A New Algorithm for Finding Minimum-Weight Words in Large Linear Codes

Anne Canteaut *

INRIA Projet Codes
Domaine de Voluceau
78153 Le Chesnay Cedex, FRANCE
email: Anne.Canteaut@inria.fr

Abstract. An algorithm for finding small-weight words in large linear codes is developed and a precise analysis of its complexity is given. It is in particular able to decode random [512,256,57]-linear binary codes in 9 hours on a DEC alpha computer. We improve with it the previously best known attacks on some public-key cryptosystems and identification schemes based on error-correcting codes: for example we reduce the work factor involved in breaking McEliece's cryptosystem, since our algorithm requires 2^{64} elementary operations that is 128 times less than Lee-Brickell's attack.

1 Presentation of the Algorithm

Let C be an $[n, k]$-linear code over $GF(2)$. We present a probabilistic algorithm for finding a codeword of weight w, where w is small. This algorithm was elaborated with Florent Chabaud [2].

1.1 A Probabilistic Method

Since the probability that a randomly chosen codeword has weight w is very small, we need to only consider words verifying a given property so that their weight will be *a priori* small. All algorithms for finding short codewords use the same method [5, 6, 10]: they only take in account codewords which are particular linear combinations of a small number of rows of a systematic generator matrix. The heuristic proposed by Stern was shown to give the best results [3].

Let $N = \{1, \cdots, n\}$. For any subset I of N, $G = (V, W)_I$ denotes the decomposition of matrix G onto I, that means $V = (G_i)_{i \in I}$ and $W = (G_j)_{j \in N \setminus I}$, where G_i is the ith column of matrix G.

Definition 1. Let I be a k-element subset of N. I is an information window for code C iff $G = (Id_k, Z)_I$ is a systematic generator matrix for the code. The complementary set, $J = N \setminus I$, is called a redundancy window.

* Also with Ecole Nationale Supérieure de Techniques Avancées, laboratoire LEI, 32 boulevard Victor, 75015 Paris, FRANCE.

¿From now on we index the rows of Z with I since $G = (Id_k, Z)_I$ is a generator matrix for the code and we denote by Z^i the i-th row of matrix Z.

The idea suggested by Stern is to randomly choose at each iteration an information window I which is split into two parts I_1 and I_2 of same size, and a subset L of J of size ℓ. We only examine codewords c verifying the following property, where p and ℓ are fixed parameters for the algorithm:

$$\mathrm{wt}(c_{|I_1}) = \mathrm{wt}(c_{|I_2}) = p \text{ and } \mathrm{wt}(c_{|L}) = 0 \qquad (1)$$

until we find such a particular codeword whose restriction on $J \setminus L$ has weight $w - 2p$.

1.2 An Iterative Procedure

Stern's algorithm therefore explores a set of randomly selected information windows by performing at each iteration a Gaussian elimination on a generator matrix. In order to avoid this time-consuming procedure, we here propose to choose at each step the new information window by modifying only one element of the previous one. This method is analogous to the one used in the simplex method as suggested by Omura [9] and van Tilburg [12].

Definition 2. Two information windows I and I' are close iff:

$$\exists \lambda \in I,\ \exists \mu \in N \setminus I, \text{ such that } I' = (I \setminus \{\lambda\}) \cup \{\mu\}$$

Proposition 3. *Let I be an information window such that $G = (Id_k, Z)_I$ is a generator matrix for $\mathcal{C}$. Let be $\lambda \in I$, $\mu \in J$ and $I' = (I \setminus \{\lambda\}) \cup \{\mu\}$. I' is an information window iff $z_{\lambda,\mu} = 1$*

The redundant part Z of the systematic generator matrix can then be updated by a simple pivoting operation:

Proposition 4. *Let I and I' be two close information windows such that $I' = (I \setminus \{\lambda\}) \cup \{\mu\}$. Let $(Id_k, Z)_I$ and $(Id_k, Z')_{I'}$ be the corresponding systematic generator matrices. Then Z' is obtained from Z by:*

- $\forall j \in J'$, $z'_{\mu,j} = z_{\lambda,j}$
- $\forall i \in I' \setminus \{\mu\}$,
  - $\forall j \in J' \setminus \{\lambda\}$, $z'_{i,j} = z_{i,j} + z_{i,\mu} z_{\lambda,j}$
  - $z'_{i,\lambda} = z_{i,\mu}$

1.3 Description of the Iterative Algorithm

The use of this iterative procedure leads then to the following algorithm:

Initialization:

Randomly choose an information window I and apply a Gaussian elimination in order to obtain a systematic generator matrix $(Id_k, Z)_I$.

