Wild McEliece

Christiane Peters

Technische Universiteit Eindhoven

joint work with Daniel J. Bernstein and Tanja Lange

SAC 2010

August 12, 2010

1. Motivation

2. Background on the McEliece cryptosystem

3. Wild McEliece

4. Decoding Wild Goppa codes

5. Attacks

6. Parameters

A code-based cryptosystem

- Code-based cryptography was proposed in 1978 by McEliece.
- Encryption is very efficient: matrix-vector multiplication.
- Patterson's decoding algorithm for binary Goppa codes also makes decryption efficient.
- Drawback of the system: public key is large.

Reducing the key size

- Binary Goppa code parameters achieving 128-bit security against the best-known attack (Bernstein, Lange, P., PQCrypto 2008) produce a 1537536-bit key.
- Smaller-key variants use other codes such as Reed-Solomon codes, generalized Reed-Solomon codes, quasi-dyadic codes or geometric Goppa codes.
- Unfortunately, many proposals turned out to be breakable.
- Goppa codes are the most confidence-inspiring choice.
- Using Goppa codes over larger fields decreases the key size at the same security level against information-set decoding (P., PQCrypto 2010).
- A Goppa code over $\mathbf{F}_{31}$ leads to a 725741-bit key for 128-bit security.
- This paper: Wild Goppa codes.

Encryption

- Given public system parameters n, k, w.
- The public key is a random-looking $k \times n$ matrix $\hat{G}$ with entries in $\mathbf{F}_q$.
- Encrypt a message $m \in \mathbf{F}_q^k$ as

$$m\hat{G} + e$$

  where $e \in \mathbf{F}_q^n$ is a random error vector of weight w.
- An attacker needs to correct w errors to find m.
- Decoding is not easy without knowing the structure of the code generated by $\hat{G}$.

Secret key

The public key $\hat{G}$ has a hidden Goppa-code structure allowing fast decoding:

$$\hat{G} = SGP$$

where

- G is the generator matrix of a Goppa code Γ of length n and dimension k and error-correcting capability w;
- S is a random $k \times k$ invertible matrix; and
- P is a random $n \times n$ permutation matrix.

The triple (G, S, P) forms the secret key.

Note: Detecting this structure, i.e., finding G given $\hat{G}$, seems even more difficult than attacking a random $\hat{G}$.

Decryption

The legitimate receiver knows S, G and P with $\hat{G} = SGP$ and a decoding algorithm for Γ.

How to decrypt $y = m\hat{G} + e$.

1. Compute $yP^{-1} = mSG + eP^{-1}$.
2. Apply the decoding algorithm of Γ to find mSG which is a codeword in Γ from which one obtains m.

Goppa codes

- Fix a prime power q; a positive integer m, a positive integer $n \leq q^m$; an integer $t < \frac{n}{m}$;
- distinct elements $a_1, \ldots, a_n$ in $\mathbf{F}_{q^m}$;
- and a polynomial $g(x)$ in $\mathbf{F}_{q^m}[x]$ of degree t such that $g(a_i) \neq 0$ for all i.

The Goppa code $\Gamma_q(a_1, \ldots, a_n, g)$ consists of all words $c = (c_1, \ldots, c_n)$ in $\mathbf{F}_q^n$ with

$$\sum_{i=1}^{n} \frac{c_i}{x - a_i} \equiv 0 \pmod{g(x)}$$

- $\Gamma_q(a_1, \ldots, a_n, g)$ has length n and dimension $k \geq n - mt$.
- The minimum distance is at least $\deg g + 1 = t + 1$ (in the binary case $2t + 1$).

Proposal

Use the McEliece cryptosystem with Goppa codes of the form

$$\Gamma_q(a_1, \ldots, a_n, g^{q-1})$$

where g is an irreducible monic polynomial in $\mathbf{F}_{q^m}[x]$ of degree t.

- Note the exponent $q-1$ in g^{q-1}.
- We refer to these codes as wild Goppa codes.

Minimum distance of wild Goppa codes

Theorem (Sugiyama-Kasahara-Hirasawa-Namekawa, 1976)

$$\Gamma_q(a_1, \ldots, a_n, g^{q-1}) = \Gamma_q(a_1, \ldots, a_n, g^q)$$

for a monic squarefree polynomial $g(x)$ in $\mathbf{F}_{q^m}[x]$ of degree t.

- Our paper contains a streamlined proof.
- The case $q = 2$ of this theorem is due to Goppa, using a different proof that can be found in many textbooks.

