B. Kaliski (Ed.), *Advances in Cryptology -- Proceedings of Crypto '97*,
Lecture Notes in Computer Science, Vol. 1294, pages 213-220,
Springer Verlag, 1997

Failure of the McEliece Public-Key Cryptosystem Under Message-Resend and Related-Message Attack

Thomas A. Berson

Anagram Laboratories
P.O. Box 791
Palo Alto, CA 94301 USA
berson@anagram.com

Abstract: The McEliece public-key cryptosystem fails to protect any message which is sent to a recipient more than once using different random error vectors. In general, it fails to protect any messages sent to a recipient which have a known linear relation to one another. Under these conditions, which are easily detectable, the cryptosystem is subject to a devastating attack which reveals plaintext with a work factor which is 10^{15} times better than the best general attack.

Keywords: McEliece, public-key cryptosystem, randomization, error-correcting codes, error vectors, message-resend attack, related-message attack, protocol failure, cryptanalysis.

1 Introduction

The McEliece public-key cryptosystem was proposed nearly 20 years ago [14]. The system is simple to explain and is very fast in execution. It is based on an **NP**-hard problem in coding theory, and features the ability of a hidden error-correcting code to recover plaintext from ciphertexts which the sender intentionally garbles with random errors. Although it has received much attention from the cryptologic community, the system remains unbroken to this day.

Despite these advantages, the McEliece public-key cryptosystem it is not widely used. Perhaps this is because it has a large public key and a low information rate. But changes in technology and economics, for example the plummeting cost of storage, keep it on the list of candidates for some applications.

In this paper we analyze and exploit the failure of the McEliece public-key cryptosystem to protect plaintext when any message is sent to a recipient more than once using different random error vectors. Our *message-resend* attack succeeds in βk^3 time, where β is a small constant, and k is the message size of the underlying code. We then generalize our attack to a *related-message* attack, which recovers any messages sent to a recipient when a linear relation between the messages is known, again in βk^3 time.

2 The McEliece Public-Key Cryptosystem

Without loss of generality we will describe the McEliece public-key cryptosystem system using the code and parameter sizes proposed originally by McEliece.

The private key consists of three matrices:

- a generator matrix for a ($n = 50$, $k = 524$, $t = 50$) Goppa code $G \in F_2^{524\times 1024}$ (Goppa codes are a large class of error-correcting codes which have efficient decoding algorithms);
- an invertible scrambler matrix $S \in F_2^{524\times 524}$, and;
- a permutation $P \in F_2^{1024\times 1024}$.

The public key is the matrix product *SGP*. Note that *S* and *P* disguise *G* as a general linear code.

Now suppose a message $m \in F_2^{524}$ is to be sent. The parameters of the Goppa code (an irreducible polynomial $g(x) \in F_2[X]$ of degree 50 and an ordering of $F_{2^{10}}$) allow for the fast error correction of up to 50 errors. So a random error vector $e \in F_2^{1024}$ is chosen where the Hamming weight $wt(e) = 50$, and the cryptogram

$$c = mSGP + e$$

is sent.

The intended recipient then computes

$$cP^{-1} = mSG + eP^{-1}.$$

Since *P* is a permutation, $wt\left(eP^{-1}\right) = 50$. So decoding the Goppa code recovers *mS*, from which, finally, the intended recipient recovers $m = (mS)S^{-1}$.

Remarks

A great many workers, starting with Adams and Meijer [1,2], Hin [9], and Jorissen [10], have explored the relationship between the parameters of the underlying code, the security of the cryptosystem, and the data rate. For a description of this line of research see van Tilburg [17]. Optimizations have been suggested where n = 1024, k ranges from 524 to 654, and t ranges from 37 to 50. Our attack is not blunted by such adjustments.

Other workers have explored replacing the Goppa code with other types of error-correcting code. For example, Gabidulin et al. [5] tried using maximum-rank-distance codes. These schemes were shown to be insecure by Gibson [6,7]. In any event, such code replacements would not prevent our attack, which does not depend on the structure of the code.

3 Cryptanalytic Background

McEliece stated that the most promising line of attack on his public-key cryptosystem consists of decoding an arbitrary linear code containing correctable errors. Therefore, the security of the cryptosystem seems to be based on solving the corresponding the BHDD[1] problem.