Until a codeword of weight w will be found:

1. randomly split I in two subsets I_1 and I_2 where $|I_1| = \lfloor k/2 \rfloor$ and $|I_2| = \lceil k/2 \rceil$. (the rows of Z are then split in two parts Z_1 and Z_2); select an ℓ-element subset L of J.
2. for each linear combination Λ_1 of p rows of matrix Z_1, compute $\Lambda_{1|L}$. For each linear combination Λ_2 of p rows of matrix Z_2, compute $\Lambda_{2|L}$.
3. if $\Lambda_{1|L} = \Lambda_{2|L}$, check whether $\mathrm{wt}((\Lambda_1 + \Lambda_2)_{|J \setminus L}) = w - 2p$.
4. randomly choose $\lambda \in I$ and $\mu \in J$. Replace I with $(I \setminus \{\lambda\}) \cup \{\mu\}$ by updating matrix Z according to the preceding proposition.

2 Theoretical Running Time

We here give a computable expression for the work factor of this algorithm, *i.e.* the average number of elementary operations it requires.

2.1 Average Number of Operations by Iteration

The average number of elementary operations performed at each iteration is:

$$\Omega_{p,\ell} = 2p\ell\binom{k/2}{p} + 2p(n-k-\ell)\frac{\binom{k/2}{p}^2}{2^\ell} + \frac{k(n-k)}{2} + K(p\binom{k/2}{p} + 2^\ell) \quad (2)$$

The first 3 terms correspond respectively to steps 2, 3 and 4 of the algorithm and the last one corresponds to the cost of the dynamic memory allocation (K is the size of a computer word).

2.2 Expected Number of Iterations

Since the successive information windows are not independent, the algorithm must be modeled by a discrete-time stochastic process.

Let c be the codeword of weight w to recover and $supp(c)$ its support. Let I be the information window and I_1, I_2 and L the other selections corresponding to the i-th iteration.

The i-th iteration can then be associated with the random variable X_i whose state space is $\mathcal{E} = \{0, \ldots, 2p-1\} \cup \{(2p)_S, (2p)_F\} \cup \{2p+1, \ldots, w\}$ where

$$\begin{array}{lll} X_i = u & \text{iff} & |I \cap supp(c)| = u, \;\; \forall u \in \{0, \ldots, 2p-1\} \cup \{2p+1, \ldots, w\} \\ X_i = (2p)_F & \text{iff} & |I \cap supp(c)| = 2p \text{ and } (|I_1 \cap supp(c)| \neq p \\ & & \text{or } |I_2 \cap supp(c)| \neq p \text{ or } |L \cap supp(c)| \neq 0) \\ X_i = (2p)_S & \text{iff} & |I_1 \cap supp(c)| = |I_2 \cap supp(c)| = p \text{ and } |L \cap supp(c)| = 0 \end{array}$$

The success space is then $\mathcal{S} = \{(2p)_S\}$ and the failure space is $\mathcal{F} = \{0, \ldots, (2p)_F, \ldots, w\}$.

Proposition 5. *The stochastic process $\{X_i\}_{i \in \mathrm{N}}$ associated with the algorithm is an homogeneous Markov chain.*

The corresponding transition matrix P can be easily computed [2]. Since this Markov chain is a transient chain, the following theorem can be applied for computing the expected number of iterations performed by the algorithm.

Theorem 6. *The expectation $\bar{N}$ of the number of iterations required until X_n reaches a success state is given by:*

$$\bar{N} = \sum_{i \in \mathcal{F}} \pi_0(i) \sum_{j \in \mathcal{F}} R_{i,j}$$

where R is the fundamental matrix i.e. $R = \sum_{m=0}^{\infty} Q^m = (Id - Q)^{-1}$

Assuming that the number of codewords of weight w is $\mathcal{A}_w$, the overall work factor required by the algorithm is then:

$$W_{p,l} = \frac{\Omega_{p,\ell} \bar{N}}{\mathcal{A}_w} \tag{3}$$

where $\bar{N}$ is given by theorem 6 and $\Omega_{p,\ell}$ by equation 2.

3 Applications

3.1 Decoding Random Linear Codes

We here study the work factor required for decoding an $[n, k, d]$-random linear binary code where d is given by the Gilbert-Varshamov's bound.

Using expression 3 we show that, for a fixed rate $r = k/n$, $\log_2(W)$ linearly depends on n only when parameters p and ℓ are optimized and that the work factor can be written in the form $W_{opt} = 2^{na(r)+b}$ (see figure 1).

Furthermore figure 2 shows that $a(r)$ is closed to the entropy function $H_2(r)$ multiplied by a fixed coefficient, where $H_2(x) = -x \log_2(x) - (1-x)\log_2(1-x)$.

