The McEliece Cryptosystem

Introduction

This public key cryptosystem, introduced by McEliece in 1978, is similar to the Merkle-Hellman Knapsack cryptosystem in that it takes an easy case of an NP-problem and disguises it to look like the hard instance of the problem. In this cryptosystem, the problem that is used is drawn from the theory of error-correcting codes.

The Problem

Syndrome decoding of linear codes (when considered as a decision problem) is an NP-complete problem if the number of errors is not bounded. However, there are classes of linear codes which have very fast decoding algorithms. The basic idea of the McEliece system is to take one of these linear codes and disguise it so that Oscar, when trying to decrypt a message, is forced to use syndrome decoding, while Bob, who set up the system, can remove the disguise and use the fast decoding algorithm. McEliece suggested using **Goppa Codes**, which are linear codes with a fast decoding algorithm, in the system, but any linear code with a good decoding algorithm can be used.

The Cryptosystem

Let C be an [n,k]-linear code with a fast decoding algorithm that can correct t or fewer errors. Let G be a generator matrix for C. To create the disguise, let S be a $k \times k$ invertible matrix (the ***scrambler***) and let P be an $n \times n$ permutation matrix (i.e., having a single 1 in each row and column and 0's everywhere else). The matrix,

$$G' = SGP$$

is made public while S, G and P are kept secret by Bob. For Alice to send a message to Bob, she blocks her message into binary vectors of length k. If x is one such block, she randomly constructs a binary n-vector of weight t (that is, she randomly places t 1's in a zero vector of length n), call it e and then sends to Bob the vector

$$y = xG' + e.$$

The Cryptosystem

Oscar, upon intercepting this message, would have to find the nearest codeword to y of the code generated by G'. This would involve calculating the syndrome of y and comparing it to the syndromes of all the error vectors of weight t. As there are $\binom{n}{t}$ of these error vectors, good choices of n and t will make this computation infeasible.

Bob, on the other hand, would calculate

$$yP^{-1} = (xG' + e)P^{-1} = xSG + eP^{-1} = xSG + e'$$

where e' is a vector of weight t (since P^{-1} is also a permutation matrix). Bob now applies the fast decoding algorithm to strip off the error vector e' and get the code word (xS)G.

The Cryptosystem

The vector xS can now be obtained by multiplying by G^{-1} on the right (however, if Bob had been smart, he would have written G in standard form $[I_k\ A]$, and then xS would just be the first k positions of xSG and this multiplication would not be needed). Finally, Bob gets x by multiplying xS on the right by S^{-1}.

For McEleice's Goppa Code example, n = 1024 and t = 50 which gives Oscar more than 10^{80} syndromes to calculate.

An Example

For an example we shall use the (7,4) Hamming code which corrects all single errors. A generator matrix for this code is given by (note the clever choice):

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

and Bob chooses the scrambler matrix

$$S = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

An Example

and the permutation matrix

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Bob makes public the generator matrix

$$G' = S\,G\,P = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

An Example

If Alice wishes to send the message x = (1 1 0 1) to Bob, she first constructs a weight 1 error vector, say e = (0 0 0 0 1 0 0) and computes

$$\begin{aligned} y &= xG' + e \\ &= (0\ 1\ 1\ 0\ 0\ 1\ 0) + (0\ 0\ 0\ 0\ 1\ 0\ 0) \\ &= (0\ 1\ 1\ 0\ 1\ 1\ 0) \end{aligned}$$

which she then sends to Bob.

Upon receiving y, Bob first computes $y' = yP^{-1}$, where

$$P^{-1} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

obtaining y' = (1 0 0 0 1 1 1).

An Example

Now Bob decodes y' by the fast decoding algorithm (Hamming decoding in this example). The error occurs in position 7 (details omitted). Bob now has the code word y" = (1 0 0 0 1 1 0). Because of the clever choice for G, Bob knows that xS = (1 0 0 0), and he can now obtain x by multiplying by the matrix

$$S^{-1} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

obtaining $x = (1\ 0\ 0\ 0)S^{-1} = (1\ 1\ 0\ 1)$.

Drawbacks

There are three major concerns with the McEliece cryptosystem.

1. The size of the public key (G') is quite large. Using the Goppa code with parameters suggested by McEliece, the public key would consist of 2^{19} bits. This will certainly cause implementation problems.

2. The encrypted message is much longer than the plaintext message. This increase of the bandwidth makes the system more prone to transmission errors.

3. The cryptosystem can not be used for authentication or signature schemes because the encryption algorithm is not one-to-one and the total algorithm is truly asymmetric (encryption and decryption do not commute).

Security

The McEliece cryptosystem is considered to be fairly secure. However, in 1986 Rao and Nam proposed a variant of the system using only one matrix to disguise the problem and the following year Struik and Tilburg showed how to break the Rao-Nam system.

Goppa Codes

Although we will not describe the Goppa Codes here, we will present a few facts about them.

For each irreducible polynomial of degree **t** over $GF(2^m)$ there corresponds a binary, irreducible Goppa Code of length $n = 2m$, dimension $k \geq n\text{-}tm$ and minimum distance $d \geq 2t+1$. A fast decoding algorithm, with running time nt, exists. Goppa Codes are easily set up once the irreducible polynomial is found. This is not difficult since there are about $2^{mt}/t$ irreducible polynomials of degree t over $GF(2^m)$. So, a random polynomial of degree t over $GF(2^m)$ will be irreducible with probability 1/t. Since there is a fast algorithm for testing irreduciblity, one can find one quickly by simply guessing and testing.