Code-based Cryptography

Christiane Peters

Technical University of Denmark

ECC 2011
September 20, 2011

Code-based cryptography

1. Background
2. The McEliece cryptosystem
3. Information-set-decoding attacks
4. Designs: Wild McEliece
5. Announcements

1. Background

2. The McEliece cryptosystem

3. Information-set-decoding attacks

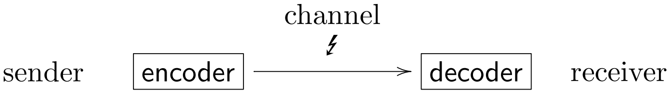
4. Designs: Wild McEliece

5. Announcements

Coding Theory

- An encoder transforms a message word into a codeword by adding redundancy.

- Goal: protect against errors in a noisy channel.



- The decoder uses a decoding algorithm to correct errors which might have occurred during transmission.

Error-correcting linear codes

- A linear code C of length n and dimension k is a k-dimensional subspace of $\mathbf{F}_q^n$.

- A generator matrix for C is a $k \times n$ matrix G such that $C = \left\{ m\, G : m \in \mathbf{F}_q^k \right\}$.

- The matrix G corresponds to a map $\mathbf{F}_q^k \to \mathbf{F}_q^n$ sending a message m of length k to a length-n codeword in $\mathbf{F}_q^n$.

Generator matrix of a linear code

The rows of the matrix

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

generate a linear code of length $n = 7$ and dimension $k = 4$ over $\mathbf{F}_2$.

Example of a codeword: $c = (0011)G = (0011010)$.

Hamming distance

- The Hamming distance between two words in $\mathbf{F}_q^n$ is the number of coordinates where they differ.
- The Hamming weight of a word is the number of non-zero coordinates.
- The minimum distance of a linear code C is the smallest Hamming weight of a non-zero codeword in C.

The example code is in fact the $(7, 4, 3)$ binary Hamming code which has minimum distance 3. And the example codeword has minimum weight $c = (0011010)$.

Decoding problem

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

There are lots of code families with fast decoding algorithms

- E.g., Hamming codes, BCH codes, Reed-Solomon codes, Goppa codes/alternant codes, Gabidulin codes, Reed-Muller codes, Algebraic-geometric codes, etc.

Generic decoding is hard

However, given a binary linear code with no obvious structure.

- Berlekamp, McEliece, van Tilborg (1978) showed that the general decoding problem for linear codes over $\mathbf{F}_2$ is NP-complete.

- About $2^{(0.5+o(1))n/\log_2(n)}$ binary operations required for a code of length n and dimension $\approx 0.5n$.

Parity-check matrix of a linear code

- Recall that a linear code C is generated by some matrix G
- Switch perspective and look at the corresponding parity-check matrix H.

$$H \quad G^T \quad = 0.$$

- In particular, $Hc^T = 0$ for all codewords c.
- Use Gaussian elimination to compute the $(n-k) \times n$ kernel matrix H from given G.

Syndrome decoding

- Decoder gets input $y \in \mathbf{F}_q^n$ and tries to determine an error vector e of a given weight w such that $c = y - e$ is a codeword.

Syndrome-formulation of the problem:

- Given y compute the syndrome

$$s = Hy^T = H(c + e)^T = He^T.$$

- Tricky part is to find a weight-w word e such that $s = He^T$.

1. Background

2. The McEliece cryptosystem

3. Information-set-decoding attacks

4. Designs: Wild McEliece

5. Announcements

Assumptions

- This talk looks at “text-book” versions of cryptosystems.
- Plaintexts are not randomized.
- There exist CCA2-secure conversions of code-based cryptography which should be used when implementing the systems.

Code-based cryptography

- McEliece proposed a public-key cryptosystem based on error-correcting codes in 1978.

- Secret key is a linear error-correcting code with an efficient decoding algorithm.

- Public key is a transformation of the secret inner code which is hard to decode.

Encryption

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

$$mG + e$$

where $e \in \mathbf{F}_q^n$ is a random error vector of weight w.