Then we obtain the following result:

Proposition 7. *The theoretical work factor required for decoding a random $[n, k]$-binary code can be approximated by the following formula:*

$$W_{opt} = 2^{anH_2(k/n)+b} \textit{ where } a = 5.511\ 10^{-2} \textit{ and } b = 12$$

The same study was made concerning the work factor required for recovering a minimum-weight codeword in a random linear binary code ($w = d$) and we similarly obtain:

Proposition 8. *The theoretical work factor required for finding a minimum-weight word in a random $[n, k]$-binary code can be approximated by the following formula:*

$$W_{opt} = 2^{cnH_2(\frac{k}{n}+r_0)+d} \textit{ where } c = 0.12,\ d = 10 \textit{ and } r_0 = 3.125\ 10^{-2}$$

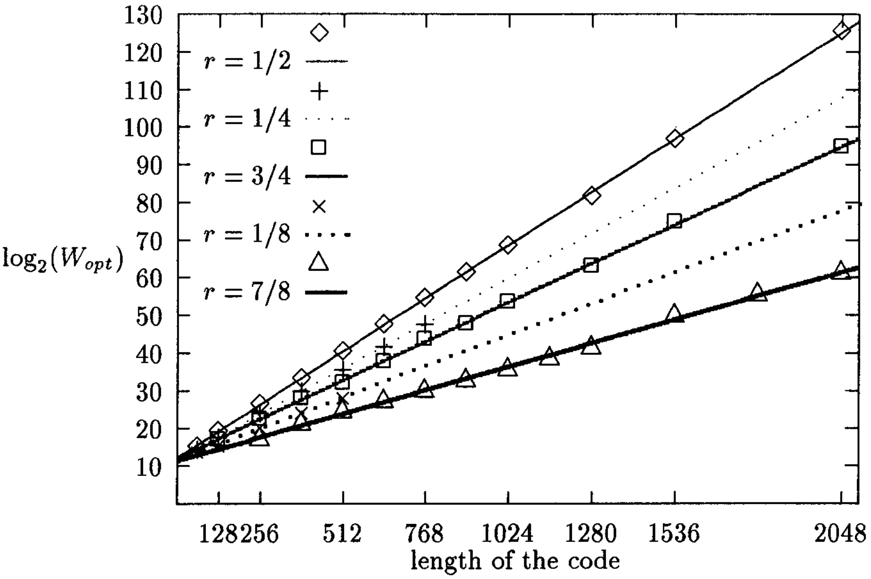

Fig. 1. Evolution of the theoretical work factor with optimized parameters for decoding random $[n, nr]$-binary codes

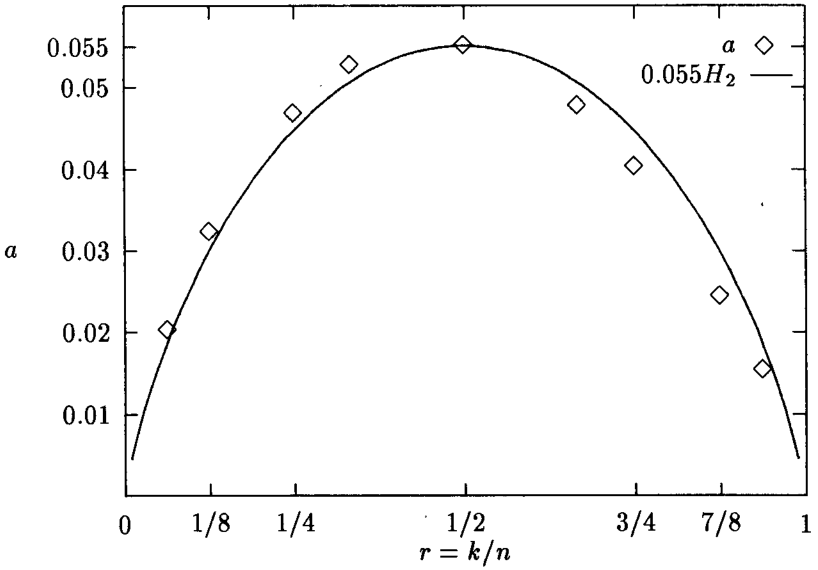

Fig. 2. Evolution of coefficient a vs. $r = k/n$

3.2 Cryptanalysis of Some Cryptosystems Based on Error-Correcting Codes

This algorithm can also be applied to attack McEliece's [7] and Niederreiter's [8] cryptosystems and some identification schemes based on syndrome decoding [4, 11] since they rely on the following NP-complete problem: given an $[n, k]$-linear binary code about nothing is known but one of its generator matrices G and given an n-bit vector y, we have to find a word of weight w in the coset of y, where w is equal to the error-correcting capability of the code for the cryptosystems, or slightly below its minimal distance for the identification schemes.

Considering the $[n, k+1]$-code whose generator matrix is $\left(\frac{G}{y}\right)$, an attack on these systems consists in searching a word of weight w in this new code.