Error-correcting capability

- Since $\Gamma_q(\ldots, g^{q-1}) = \Gamma_q(\ldots, g^q)$ the minimum distance of $\Gamma_q(\ldots, g^{q-1})$ equals the one of $\Gamma_q(\ldots, g^q)$ and is thus $\geq \deg g^q + 1 = qt + 1$.

- We present an alternant decoder who allows to efficiently correct $\lfloor qt/2 \rfloor$ errors for $\Gamma_q(\ldots, g^{q-1})$.

- Note that the number of efficiently decodable errors increases by a factor of $q/(q-1)$ while the dimension $n - m(q-1)t$ of $\Gamma_q(\ldots, g^{q-1})$ stays the same.

The "wild" terminology.

- A prime p ramifies in a number field L if the unique factorization $p\mathcal{O}_L = Q_1^{e_1} Q_2^{e_2} \cdots$ has an exponent e_i larger than 1, where $\mathcal{O}_L$ is the ring of integers of L and $Q_1, Q_2, \ldots$ are distinct maximal ideals of $\mathcal{O}_L$.
- Each Q_i with $e_i > 1$ is ramified over p; this ramification is wild if e_i is divisible by p.
- If $\mathcal{O}_L/p$ is $\mathbf{F}_p[x]/f$ for f a monic polynomial in $\mathbf{F}_p[x]$. Then $Q_1, Q_2, \ldots$ correspond to the irreducible factors of f, and $e_1, e_2, \ldots$ correspond to the exponents in the factorization of f.
- In particular, the ramification corresponding to an irreducible factor ϕ of f is wild if and only if the exponent is divisible by p.
- We also refer to φ^p as being wild, and refer to the corresponding Goppa codes as wild Goppa codes.

Polynomial description of Goppa codes

Recall that

$$\begin{aligned}\Gamma &= \Gamma_q(a_1, \ldots, a_n, g^q)\\ &\subseteq \Gamma_{q^m}(a_1, \ldots, a_n, g^q)\\ &= \left\{ \left(\frac{f(a_1)}{h'(a_1)}, \ldots, \frac{f(a_n)}{h'(a_n)} \right) : f \in g^q \mathbf{F}_{q^m}[x], \deg f < n \right\}\end{aligned}$$

where $h = (x - a_1) \cdots (x - a_n)$.

- View target codeword $c = (c_1, \ldots, c_n) \in \Gamma$ as a sequence $(f(a_1)/h'(a_1), \ldots, f(a_n)/h'(a_n))$ of function values, where f is a multiple of g^q of degree below n.

Classical decoding

Given y, a word of distance $\lfloor qt/2 \rfloor$ from our target codeword.

Reconstruct c from $y = (y_1, \ldots, y_n)$ as follows:

- Interpolate $y_1 h'(a_1)/g(a_1)^q, \ldots, y_n h'(a_n)/g(a_n)^q$ into a polynomial φ: i.e., construct the unique $\varphi \in \mathbf{F}_{q^m}[x]$ such that $\varphi(a_i) = y_i h'(a_i)/g(a_i)^q$ and $\deg \varphi < n$.
- Compute the continued fraction of φ/h to degree $\lfloor qt/2 \rfloor$: i.e., apply the Euclidean algorithm to h and φ, stopping with the first remainder $v_0 h - v_1 \varphi$ of degree $< n - \lfloor qt/2 \rfloor$.
- Compute $f = (\varphi - v_0 h / v_1) g^q$.
- Compute $c = (f(a_1)/h'(a_1), \ldots, f(a_n)/h'(a_n))$.

This algorithm uses $n^{1+o(1)}$ operations in $\mathbf{F}_{q^m}$ using standard FFT-based subroutines.

- A Python script can be found on my website: `http://www.win.tue.nl/~cpeters/wild.html`

Decoders

- Can use any Reed-Solomon decoder to reconstruct f/g^q from the values $f(a_1)/g(a_1)^q, \ldots, f(a_n)/g(a_n)^q$ with $\lfloor qt/2 \rfloor$ errors.

- This is an illustration of the following sequence of standard transformations:

  Reed–Solomon decoder ⇒ generalized Reed–Solomon decoder
  ⇒ alternant decoder ⇒ Goppa decoder.

- The resulting decoder corrects $\lfloor (\deg g)/2 \rfloor$ errors for general Goppa codes $\Gamma_q(a_1, \ldots, a_n, g)$.

