8%

In our last example, we looked at a $(2^6, 22, 13)$ random code. The values for the relative frequency estimates and the sizes of $\mathcal{H}_w$ resulting from equation (5) are given by table 3.2. The expected success probability of StatDec is $\approx 0.95^6 = 73.5\%$ for each set $\mathcal{H}_w$. However, the experimented success probability for StatDec is larger, compare table 3.2. In this case we were able to compute the desired sets in reasonable time. Again, we made 1000 attempts to correct errors of weight 6. With StatDec+ we were able to correct all errors, whereas with StatDec we would have been able to correct them in 99.2% of the cases.

4 On the Problem of Finding Weights

Al Jabri proposes to use a variant of Sterns algorithm to solve the problem of finding weights, i.e. to compute $\mathcal{H}_w$. J. Stern designed his algorithm to find a (unique) shortest codeword of a binary linear code. Such an algorithm can be used to correct up to $t := (d-1)/2$ errors in a binary (n, k, d) code $\mathcal{G}$: Let c be a binary n-vector with distance t to $\mathcal{G}$ and G be the generator matrix of $\mathcal{G}$. Then the sum of c and the unique shortest codeword of the code generated by

$$\left(\frac{G}{c}\right)$$

is the solution to the general decoding problem for G and c.

We recall the original algorithm of Stern [7], which tries to find a vector of low weight w. Let H be the check matrix of the code G. Given the parameters p and l, successively choose two disjoint sets of $p < k/2$ columns $\mathcal{I}_1$ and $\mathcal{I}_2$ at random. Then choose a set $\mathcal{J} \subseteq \{1, \cdots, n\} \setminus (\mathcal{I}_1 \cup \mathcal{I}_2)$ of l rows at random. We may assume without loss of generality, that $\mathcal{I}_1 = \{n-k+1, \cdots, n-k/2\}$ and $\mathcal{I}_1 = \{n-k/2+1, \cdots, n\}$. If we can not transform the matrix H into a systematic matrix, the algorithm fails at this point, and is started anew. Otherwise we transform H into the desired form. Now we may assume, that $\mathcal{J} = \{1, \cdots, l\}$ and get a check matrix of the following form:

$$\mathsf{H} = \left(\mathsf{Id}_{n-k} \left| \begin{array}{c} \mathsf{Z}_1 | \mathsf{Z}_2 \\ \hline \mathsf{B} \end{array} \right. \right),$$

where Z_1 and Z_2 are $l \times k/2$ matrices, and B is a $(k-l) \times k$ matrix. For all pairs of vectors $(\mathbf{e}_1, \mathbf{e}_2) \in (\ \{0,1\}^{k/2}\ )^2$ where $\mathrm{wt}(\mathbf{e}_1) = \mathrm{wt}(\mathbf{e}_2) = p$ we check whether $\mathbf{e}_1\mathsf{Z}_1 = \mathbf{e}_2\mathsf{Z}_2$. If the condition is fulfilled for such a pair, then we compute the unique vector $\mathbf{e}_0 \in \{0,1\}^{n-k}$, such that $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)\,\mathsf{H}^\top = 0$. The vector

$\mathbf{e} = (\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ is our candidate for a short codeword. One can observe, that the fist l entries of $\mathbf{e}$ are zeros and thus the weight of $\mathbf{e}$ is smaller than $n-k-l+2p$. If none of the constructed vectors $\mathbf{e}$ is of the desired weight, then the algorithm fails. The success probability of one iteration of the algorithm is

$$\mathcal{P}_{p,l,w} = \frac{\binom{n-w}{k/2-p}\binom{w}{p}\binom{n-w-k/2-p}{k/2-p}\binom{w-p}{p}\binom{n-k-(w-2p)}{l}}{\binom{n}{k/2}\binom{n-k/2}{k/2}\binom{n-k}{l}}$$

in the case of a unique code word $\mathbf{e}'$ of weight w.

To improve the performance of Sterns algorithm, one can view its dual variant – depending on the ratio of k/n – and try to avoid the costly Gaussian elimination by choosing $\mathcal{I}_1$ and $\mathcal{I}_2$ iteratively and not at random. This method was introduced and analyzed by Canteaut and Chabaud, compare [2]. The success probability of the algorithm for finding the shortest codeword is to be modeled by a Markov chain in that case. We omit details and just take the result, that the work factor for one iteration becomes

$$\Omega_{p,l} = \left(\frac{1}{2}n(n-k) + 2l\binom{k/2}{p}(p-1) + (n-k-l)(2p-1)\binom{k/2}{p}^2\frac{1}{2^l}\right).$$

The work factor of the resulting algorithm may be approximated by

$$\mathcal{O}(n^3)2^{-t\log_2(1-k/n)},$$

if t is small and k/n is not too close to 1 (compare [6]).

In the case of statistical decoding we use the algorithm from [2] not to find a single lowest weight code word, but several code words of a certain weight w. If there are several code words of weight w, the work factor decreases by a factor equal to the number of such code words. As the expected number of vectors of weight w is given by the binomial distribution, we get the expected workfactor to compute a set $\mathcal{H}_w$ of vectors of weight w as

$$\mathcal{W}_{p,l,w} = \frac{2^k}{\binom{n}{w}}\frac{\Omega_{p,l}}{\mathcal{P}_{p,l,w}} \cdot \sum_{i=0}^{|\mathcal{H}_w|-1}\left(1 - \frac{i\cdot 2^k}{\binom{n}{w}}\right)^{-1}. \qquad (6)$$

If one wants to compute a set $\mathcal{H}$, which serves as an input for the StatDec+, we expect, that every execution of a single round of the algorithm returns

$$\sum_{w=b}^{B}\frac{2^k}{\binom{n}{w}}\mathcal{P}_{p,l,w}^{-1}$$

vectors of weight w satisfying $b \leq w \leq B$. However, using the algorithm from [2] might not always be the best choice to use, when trying to find multiple words of any given weight, even if we did not find a better way to do so.