We then obtain the following work factors.

cryptosystem	original McEliece	McEliece	Stern	McEliece
code	[1024,256]	[1024,614]	[512,256]	[512,260]
w	56	41	56	28
optimal parameters	$p=2$ $\ell=18$	$p=2$ $\ell=18$	$p=2$ $\ell=15$	$p=1$ $\ell=9$
work factor	$2^{64.2}$	$2^{66.0}$	$2^{69.8}$	$2^{40.9}$

Table 1. Work factors required for breaking some cryptosystems based on error-correcting codes

We otherwise notice that the parameters for McEliece's cryptosystem which maximize the work factor of this new attack are $n = 1024$, $k = 614$ and $w = 41$; the corresponding work factor is 2^{66}.

3.3 Experimental Results

We have made a great number of simulations for a small problem: decoding a $[256, 128, 29]$-random linear code whose minimal distance is obtained by Gilbert-Varshamov's bound. They confirm the validity of the previous theory concerning both optimized parameters and number of iterations (see table 2).

Decoding a $[256, 128, 29]$-random linear code requires 2 seconds on a DEC alpha station at 175 MHz. Thus decoding a [512,256,57]-one or breaking McEliece's cryptosystem with a $[512, 260]$ Goppa code requires around 9 hours on our computer.

Acknowledgments: I wish to thank Florent Chabaud with whom this algorithm was elaborated, and Hervé Chabanne who initiated this work.

parameters	theoretical work factor $\log_2(W)$	theoretical average iteration number	experimental average iteration number	deviation %	average CPU time (s)	corrected CPU time (s)
$p = 1, \ell = 5$	27.37	3961	4072	+2.80	2.76	2.68
$p = 1, \ell = 6$	26.80	4045	3985	-1.48	2.11	2.14
$p = 1, \ell = 7$	26.51	4139	4190	+1.23	2.07	2.04
$p = 1, \ell = 8$	26.56	4244	4338	+2.21	2.13	2.09
$p = 1, \ell = 9$	26.95	4362	4417	+1.26	2.90	2.87
$p = 2, \ell = 10$	29.80	432	433	+0.35	13.84	13.79
$p = 2, \ell = 11$	29.04	442	470	+6.51	13.00	12.21
$p = 2, \ell = 12$	28.51	454	446	-1.76	12.13	12.35
$p = 2, \ell = 13$	28.37	466	487	+4.51	17.70	16.91
$p = 2, \ell = 14$	28.68	480	508	+5.83	29.40	27.72

Table 2. Decoding random [256, 128, 29]-binary codes

References

1. A. Canteaut and H. Chabanne. A further improvement of the work factor in an attempt at breaking McEliece's cryptosystem. In P. Charpin, editor, *EUROCODE 94*, pages 163–167. INRIA, 1994.
2. A. Canteaut and F.Chabaud. Improvements of the attacks on cryptosystems based on error-correcting codes. Rapport interne du Département Mathématiques et Informatique LIENS-95-21, Ecole Normale Supérieure, Paris, July 1995.
3. F. Chabaud. On the security of some cryptosystems based on error-correcting codes. In A. De Santis, editor, *Advances in Cryptology – EUROCRYPT '94*, number 950 in Lecture Notes in Computer Science, pages 131–139. Springer-Verlag, 1995.
4. M. Girault. A (non-practical) three-pass identification protocol using coding theory. In J. Seberry and J. Pieprzyk, editors, *Advances in Cryptology – AUSCRYPT '90*, number 453 in Lecture Notes in Computer Science, pages 265–272. Springer-Verlag, 1991.
5. P.J. Lee and E.F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In C.G. Günther, editor, *Advances in Cryptology – EUROCRYPT '88*, number 330 in Lecture Notes in Computer Science, pages 275–280. Springer-Verlag, 1988.
6. J.S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Trans. Inform. Theory*, IT-34(5):1354–1359, September 1988.
7. R.J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report 42-44*, pages 114–116, 1978.
8. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.

9. J.K. Omura. Iterative decoding of linear codes by a modulo-2 linear program. *Discrete Math*, 3:193–208, 1972.
10. J. Stern. A method for finding codewords of small weight. In G. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, number 388 in Lecture Notes in Computer Science, pages 106–113. Springer-Verlag, 1989.
11. J. Stern. A new identification scheme based on syndrome decoding. In D.R. Stinson, editor, *Advances in Cryptology – CRYPTO '93*, number 773 in Lecture Notes in Computer Science. Springer-Verlag, 1994.
12. J. van Tilburg. On the McEliece public-key cryptosystem. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO '88*, number 403 in Lecture Notes in Computer Science, pages 119–131. Springer-Verlag, 1990.