- In particular, $\lfloor q(\deg g)/2 \rfloor$ errors for $\Gamma_q(a_1, \ldots, a_n, g^q)$; and so $\lfloor q(\deg g)/2 \rfloor$ errors for $\Gamma_q(a_1, \ldots, a_n, g^{q-1})$.

List decoding

- Using the Guruswami–Sudan list-decoding algorithm we can efficiently correct $n - \sqrt{n(n-qt)} > \lfloor qt/2 \rfloor$ errors in the function values $f(a_1)/g(a_1)^q, \ldots, f(a_n)/g(a_n)^q$.
- Not as fast as a classical decoder but still takes polynomial time.
- Consequently we can handle $n - \sqrt{n(n-qt)}$ errors in the wild Goppa code $\Gamma_q(a_1, \ldots, a_n, g^{q-1})$.

Note:

- This algorithm can produce several possible codewords c. No problem for CCA2-secure variants of the McEliece system (Kobara, Imai, PKC 2001).
- We do not claim that this algorithm is the fastest possible decoder. Bernstein (2008) obtains for $q = 2$ the same error-correcting capability using a more complicated Patterson-like algorithm.

Attacks on Wild McEliece

- The wild McEliece cryptosystem includes, as a special case, the original McEliece cryptosystem.

- A complete break of the wild McEliece cryptosystem would therefore imply a complete break of the original McEliece cryptosystem.

Polynomial-searching attacks

- There are approximately q^{mt}/t monic irreducible polynomials g of degree t in $\mathbf{F}_{q^m}[x]$, and therefore approximately q^{mt}/t choices of g^{q-1}.

- An attacker can try to guess the Goppa polynomial g^{q-1} and then apply Sendrier's "support-splitting algorithm" to compute a permutation-equivalent code using the set $\{a_1, \ldots, a_n\}$.

- The support-splitting algorithm takes $\{a_1, \ldots, a_n\}$ as an input along with g.

Defenses

The first defense is well known and appears to be strong:

- Keep q^{mt}/t extremely large, so that guessing g^{q-1} has negligible chance of success. Our recommended parameters have q^{mt}/t dropping as q grows.

The second defense is unusual (strength is unclear):

- It is traditional, although not universal, to take $n = 2^m$ and $q = 2$, so that the only possible set $\{a_1, \ldots, a_n\}$ is $\mathbf{F}_{2^m}$.
- Keep n noticeably lower than q^m, so that there are many possible subsets $\{a_1, \ldots, a_n\}$ of $\mathbf{F}_{q^m}$.
- Can the support-splitting idea be generalized to handle many sets $\{a_1, \ldots, a_n\}$ simultaneously?

Information-set decoding

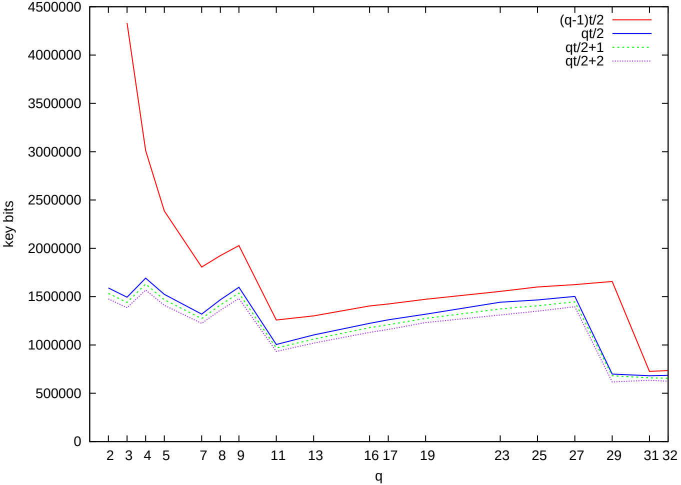
- The top threat against the original McEliece cryptosystem is information-set decoding.

- The same attack also appears to be the top threat against the wild McEliece cryptosystem for $\mathbf{F}_3$, $\mathbf{F}_4$, etc.

- Use complexity analysis of state-of-the-art information-set decoding for linear codes over $\mathbf{F}_q$ from [P. 2010] to find parameters (q, n, k, t) for Wild McEliece.

Key sizes for various q at a 128-bit security level

McEliece with $\Gamma_q(a_1, \ldots, a_n, g^{q-1})$ and $\lfloor (q-1)t/2 \rfloor$, $\lfloor qt/2 \rfloor$, $\lfloor qt/2 \rfloor + 1$, or $\lfloor qt/2 \rfloor + 2$ added errors.



Thank you for your attention!