Secret key

The public key G has a hidden Goppa-code structure allowing fast decoding:

$$G = SG'P$$

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 G, seems even more difficult than attacking a random G.

Decryption

The legitimate receiver knows S, G' and P with $G = SG'P$ and a decoding algorithm for Γ.

How to decrypt $y = mG + e$.

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.

1. Background

2. The McEliece cryptosystem

3. Information-set-decoding attacks

4. Designs: Wild McEliece

5. Announcements

Generic attack

Disclaimer: for simplicity, focus on codes over $\mathbf{F}_2$ in the following.

Attacker tries to build a decoder which gets as input

- the parity-check matrix H (compute from public matrix G),
- the ciphertext $y \in \mathbf{F}_q^n$, and
- the public error weight w.

The algorithm tries to determine an error vector e of weight w such that $s = Hy^T = He^T$.

The best known generic decoders rely on information-set decoding.

Problem

$$\begin{array}{cccccc:c} & \vdots & & & & \vdots & \vdots \\ 1 & 1 & 1 & & & 0 & 0 \\ 1 & 0 & 0 & & & 1 & 1 \\ 0 & 1 & 1 & & \dots\dots & 0 & 0 \\ 0 & 1 & 0 & & & 1 & 1 \\ 1 & 1 & 1 & & & 1 & 0 \\ & \vdots & & & & \vdots & \vdots \\ \mathrm{c}_1 & \mathrm{c}_2 & \mathrm{c}_3 & & \dots\dots & \mathrm{c}_n & \mathrm{s} = \mathrm{c}_2 \oplus \mathrm{c}_3 \oplus \mathrm{c}_{18} \oplus \mathrm{c}_{20} \oplus \mathrm{c}_{24} \oplus \end{array}$$

Given an $(n-k) \times n$ matrix, a syndrome s.
Goal: find w columns of H with xor s.

Row randomization

$$
\begin{array}{ccccc:c}
 & \vdots & & & \vdots & \vdots \\
1 & 1 & 1 & & 0 & 0 \\
1 & 0 & 0 & & 1 & 1 \\
0 & 1 & 1 & \cdots\cdots & 0 & 0 \\
0 & 1 & 0 & & 1 & 1 \\
1 & 1 & 1 & & 1 & 0 \\
 & \vdots & & & \vdots & \vdots \\
\end{array}
$$

$\mathrm{c}_1 \mathrm{c}_2 \mathrm{c}_3 \quad \cdots\cdots \quad \mathrm{c}_n \quad \mathrm{s} = \mathrm{c}_2 \oplus \mathrm{c}_3 \oplus \mathrm{c}_{18} \oplus \mathrm{c}_{20} \oplus \mathrm{c}_{24} \oplus$

Can arbitrarily permute rows without changing the problem.

Goal: find w columns of H with xor s.

Row randomization

$$
\begin{array}{ccccc|c}
 & \vdots & & & \vdots & \vdots \\
1 & 0 & 0 & & 1 & 1 \\
1 & 1 & 1 & & 0 & 0 \\
0 & 1 & 1 & \cdots\cdots & 0 & 0 \\
0 & 1 & 0 & & 1 & 1 \\
1 & 1 & 1 & & 1 & 0 \\
 & \vdots & & & \vdots & \vdots \\
c_1 & c_2 & c_3 & \cdots\cdots & c_n & s = c_2 \oplus c_3 \oplus c_{18} \oplus c_{20} \oplus c_{24} \oplus
\end{array}
$$

Can arbitrarily permute rows without changing the problem.
Goal: find w columns of H with xor s.