The obvious [14,1] attack is this: if a cryptanalyst could guess 524 coordinates of *c* that are not garbled by *e*, then the restriction to those 524 columns of the cryptogram and the public key

$$\bar{c} = m\overline{SGP}$$

relates *m* to $\bar{c}$ by a known $\overline{SGP} \in F_2^{524\times 524}$. If $\overline{SGP}$ is invertible, then *m* can be recovered.

Notice that this is a per-message attack; the secret key of the system remains unknown to the cryptanalyst.

What is the work factor for this attack? The cryptanalyst must correctly guess 524 ungarbled columns out of the possible 974 = 1024-50. So we can calculate that it will require

$$\frac{\binom{1024}{524}}{\binom{974}{524}} \approx 1.37 \times 10^{16} \text{ guesses to succeed.}$$

So the work factor is

$$w = \alpha \cdot 1.37 \times 10^{16},$$

where α is the cost of inverting a 524-square matrix, roughly 524^3.

Notice that the relatively low-weight error vector is crucial to the success of the Goppa decoding algorithm, and that it also impacts the work necessary for the cryptanalyst.

[1] BHDD (Binary Hamming Distance Decoding) is the name given to the problem of decoding an arbitrary binary word to the nearest codeword in an arbitrary linear code under the restriction that the “arbitrary” binary word be at distance at most $(d-1)/2$ from a codeword. Berlekamp, McEliece and van Tilborg [4] showed that BHDD is **NP**-hard.

Remarks

The attack described above can be, and has been, improved slightly by taking partial information into account. See Lee and Brickell [12], Li, Deng and Wang [13], and van Tilburg [16].

There was some excitement and confusion about the cryptanalysis of the McEliece public-key cryptosystem a few years ago. Korzhik and Turkin, announced that they had broken the cryptosystem. They gave a “demonstration” of their “attack” at Eurocrypt ’91 [11]. However, the demonstration was only a toy: in place of the Goppa code it used a BCH code of dimension 36 in $F_2^{63} = GF(2)^{63}$, with minimum distance 11, and an error vector of weight 5. Even with this simplification, their attack achieved only a five-fold speedup over exhaustion. The details have never appeared. More generally, Korzhik and Turkin claimed to have found a polynomial time algorithm for BHDD, which is known to be **NP**-hard. But the published description and analysis of their algorithm are not precise, and its correct functioning within the claimed time bound has never been confirmed. In summary, their attack on the McEliece public-key cryptosystem is not believed to be effective.

4 Failure Under Message-Resend Conditions

Suppose now that, through some accident, or as a result of action in the part of the cryptanalyst, both

$$c_1 = mSGP + e_1$$

and $$e_1 \neq e_2$$

$$c_2 = mSGP + e_2$$

are sent. We call this a *message-resend* condition. In this case it is easy for the cryptanalyst to recover *m* from the system of c_i. (We will examine only the case where the number of different cryptograms of the same message, which we call the *resend depth*, is 2. The attack is even easier at greater resend depths.)

Notice that $c_1 + c_2 = e_1 + e_2$ (mod 2).

The cryptanalyst can easily detect a message-resend condition by observing the Hamming weight of the sum of any two cryptograms. When the underlying messages are different, the expected weight of the sum is about 512. When the underlying messages are identical, the weight of the sum cannot exceed 100. Heiman [8] showed that a message-resend condition can be detected; we will show how to exploit it.

4.1 Method of Attack

We will compute two sets from $(c_1 + c_2)$. The set L_0 will be the locations where $(c_1 + c_2)$ contains zeroes. The set L_1 will be the locations where $(c_1 + c_2)$ contains ones.

Let

$$L_0 = \{l \in \{1,2,\cdots,1024\} : c(l)_1 + c_2(l) = e_1(l) + e_2(l) = 0\}$$

and

$$L_1 = \{l \in \{1,2,\cdots,1024\} : c_1(l) + c_2(l) = e_1(l) + e_2(l) = 1\}.$$

We aim to take advantage of the fact (and to quantify the claim) that

- $l \in L_0 \Rightarrow$ most probably neither $c_1(l)$ nor $c_2(l)$ is garbled by an error vector, while
- $l \in L_1 \Rightarrow$ certainly precisely one of $c_1(l)$ or $c_2(l)$ is garbled by an error vector.

