A successful attack on the McEliece cryptosystem with original parameters

Christiane Peters
joint work with Dan Bernstein and Tanja Lange

Technische Universiteit Eindhoven

S^3CM 2009

July 17, 2009

1. Motivation

2. Review of the McEliece cryptosystem

3. Stern's attack

4. Attack optimization and comparison

Post-quantum cryptography

- Quantum computers will break the most popular public-key cryptosystems (PKCs).

- Post-quantum cryptography—a very recent field of cryptography—deals with cryptosystems that run on conventional computers and are secure against attacks by quantum computers.

- The McEliece cryptosystem—introduced by R.J. McEliece in 1978—is one of the public-key systems without known vulnerabilities to attacks by quantum computers.

Linear codes

A binary $[n,k]$ code is a binary linear code of length n and dimension k, i.e., a k-dimensional subspace of $\mathbf{F}_2^n$.

A generator matrix of an $[n,k]$ code C is a $k \times n$ matrix G such that $C = \{\mathbf{x}\,G : \mathbf{x} \in \mathbf{F}_2^k\}$.

The matrix G corresponds to a map $\mathbf{F}_2^k \to \mathbf{F}_2^n$ sending a message of length k to an n-bit string.

A parity-check matrix of an $[n,k]$ code C is an $(n-k) \times n$ matrix H such that $C = \{\mathbf{c} \in \mathbf{F}_2^n : H\,\mathbf{c}^T = 0\}$.

Decoding problem

We only consider binary codes, i.e., codes over $\mathbf{F}_2$. In particular, we consider codes with no obvious structure.

Classical decoding problem: find the closest codeword $\mathbf{x} \in C$ to a given $\mathbf{y} \in \mathbf{F}_2^n$, assuming that there is a unique closest codeword.

Berlekamp, McEliece, van Tilborg (1978) showed that the general decoding problem for linear codes is NP-complete.

McEliece PKC from an attacker's point of view

Given a $k \times n$ generator matrix G of a public code, and an error weight w.

To encrypt a message $\mathbf{m} \in \mathbf{F}_2^k$, the sender computes $\mathbf{m}G$, adds a random weight-w error vector $\mathbf{e}$, and sends $\mathbf{y} = \mathbf{m}G + \mathbf{e}$.

McEliece proposed choosing a random degree-t classical binary Goppa codes as secret key; G generates a permutation-equivalent code.

The standard parameter choices are $k = n - t\lceil \lg n \rceil$ and $w = t$, typically with n a power of 2.

McEliece's original suggestion: $n = 1024$, $k = 524$, and $w = 50$.

Attacking the McEliece cryptosystem

Not knowing the secret code and its decoding algorithm the attacker is faced with the problem of decoding $\mathbf{y}$ in a random-looking code.

Two possible attacks:

- Find out the secret code.
- Or decode $\mathbf{y}$ without knowing an efficient decoding algorithm for the public code given by G.

Attacks on the McEliece PKC

- Most effective attack against the McEliece PKC is information-set decoding; used for decoding a given number of errors in $\mathbf{y}$ without knowledge of a decoding algorithm.
- Many variants: McEliece (1978), Leon (1988), Lee and Brickell (1988), Stern (1989), van Tilburg (1990), Canteaut and Chabanne (1994), Canteaut and Chabaud (1998), and Canteaut and Sendrier (1998).
- Our complexity analysis showed that Stern's original attack beats Canteaut et al. when aiming for 128-bit security
- Our attack is most easily understood as a variant of Stern's attack.
- Our attack is faster by a factor of more than 150 than previous attacks; now within reach of a moderate cluster of computers.

1. Motivation

2. Review of the McEliece cryptosystem

3. Stern's attack

4. Attack optimization and comparison

Reduce decoding to minimum-weight-word finding

McEliece ciphertext $\mathbf{y} \in \mathbf{F}_2^n$ has distance t from a unique closest codeword $\mathbf{c} = \mathbf{m}G$ in a code C which has minimum distance at least $2t + 1$.

Find $\mathbf{e}$ of weight t such that $\mathbf{c} = \mathbf{y} - \mathbf{e}$:

- append $\mathbf{y}$ to the list of generators
- and form a generator matrix for $C + \{0, \mathbf{y}\}$.

Then

$$\mathbf{e} = (\mathbf{m}, 1) \left(\frac{G}{\mathbf{m}G + \mathbf{e}} \right)$$

is a codeword in $C + \{0, \mathbf{y}\}$; and it is the only weight-t word.

Bottleneck in all of these attacks is finding the weight-t codeword in $C + \{0, \mathbf{y}\}$ which has slightly larger dimension, namely $k + 1$.

Stern's attack

- Given $w \geq 0$ and an $(n-k) \times n$ parity check matrix H for a binary $[n, k]$ code C. Find $\mathbf{c} \in C$ of weight w.

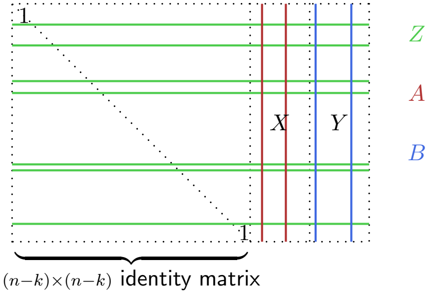
- Construct $\mathbf{c}$ by looking for exactly w columns of H which add up to 0.