Column normalization

$$
\begin{array}{ccccc|c}
 & \vdots & & & \vdots & \vdots \\
1 & 0 & 0 & & 1 & 1 \\
1 & 1 & 1 & & 0 & 0 \\
0 & 1 & 1 & \cdots\cdots & 0 & 0 \\
0 & 1 & 0 & & 1 & 1 \\
1 & 1 & 1 & & 1 & 0 \\
 & \vdots & & & \vdots & \vdots \\
c_1 & c_2 & c_3 & \cdots\cdots & c_n & s = c_2 \oplus c_3 \oplus c_{18} \oplus c_{20} \oplus c_{24} \oplus
\end{array}
$$

Can arbitrarily permute columns without changing the problem.

Goal: find w columns of H with xor s.

Column normalization

$$
\begin{array}{ccccc|c}
 & \vdots & & & \vdots & \vdots \\
0 & 1 & 0 & & 1 & 1 \\
1 & 1 & 1 & & 0 & 0 \\
1 & 0 & 1 & \ldots\ldots & 0 & 0 \\
1 & 0 & 0 & & 1 & 1 \\
1 & 1 & 1 & & 1 & 0 \\
 & \vdots & & & \vdots & \vdots \\
\hline
c_1 & c_2 & c_3 & \ldots\ldots & c_n & s = c_1 \oplus c_3 \oplus c_{18} \oplus c_{20} \oplus c_{24} \oplus
\end{array}
$$

Can arbitrarily permute columns without changing the problem.
Goal: find w columns of H with xor s.

Information-set decoding

$$
\begin{array}{cccccccccc|c}
1 & 0 & 0 & 0 & \cdots\cdots & 0 & 1 & \cdots\cdots & 1 & & 0 \\
0 & 1 & 0 & 0 & \cdots\cdots & 0 & 0 & \cdots\cdots & 1 & & 1 \\
0 & 0 & 1 & 0 & \cdots\cdots & 0 & 1 & \cdots\cdots & 0 & & 1 \\
0 & 0 & 0 & 1 & \cdots\cdots & 0 & 0 & \cdots\cdots & 1 & & 0 \\
 & & & & \ddots & & & & & & \vdots \\
0 & 0 & 0 & 0 & \cdots\cdots & 1 & 0 & \cdots\cdots & 1 & & 0 \\
\end{array}
$$

$$c_1 c_2 c_3 c_4 \quad \ldots \quad c_{n-k} \qquad\qquad c_n \quad s = c_3 \oplus c_7 \oplus c_{28} \oplus c_{30} \oplus c_{37} \oplus$$

Can add one column to another. Built identity matrix.
Goal: find w columns which xor s.

Basic information-set decoding

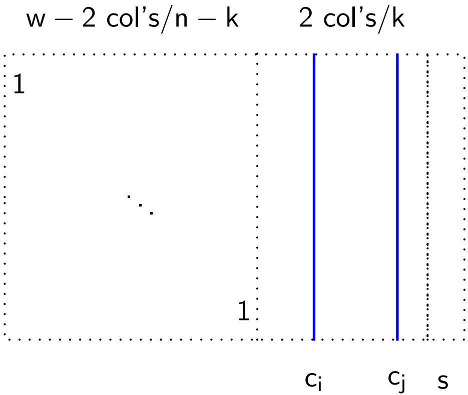
1962 Prange:

- Perhaps xor involves none of the last k columns.
- If so, immediately see that s is constructed from w columns of H.
- If not, re-randomize and restart.

1988 Lee–Brickell:

- More likely that xor involves exactly 2 of the last k columns.
- Check for each pair (i, j) with $n - k < i < j \leq n$ if $\mathrm{s} \oplus \mathrm{c_i} \oplus \mathrm{c_j}$ has weight $w - 2$.

Lee–Brickell



Check for each pair (i, j) with $n - k < i < j \leq n$ if $\mathrm{s} \oplus \mathrm{c_i} \oplus \mathrm{c_j}$ has weight $w - 2$.