Every $l \in L_0$ means that either $e_1(l) = 0 = e_2(l)$ or $e_1(l) = 1 = e_2(l)$. Assuming the error vectors e_1 and e_2 are chosen independently, then for any *l*

$$\Pr(e_1(l) = 1 = e_2(l)) = \left(\frac{50}{1024}\right)^2 \approx 0.0024.$$

In other words, most $l \in L_0$ signify $e_1(l) = 0 = e_2(l)$. Thus the cryptanalyst should try to guess 524 ungarbled columns from those indexed by L_0.

How good is this strategy? Let p_i be the probability that precisely *i* coordinates are simultaneously garbled by e_1 and e_2. Then

$$p_i = \Pr(|\{l : e_1(l) = 1\} \cap \{l : e_2(l) = 1\}| = i) = \frac{\binom{50}{i}\binom{974}{50-i}}{\binom{1024}{50}}$$

since, say, e_2 must choose *i* error locations from those 50 of e_1 and the remaining 50-*i* from those ungarbled by e_1, this out of a total of $\binom{1024}{50}$ possible error vectors.

Therefore the expected cardinality of L_1 is

$$E(|L_1|) = \sum_{i=0}^{50} (100 - 2i) p_i \approx 95.1$$

since every l for which $e_1(l) = 1 = e_2(l)$ reduces $|L_1|$ by two.

For example, suppose $|L_1| = 94$. Then $|L_0| = 930$, of which only 3 are garbled. We see that the probability of guessing 524 ungarbled columns from those indexed by L_0 is

$$\frac{\binom{927}{524}}{\binom{930}{524}} \approx 0.0828$$

so the cryptanalyst expects to succeed in this case with only 12 guesses, at a cost of 12α.

When $|L_1| = 96$ only about 5 guesses are required!

These results are a factor of 10^{15} better than the exhaustive attack analyzed in Section 3.

Note that this attack does not recover the private key. We do not claim to have broken the McEliece public-key cryptosystem. But we have shown how a cryptanalyst may recover the plaintext of a resent message with very little work.

5 Failure Under Related-Message Conditions

We will now generalize the message-resend attack. Suppose that there are two cryptograms

$$c_1 = m_1SGP + e_1$$

and $\qquad m_1 \neq m_2,\ e_1 \neq e_2$

$$c_2 = m_2SGP + e_2$$

and that the cryptanalyst knows a linear relation, for example $m_1 + m_2$, between the messages. We call this a *related-message* condition. In this case the cryptanalyst may recover the m_i from the set of c_i by doing one encoding and by then following the attack method of Section 4.1. Here are the details.

Combining the two cryptograms we get

$$c_1 + c_2 = m_1SGP + m_2SGP + e_1 + e_2.$$

Notice that $m_1SPG + m_2SGP = (m_1 + m_2)SGP$, a value the cryptanalyst may calculate in a related-message condition from the known relationship and the public key.

The cryptanalyst solves

$$c_1 + c_2 + (m_1 + m_2)SGP = e_1 + e_2$$

and proceeds with the attack as in Section 4.1, using $(c_1 + c_2 + (m_1 + m_2)SGP)$ in place of $(c_1 + c_2)$.

Remark

The message-resend attack is that special case of the related-message attack where $m_1 + m_2 = 0$.

6 Conclusions

The McEliece public-key cryptosystem fails to protect any message which is sent to a recipient more than once using different random error vectors.

The McEliece public-key cryptosystem fails to protect messages sent to a recipient which are have a known linear relation to one another.

Our attack is a general attack on the class of public-key cryptosystems which use an error-correcting code and the introduction of random errors by the sender.

Our attack under these conditions is a factor of 10^{15} better than the best attack under general conditions.

Users of the McEliece public-key cryptosystem, and of cryptosystems with similar structure, should guard against sending related messages. One countermeasure which comes to mind is to introduce an element of local randomness into any message before it is encrypted. But note that the obvious $c = (m \| r)SGP$ falls quickly to a synthesized related-message attack. A scheme is required which spreads randomness through the plaintext in some complicated fashion. Bellare and Rogaway's OAEP [3] *et seq.* are instructive. Of course, any such scheme extracts a penalty in data rate.

