An Observation on the Security of McEliece's Public-Key Cryptosystem

P. J. Lee and E. F. Brickell †

Bell Communications Research
Morristown, N. J., 07960 U. S. A.

Abstract

The best known cryptanalytic attack on McEliece's public-key cryptosystem based on algebraic coding theory is to repeatedly select k bits at random from an n-bit ciphertext vector, which is corrupted by at most t errors, in hope that none of the selected k bits are in error until the cryptanalyst recovers the correct message. The method of determining whether the recovered message is the correct one has not been throughly investigated. In this paper, we suggest a systematic method of checking, and describe a generalized version of the cryptanalytic attack which reduces the work factor significantly (factor of 2^{11} for the commonly used example of n=1024 Goppa code case). Some more improvements are also given. We also note that these cryptanalytic algorithms can be viewed as generalized probabilistic decoding algorithms for any linear error correcting codes.

I. Introduction

McEliece [1] introduced a public-key cryptosystem based on algebraic coding theory. Specifically, an (n,k) binary Goppa code [2] was chosen for this purpose since the error correction capability grows linearly with its dimension for a given code rate k/n. The correctable number of errors t for an (n,k) Goppa code with $n = 2^l$ is given by :

$$t \geq (n-k) / l. \tag{1}$$

---

† E. F. Brickell is now with Sandia National Laboratories, Albuquerque, NM 87183 U.S.A.

C.G. Guenther (Ed.): Advances in Cryptology - EUROCRYPT '88, LNCS 330, pp. 275-280, 1988.

The vectors, matrices and operations in the following discussion are all binary.

The next section describes McEliece's cryptosystem and the following section explains the best known cryptanalytic attack. After describing a systematic method of checking whether the recovered message is correct or not, we will suggest a generalization of the attack. Our analysis will show that the factor of improvement will be significant. Further improvements will also be discussed and conclusions and other discussions will follow.

II. Description of McEliece's Public-Key Cryptosystem

McEliece's system works as follows: The system user (receiver) secretly constructs a linear t-error correcting Goppa code with $k \times n$ code generator matrix $\boldsymbol{G}$, a $k \times k$ scrambler matrix $\boldsymbol{S}$ that has an inverse over $GF(2)$, and an $n \times n$ permutation matrix $\boldsymbol{P}$. Then he computes

$$\overline{\boldsymbol{G}} \equiv \boldsymbol{S}\ \boldsymbol{G}\ \boldsymbol{P} \tag{2}$$

which is also a linear code (but supposedly hard-to-decode) with the same rate and error correction capability as the original code generated by $\boldsymbol{G}$. He publishes $\overline{\boldsymbol{G}}$ as his public encryption key. The sender encrypts a k-bit message vector $\boldsymbol{m}$ into an n-bit ciphertext vector $\boldsymbol{c}$ as

$$\boldsymbol{c} = \boldsymbol{m}\ \overline{\boldsymbol{G}} + \boldsymbol{e} \tag{3}$$

where $\boldsymbol{e}$ is a random n-bit error vector of weight less than or equal to t. The receiver computes $\boldsymbol{c}\ \boldsymbol{P}^{-1} = (\boldsymbol{m}\ \boldsymbol{S})\ \boldsymbol{G} + \boldsymbol{e}\ \boldsymbol{P}^{-1}$ and uses the decoding algorithm for the original code with $\boldsymbol{G}$ to get rid of $\boldsymbol{e}\ \boldsymbol{P}^{-1}$. Finally to get $\boldsymbol{m}$ he descrambles $\boldsymbol{m}\ \boldsymbol{S}$ by multiplying $\boldsymbol{S}^{-1}$.

III. The Best Known Cryptanalytic Attack

There have been several methods proposed for attacking McEliece's system, [1], [3], [4], etc. Among them, the best attack with least complexity is to repeatedly select k bits at random from the n-bit ciphertext vector $\boldsymbol{c}$ to form $\boldsymbol{c}_k$ in hope that none of the selected k bits are in error. If there is no error in them, then $\boldsymbol{c}_k\ \overline{\boldsymbol{G}}_k^{-1}$ is equal to $\boldsymbol{m}$ where $\overline{\boldsymbol{G}}_k$ is the $k \times k$ matrix obtained by choosing k columns of $\overline{\boldsymbol{G}}$ according to the same selection of $\boldsymbol{c}_k$.