Improvements

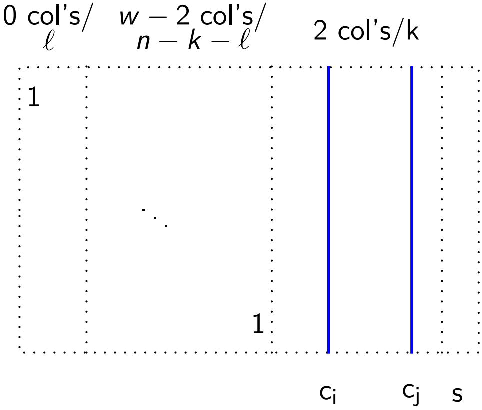
1989 Leon, 1989 Krouk:

- Check for each i,j whether s$\oplus$ c$_i\oplus$ c$_j$ has weight $w-2$ and the first ℓ bits all zero.
- Fast to test.

1989 Stern:

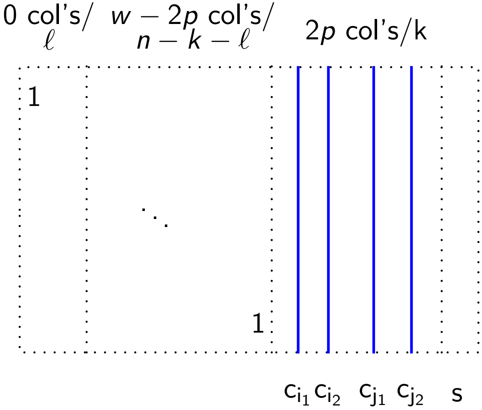
- Collision decoding: square-root improvement.
  Find collisions between first ℓ bits of $s \oplus c_i$ and the first ℓ bits of c_j.
- For each collision, check whether $s \oplus c_i \oplus c_j$ has weight $w-2$.

Collision decoding



Check for collisions on ℓ bits of $s \oplus c_i$ and c_j.

Collision decoding



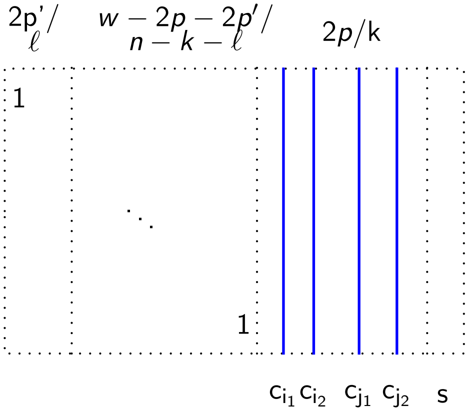
Check for collisions on ℓ bits of $s \oplus c_{i_1} \oplus \cdots \oplus c_{i_p}$ and $c_{j_1} \oplus \cdots \oplus c_{j_p}$.

Ball-collision decoding

Joint work with Dan Bernstein and Tanja Lange: Smaller decoding exponents: ball-collision decoding.

- Find collisions between the Hamming ball of radius p' around $\mathrm{s} \oplus \mathrm{c}_{i_1} \oplus \cdots \oplus \mathrm{c}_{i_p}$ and the Hamming ball of radius p' around $\mathrm{c}_{j_1} \oplus \cdots \oplus \mathrm{c}_{j_p}$.
- Main theorem: asymptotically get exponential speedup of ball-collision decoding over collision decoding.
- Reference implementation of ball-collision decoding: `http://cr.yp.to/ballcoll.html`

Ball-collision-decoding algorithm



Look for collisions among $s \oplus c_{i_1} \oplus \cdots \oplus c_{i_p} \oplus c_{l_1} \oplus \cdots \oplus c_{l_{p'}}$ and $c_{j_1} \oplus \cdots \oplus c_{j_p} \oplus c_{r_1} \oplus \cdots \oplus c_{r_{p'}}$.

1. Background

2. The McEliece cryptosystem

3. Information-set-decoding attacks

4. Designs: Wild McEliece

5. Announcements

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 $a_1, \dots, a_n \in \mathbf{F}_{q^m}$;

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

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

Properties of Goppa codes

- $\Gamma_q(a_1, \ldots, a_n, g)$ has length n and dimension $k \geq n - mt$.
- The minimum distance is at least $\deg g + 1 = t + 1$ (in the binary case $2t + 1$).
- Patterson decoding efficiently decodes t errors in the binary case; otherwise only $t/2$ errors can be corrected.