Cryptosystems which are based on the use of linear codes but without per-message error vectors, for example Neiderreiter [15], are not directly threatened by our attack. However, prudence dictates that all such systems now be reexamined for vulnerability to message-resend or related-message attack.

Acknowledgments

This work was begun while the author was a visitor at the Isaac Newton Institute for Mathematical Sciences in the University of Cambridge.

The author is grateful to Susan Langford, to Kevin McCurley, and to Matt Robshaw for their clarifying discussions.

References

1. C. ADAMS AND H. MEIJER, “Security-related comments regarding McEliece’s public-key cryptosystem”, *Advances in Cryptology—Crypto ’87 (LNCS 293*), 224-228, 1988.

2. C. ADAMS AND H. MEIJER, “Security-related comments regarding McEliece’s public-key cryptosystem”, *IEEE Transactions on Information Theory*, 35 (1989), 454-455.

3. M. BELLARE AND P. ROGAWAY, “Optimal asymmetric encryption”, *Advances in Cryptology – EUROCRYPT ’94 (LNCS 950)*, 232-249, 1994.

4. E.R. BERLEKAMP, R.J. MCELIECE, AND H.C.A. VAN TILBORG, “On the inherent intractability of certain coding problems”, *IEEE Transactions on Information Theory*, 24 (1978), 384-386.

5. E.M. GABIDULIN, A.V. PARAMONOV, AND O.V. TRETJAKOV, “Ideals over a non-commutative ring and their application in cryptology”, *Advances in Cryptology—EUROCRYPT ’91 (LNCS 547), 482-489*, 1991.

6. J.K. GIBSON, “Severely denting the Gabidulin version of the McEliece public key cryptosystem”, *Designs, Codes and Cryptography*, 6 (1995), 37-45.

7. J.K. GIBSON, “The security of the Gabidulin public key cryptosystem”, *Advances in Cryptology—EUROCRYPT ’96 (LNCS 1070)*, 212-223, 1996.

8. R. HEIMAN, “On the security of cryptosystems based on linear error-correcting codes”, M.Sc. Thesis, Feinburg Graduate School, Weitzmann Institute of Science, Rehovot, August, 1987.

9. P.J.M. HIN, “Channel-error-correcting privacy cryptosystems”, M.Sc. Thesis, Delft University of Technology, Delft, 1986.

10. F. JORISSEN, “A security evaluation of the public-key cipher system proposed by R.J. McEliece, used as a combined scheme”, Technical report, Katholieke Universiteit Leuven, Dept. Elektrotechniek, January, 1986.

11. V.I. KORZHIK AND A.I. TURKIN, “Cryptanalysis of McEliece’s public-key cryptosystem”, *Advances in Cryptology—EUROCRYPT ’91 (LNCS 547)*, 68-70, 1991.

12. P.J. LEE AND E.F. BRICKELL, “An observation on the security of McEliece’s public-key cryptosystem”, *Advances in Cryptology—EUROCRYPT ’88 (LNCS 330)*, 275-280, 1988.

13. Y.X. LI, R.H. DENG, AND X.M. WANG, “On the equivalence of McEliece’s and Neiderreiter’s public-key cryptosystem”, *IEEE Transactions on Information Theory*, 40 (1994), 271-273.

14. R.J. MCELIECE, “A public-key cryptosystem based on algebraic coding theory”, DSN Progress Report 42-44, Jet Propulsion Laboratory, Pasadena, 1978.

15. H. NEIDERREITER, “Knapsack-type cryptosystems and algebraic coding theory”, *Problems of Control and Information Theory*, 15 (1986), 159-166.

16. J. VAN TILBURG, “On the McEliece public-key cryptosystem”, *Advances in Cryptology—Crypto ’88 (LNCS 403)*, 119-131, 1990.

17. J. van Tilburg, “Security analysis of a class of cryptosystems based on linear error-correcting codes”, Ph.D. Thesis, Technische Universiteit Eindhoven, Eindhoven, November, 1994.