- Stern: Choose three disjoint subsets X, Y, Z among the columns of H.
  Search for words having exactly $p, p, 0$ ones in those column sets and exactly $w - 2p$ nonzero in the remaining columns.

One iteration of Stern's algorithm

Let $p \in \{0, 1, \ldots, w\}$ and $\ell \in \{0, 1, \ldots, n-k\}$; $\ell \approx \lg \binom{k/2}{p}$.

- Select $n-k$ linearly independent columns; apply elementary row ops to get the identity matrix
- Divide remaining k columns into two subsets X and Y.
- Form a set Z of ℓ rows



- For every size-p subset A of X compute the ℓ-bit vector $\pi(A) = \sum_{i \in Z, j \in A} H_{i,j}$. Similarly, compute $\pi(B)$.
- For each collision $\pi(A) = \pi(B)$ compute the sum of the $2p$ columns in $A \cup B$. This sum is an $(n-k)$-bit vector.
- **If** the sum has weight $w - 2p$, we obtain 0 by adding the corresponding $w - 2p$ columns in the $(n-k) \times (n-k)$ submatrix.

  Else select $n-k$ new columns.

1. Motivation

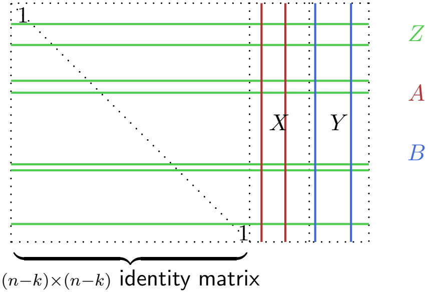
2. Review of the McEliece cryptosystem

3. Stern's attack

4. Attack optimization and comparison

Bernstein, Lange, P. at PQCrypto 2008:

Step 1



- Starting linear algebra part by using column selection from previous iteration.
- Forcing more existing pivots: *reuse exactly $n-k-c$ column selections (Canteaut et al.: $c=1$)*
- Faster pivoting
- Multiple choices of Z: *allow m disjoint sets $Z_1, \ldots, Z_m$ s.t. the word we're looking for has weight $p, p, 0 \ldots, 0$ on the sets $X, Y, Z_1, \ldots, Z_m$*

Step 2

- Reusing additions of the ℓ-bit vectors for p-element subsets A of X
- Faster additions after collisions: *consider at most w instead of $n-k$ cols*

Iterations

- Stern: iterations are independent (in each step $n-k$ linearly independent columns are randomly chosen);
- Our attack reuses existing pivots: Number of errors in the selected $n-k$ columns is correlated with the number of errors in the columns selected in the next iteration.
- Extreme case $c=1$ considered by Canteaut et al.: swapping one selected column for one deselected column is quite likely to preserve the number of errors in the selected columns.
- We analyzed the impact of selecting c new columns on the number of iterations with a Markov chain computation (generalizing from Canteaut et al.)

  `www.win.tue.nl/~cpeters/mceliece.html`
- Program can be used to optimize our attack parameters.

Complexity

- Canteaut, Chabaud, and Sendrier: an attacker can decode 50 errors in a $[1024, 524]$ code over $\mathbf{F}_2$ in $2^{64.1}$ bit operations.

- Choosing parameters $p = 2$, $m = 2$, $\ell = 20$, $c = 7$, and $r = 7$ in our new attack shows that the same computation can be done in only $2^{60.55}$ bit operations, almost a $12\times$ improvement over Canteaut et al.

- The number of iterations drops from $9.85 \cdot 10^{11}$ to $4.21 \cdot 10^{11}$, and the number of bit operations per iteration drops from $20 \cdot 10^6$ to $4 \cdot 10^6$.

Running time in practice

- Our attack software extracts a plaintext from a ciphertext by decoding 50 errors in a $[1024, 524]$ binary code.
- Attack on a single computer with a 2.4GHz Intel Core 2 Quad Q6600 CPU would need, on average, approximately 1400 days (2^{58} CPU cycles) to complete the attack.
- Running the software on 200 such computers would reduce the average time to one week.
- Canteaut, Chabaud, and Sendrier: implementation on a 433MHz DEC Alpha CPU; one such computer would need approximately 7400000 days (2^{68} CPU cycles).
- Note: Hardware improvements only reduce 7400000 days to 220000 days.
- The remaining speedup factor of 150 comes from our improvements of the attack itself.

First successful attack

We were able to extract a plaintext from a ciphertext by decoding 50 errors in a $[1024, 524]$ binary code.

- There were about 200 computers involved, with about 300 cores
- Computation finished in under 90 days
  (most of the cores put in far fewer than 90 days of work; some of which were considerably slower than a Core 2)
- Used about 8000 core-days
- Error vector found by Walton cluster at SFI/HEA Irish Centre of High-End Computing (ICHEC)
- Improved attack such that only 5000 core-days would be needed on average

Conclusions

- Our attack demonstrated that the parameters were chosen too small.
- It should not be interpreted as destroying the McEliece cryptosystem.
- In fact, the best known attacks are exponential in the main parameter and thus larger parameters lead to secure systems.

Thank you for your attention!