Key sizes for the classical binary codes

- Taking a binary Goppa code yields a 194KB public key for 128-bit security for the McEliece cryptosystem.

- Smaller-key variants use other codes such as Reed-Solomon codes, generalized Reed-Solomon codes, quasi-cyclic codes, quasi-dyadic codes or geometric Goppa codes.

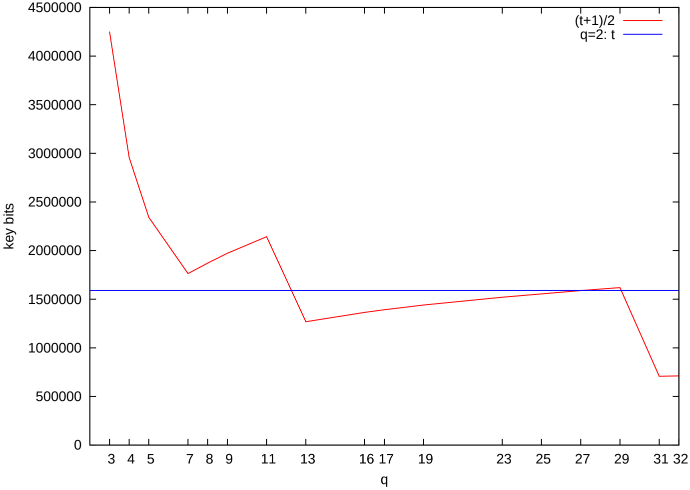
Goal: reduce the key size!

Reducing the key size

- Classical 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).
- Taking a Goppa code over $\mathbf{F}_{31}$ yields a 87KB public key for 128-bit security for the McEliece cryptosystem.
- Drawback: can correct only $t/2$ errors if $q > 2$ (vs. t in the binary case).
- However, Goppa codes over smaller fields such as $\mathbf{F}_3$ are not competitive in key size with codes over $\mathbf{F}_2$.

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

McEliece with $\Gamma_q(a_1, \ldots, a_n, g)$ with an alternant decoder.



Proposal: Wild McEliece

Bernstein, Lange, P. at SAC 2010:

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.

- 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 that allows efficient correction of $\lfloor qt/2 \rfloor$ 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.

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

$$\left(\frac{f(a_1)}{h'(a_1)},\ldots,\frac{f(a_n)}{h'(a_n)}\right)$$

  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

  $$\frac{y_1 h'(a_1)}{g(a_1)^q}, \frac{y_2 h'(a_2)}{g(a_2)^q}, \ldots, \frac{y_n h'(a_n)}{g(a_n)^q}$$

  into a degree-n polynomial $\varphi \in \mathbf{F}_{q^m}[x]$.
- 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 $c = (f(a_1)/h'(a_1), \ldots, f(a_n)/h'(a_n))$.

Efficiency

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://www2.mat.dtu.dk/people/C.Peters/wild.html`

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.

Security evaluation

- 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.

Generic attacks

- 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.

Structural attacks

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}.

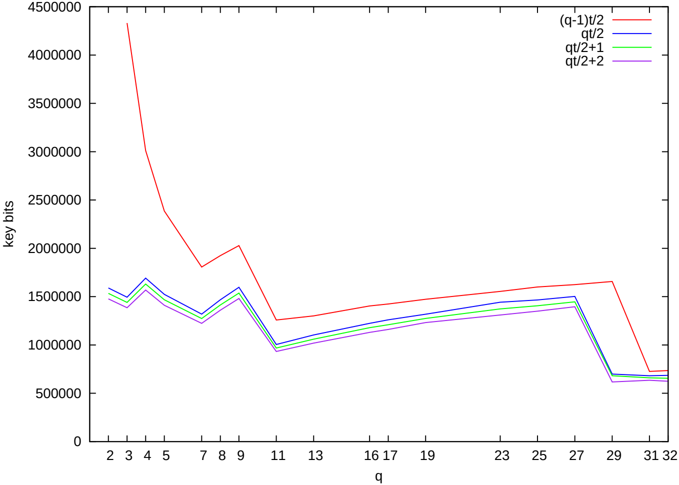
- An attacker can try to guess the Goppa polynomial g^{q-1} and then apply Sendrier's "support-splitting algorithm" to compute a permutation-equivalent code using the set $\{a_1, \ldots, a_n\}$.