Unfortunately we were not able to find an example parameter set, where the precomputation required for StatDec could be performed in less time than the one needs for a single call of Canteaut's and Chabaud's general decoding algorithm.

5 Attacking the McEliece PKC with Statistical Decoding

To our knowledge, the best way to attack the McEliece PKC is the attack proposed by Canteaut and Chabaud [2], see section 4. Since for the McEliece cryptosystem $n = 2^m$ and $k = n - tm$, N. Sendrier concludes, that the maximum degree of security for the McEliece cryptosystem against the general decoding attack from [2] is obtained for an information rate $k/n \approx 1 - 1/\exp(1)$ [6]. This would lead e.g. to the choice of $m = 11$ and $t \approx 70$ for the McEliece cryptosystem.

To attack the McEliece PKC with parameters $m = 10$ and $t = 50$ with statistical decoding, Al Jabri claims that computing a set $\mathcal{H}_w$ consisting of 2^{38} vectors is sufficient. Unfortunately Al Jabri does not name w, but we are quite sure, that he referred to the set $\mathcal{H}_{133}$. However, equation (3) implies, that the probability of correct decoding is about 2^{-50} in that case. A decoding attempt with StatDec takes 2^{48} binary operations for this input. Consequently, one would expect, that it would take approximately 2^{98} binary operations, before an attack on one of 2^{50} given ciphertexts is successful.

We have shown, that an attacker would need a set $\mathcal{H}_{137}$ consisting of approximately 2^{51} vectors to attack ciphertext of the McEliece PKC with parameters $n = 10$ and $t = 50$. Even storing a set of this size seems impossible nowadays and the work factor for a single decoding attempt would be larger than 2^{61}, which is not much faster than the general decoding algorithm of Canteaut and Chabaud [2]. However, it takes at least 2^{152} binary operations to compute the set $\mathcal{H}_{137}$ with the algorithm proposed by Canteaut and Chabaud. For this parameter set, one iteration for $l = 19$ and $p = 2$ of the algorithm requires about 2^{24} binary operations. Most of the vectors returned by the algorithm will be of weight 241. For each one of 2^{-17} iterations, we will get only one of those vectors. Thus, after performing 2^{80} Operations, one will still have computed less than 2^{39} vectors of weight 241. Having a range of $114 \leq w \leq 241$, we will have still have not enough vectors of the dual space to attack the McEliece cryptosystem. Thus, it is not possible to attack the McEliece cryptosystem with StatDec or StatDec+.

Table 5.3. StatDec for example parameter sets

McEliece parameters	w	$\lvert p_w^+ - q_w^+\rvert$	$\lvert\mathcal{H}_w\rvert$	$\binom{n}{w}2^{-k}$	Workfactor	
$(2^m, k, d = 2t + 1)$					StatDec	precomput.
$(1024, 524, 101)$	137	$0.2 \cdot 10^{-7}$	2^{51}	$2^{52.5}$	2^{61}	2^{152}
$(1024, 524, 101)$	153	$0.21 \cdot 10^{-8}$	2^{58}	2^{94}	2^{68}	2^{138}
$(2048, 1278, 141)$	363	$0.41 \cdot 10^{-14}$	2^{96}	$2^{96.9}$	2^{107}	2^{609}
$(65536, 65392, 9)$	32000	$0.17 \cdot 10^{-13}$	2^{93}	$2^{109.7}$	2^{109}	$>> 2^{131}$

The situation for the signature scheme CFS is the same: Any set, that would allow correct decoding in a non-negligible fraction of the cases is to big to be stored efficiently and it is infeasible to perform the precomputation (compare Table 5.3).

6 Conclusion

We have shown, how to improve the probability of correct error correction of the statistical decoding algorithm. We have performed experiments and have shown, that statistical decoding can be used for fast decoding of random linear codes after some precomputation. Nevertheless, we needed several sets of vectors, each about 2^{13} times larger than claimed by Al Jabri. Additionally the problem how to perform the precomputation efficiently remains unsolved. We conclude, that it is not possible to attack the McEliece cryptosystem (or the CFS signature scheme) with reasonable parameter sets by statistical decoding. However, there might exist non-standard parameter sets for the McEliece cryptosystem, which can be attacked by statistical decoding.

References

1. E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
2. A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEETIT: IEEE Transactions on Information Theory*, 44, 1998.
3. N. Courtois, M. Finiasz, and N.Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248, pages 157–174. Springer-Verlag, 2001.
4. A. Al Jabri. A statistical decoding algorithm for general linear block codes. In *Cryptography and Coding 2001*, volume 2260 of *LNCS*, pages 1–8. Springer Verlag, 2001.
5. R.J. McEliece. A public key cryptosystem based on algebraic coding theory. *DSN progress report*, 42-44:114–116, 1978.
6. N. Sendrier. On the security of the McEliece public-key cryptosystem. In M. Blaum, P.G. Farrell, and H. van Tilborg, editors, *Proceedings of Workshop honoring Prof. Bob McEliece on his 60th birthday*, pages 141–163. Kluwer, 2002.
7. J. Stern. A method for finding codewords of small weight. *Coding Theory and Applications*, 388:106–133, 1989.