The work factor for the matrix inversion is $O(k^{\gamma})$ for some γ between 2 and 3. However, all of the known algorithms for $\gamma < 2.7$ have enormous constants that make them infeasible for matrices of a reasonable size. Perhaps the Winograd algorithm ([5], p. 481) with $\gamma \approx 2.8$ might be the best for these matrices of size between 500 and 1000. However, for the following analysis, we will use as in [4] the elementary algorithm with $\gamma = 3$ and small constant α.

The probability that there is no error in randomly selected k bits, among n bits with t errors, is $\binom{n-t}{k} / \binom{n}{k}$. Therefore, the total expected work factor for this attack is ; [3],[4]

$$W = \alpha\, k^3 \binom{n}{k} / \binom{n-t}{k} . \tag{4}$$

Originally, in [1], the values of l=10 and t=50 (or n=1024, k=524) were suggested, which result in the work factor of approximately $2^{80.7}$ (with α = 1). More recently, in [4], the optimum value of t that maximizes the work factor for n=1024 was shown to be 37 (or equivalently, k=654) providing $W \approx 2^{84.1}$.

IV. Systematic Method of Checking $c_k\ G_k^{-1}$

Notice that the work factors for checking whether the obtained $c_k\ G_k^{-1}$ is really m was not discussed in [1] and [4]. While, [3] just suggested that the validity of $c_k\ G_k^{-1}$ may be determined by the redundancy in m, which might not be practical.

Here, we provide a systematic and practical method of checking whether the obtained $c_k\ G_k^{-1}$ is m or not. Since G is also a code generator matrix having minimum distance larger than $2t$, if $c_k\ G_k^{-1}$ is not the true m, then $m\ G + c_k\ G_k^{-1}\ G$ must have weight at least $2t$. Hence if $c + c_k\ G_k^{-1}\ G$ has weight less than or equal to t, then the cryptanalyst can claim that $c_k\ G_k^{-1} = m$.

V. Generalization of the Above Attack

The above cryptanalysis can be generalized by allowing a very small number of errors in the selected c_k. The following describes the algorithm :

Algorithm j :

Step 1) Randomly choose k bits from an n-bit ciphertext c (denoted as c_k). Let $\overline{G}_k$ be the $k \times k$ matrix obtained by choosing the corresponding columns of $\overline{G}$. Calculate $\overline{G}_k^{-1}\ \overline{G}$ and $c + c_k\ (\overline{G}_k^{-1}\ \overline{G})$.

Step 2) Choose an unused k-bit error pattern e_k with less than or equal to j ones. If $(c + c_k\ \overline{G}_k^{-1}\ \overline{G}) + e_k\ (\overline{G}_k^{-1}\ \overline{G})$ has weight t or less, then stop $(m = c_k\ \overline{G}_k^{-1})$.

Step 3) If there are no more unused k-bit error patterns with less than or equal to j ones, go to Step (1). Otherwise, go to Step (2).

Notice that Algorithm 0 is the attack discussed in Section III including our systematic checking of $c_k\ \overline{G}_k^{-1}$.

Let Q_i be the probability that there are exactly i errors among the randomly chosen k-bit vector c_k. It can be shown that

$$Q_i = \binom{t}{i} \binom{n-t}{k-i} / \binom{n}{k}. \tag{5}$$

Hence, the probability that the algorithm completes successfully is $\sum_{i=0}^{j} Q_i$. Therefore, the expected number of executions of Step 1), T_j, is

$$T_j = 1 / \sum_{i=0}^{j} Q_i\ . \tag{6}$$

Let N_j be the number of k-bit error patterns with less than or equal to j ones. Then,

$$N_j = \sum_{i=0}^{j} \binom{k}{i}. \tag{7}$$

Hence, N_j is the number of executions of Step 2) for a given choice of c_k with more than j errors in it.