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

Defenses are discussed in our "Wild" paper.

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.



Hiding wildness

Beelen: proof of Sugiyama et al.'s theorem based on Chinese Remainder Theorem. Hide Goppa codes by using an extra factor.

Wild McEliece Incognito (Bernstein-Lange-P., to appear at PQCrypto 2011):

- Avoid the potential problem of polynomial-searching attacks by using codes with Goppa polynomial $f \cdot g^{q-1}$.
- In particular: Goppa codes of the form $\Gamma_q(a_1, \ldots, a_n, fg^{q-1})$ where f and g are squarefree monic polynomials in $\mathbf{F}_{q^m}[x]$ of degree s and t, respectively.
- Choose f so that the number of polynomials fg^{q-1} becomes too large to search.

Getting wilder

- For $\deg(f) = s$ and $\deg(g) = t$ the codes can correct up to $\lfloor (s + qt)/2 \rfloor$ errors.

- Efficient decoding of $\lfloor (s + qt)/2 \rfloor$ errors can be done using the same alternant decoders as described before.

- Still "wild."

Wildness comparison

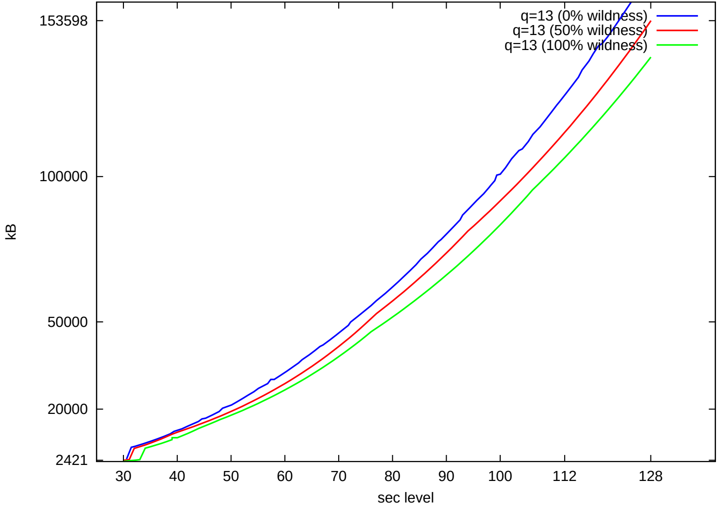
Given a wild Goppa code $\Gamma_q(a_1, \ldots, a_n, fg^{q-1})$ with f and g both squarefree and f a degree-s polynomial and g a degree t-polynomial.

- Restrict to "50% wildness", i.e., where the degrees of f and g^{q-1} are balanced by setting $s = (q-1)t$.

- Experiment: consider wild McEliece keys with 0%, 50%, and 100% wildness percentage for $q = 13$.

Key sizes for $q = 13$ for various security levels

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



1. Background

2. The McEliece cryptosystem

3. Information-set-decoding attacks

4. Designs: Wild McEliece

5. Announcements

Announcing cryptanalytic challenges

- Measure and focus progress in attacking the “wild McEliece” cryptosystem.

  `http://pqcrypto.org/wild-challenges.html`

- Each “wild” challenge consists of a public key and a ciphertext.
- Find the matching plaintext or even try to find the secret keys.

PQCrypto 2011

Nov 29 – Dec 2, Taipei

http://pq.crypto.tw/pqc11/

Code-based cryptography workshop

DTU, Lyngby

Spring 2012

Contact me for more information.

Thank you for your attention!