The work factor involved in Step 1) is approximately $\alpha\, k^3$ with small α when $k > n/2$. The work factor involved in Step 2) is approximately $\beta\, k$ with small β since we can just update the vector $e_k\ (\overline{G}_k^{-1}\ \overline{G})$ for each choice of e_k which differs in at most two positions from the previous choice of e_k. Therefore, the average overall work factor for Algorithm j, W_j, is

$$W_j = T_j\ (\alpha\, k^3 + N_j\ \beta\, k)\,. \tag{8}$$

Notice that $W = W_0$. Also notice that for any reasonable value of α and β, W_j decreases and then increases as j increases. With $\alpha = \beta$, we can show that the optimum j which minimizes the work factor is 2 for all values of useful code parameters. With $\alpha = \beta = 1$, the minimum work factor $W_2 \approx 2^{73.4}$ for the case of $n = 1024$ and $t = 37$, which is a factor of 2^{11} reduction as compared to W_0. For $n = 1024$ case, the value of t which maximizes W_2 is 38 (k=644), for which W_2 is also approximately $2^{73.4}$.

VI. Further Improvements

Instead of calculating the vector $(c + c_k \bar{G}_k^{-1} \bar{G}) + e_k (\bar{G}_k^{-1} \bar{G})$ ($\equiv \bar{e}$) first and then checking whether $\bar{e}$ has weight t or less in Step 2), one can calculate one bit by one bit of the vector $\bar{e}$ and check the accumulated weight until it exceeds t. When we assume that the vector $\bar{e}$ has average weight $n/2$ for incorrect cases, we can expect that the number of bits to be tested in this improved Step 2)' is $2t$ in average. Hence, the work factor for Step 2)' is less than that of Step 2) by a factor of $k/2t$ in average. For the previous example, this is a factor of 10 improvement.

For each Step 1) the new c_k is selected randomly. However, one can just randomly update only one bit of c_k each time. The work factor in this Step 1)' is then reduced to $\alpha' k^2$ for updating $(\bar{G}_k^{-1} \bar{G})$. In this case, however, we could not find the expected number of excutions of Step 1)' before success, T_j'. If one assumes that T_j' is the same as T_j, it can be shown that the optimum j which minimizes W_j' is 1 when $\alpha' = \beta$ (with Step 2)). And for the previous example of $l = 10$, the value of t that maximize the W_1' is also 38 resulting $W_1' = 2^{69.6}$. And, together with Step 2)', we can improve another factor of 10.

VII. Conclusions and Discussion

In conclusion, we have described a systematic method of checking the validity of the recovered cleartext. And we suggested an improved crytanalytic attack which is a factor of 2^{11} more efficient than the previously known best attack. We also suggested some more improvements over the new attack.

In [6], it was shown that the syndrome decoding of general linear algebraic code is an *NP*-complete problem and the running time for the syndrome decoding is an exponential function of its input dimension k, and it is claimed that the discovery of an algorithm which runs significantly faster than this would be an important achievement. The cryptanalytic attack of [1] described in Section III and our generalizations are general probabilistic decoding algorithms for any general linear error correction code which can run more efficiently (although still in exponential time) than the syndrome decoding of a general code when the number of errors in a code word seldom exceeds its error correcting capability.

References

[1] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," CA, May 1978.

[2] E. R. Berlekamp, "Goppa codes," *IEEE Trans. Info. Theory*, Vol. IT-19, pp. 590-592, Sept. 1973.

[3] T.R.N. Rao and K.-H. Nam, "Private-key algebraic-coded cryptosystems," *Proc. Crypto '86*, pp. 35-48, Aug. 1986

[4] C. M. Adams and H. Meijer, "Security-related comments regarding McEliece's public-key cryptosystem," to appear in *Proc. Crypto '87*, Aug. 1987

[5] D. E. Knuth, *The Art of Computer Programming, Vol. 2. Seminumerical Algorithms*, Addison-Wesley, 1981

[6] E. R. Berlekamp, *et al.*, "On the inherent intactability of certain coding problems," *IEEE Trans. Info. Theory*, Vol. IT-22, pp. 644 - 654, May 1978.