Wild McEliece Incognito

Christiane Peters

Technical University of Denmark

joint work with
Daniel J. Bernstein and Tanja Lange

PQCrypto 2011

December 2, 2011

1. Background

2. Wild McEliece Incognito

3. Attacks and defenses

4. Parameters

5. Challenges

Note

- 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.

The McEliece cryptosystem

- Take a linear error-correcting code which can efficiently correct w errors.

- Publish a permutation-equivalent code and the error weight w.

- Encryption: sender embeds message into codeword in public code and adds w errors.

The well-known drawback

Key size determined by parameters of the secret code and its error correcting capability.

McEliece 1978: use as secret code a classical binary Goppa code $\Gamma_2(a_1, \ldots, a_n, g)$ (original parameters $n = 1024$, $w = \deg(g) = 50$).

Shrink keys: find a secret code

- allowing a compact representation
- such that the structure is not detectable from the public code given by some matrix;
- evaluate its security.

Attacking code-based cryptography

Two types of attacks in code-based cryptography:

- Generic decoding attacks: correct w errors in an arbitrary linear code.
- Structural attacks: try to find the secret code given the generator matrix of the public code.

Generic decoding attacks

Best known generic decoding attack relies on so-called information-set decoding.

Quite a long history:

1962 Prange; 1981 Clark (crediting Omura); 1988 Lee–Brickell; 1988 Leon; 1989 Krouk; 1989 Stern; 1989 Dumer; 1990 Coffey–Goodman; 1990 van Tilburg; 1991 Dumer; 1991 Coffey–Goodman–Farrell; 1993 Chabanne–Courteau; 1993 Chabaud; 1994 van Tilburg; 1994 Canteaut–Chabanne; 1998 Canteaut–Chabaud; 1998 Canteaut–Sendrier; 2008 Bernstein–Lange–P.; 2009 Finiasz–Sendrier; 2010 P.; 2011 Bernstein–Lange–P.; 2011 Sendrier; 2011 May–Meurer–Thomae.

Security

Suitable codes for code-based cryptography are such that the best attack is generic decoding.

- We say that a system has *b-bit security* if an attacker needs at least 2^b bit operations to decrypt a single ciphertext.

Structural attacks

- Generalized Reed–Solomon codes – broken by Sidelnikov and Shestakov (GRS variants broken by Wieschebrink)
- Gabidulin codes: broken by Gibson (variants broken by Overbeck)
- AG codes corresponding to GRS codes – broken by Minder, Minder–Faure, Pellikaan et al. (attacks generalize the Sidelnikov–Shestakov attack on GRS codes)
- quasi-cyclic Goppa codes and "non-binary" quasi-dyadic Goppa codes – broken (Gauthier–Leander, Faugère et al.)

Holding up

So far no structural attacks on code-based crypto using classical Goppa codes $\Gamma_q(a_1, \ldots, a_n, g)$.

- For given code parameters can build many different Goppa codes (thanks to subfield-subcode construction).

- Monoidic codes need further investigation.

Future work:

- Subfield subcodes of algebraic-geometric codes (avoid Sidelnikov–Shestakov-like attacks).

Way(s) to go

The more errors the secret code can correct — the harder the generic decoding problem.

- Yesterday's talk by Dan Bernstein: 1 extra error translates to a factor 5 in the complexity for generic attacks.

Two ways:

1. Improve decoding
   - Use list decoding for Goppa codes over $\mathbf{F}_2$ (see e.g., yesterday's talk by Dan Bernstein).
   - Better decoding algorithms for Goppa codes over bigger fields (yesterday's talk by Rafael Misoczki; improvement for $q = 3$).
2. Use subfamily of q-ary Goppa codes which can correct more errors with classical decoding algorithms.

Goppa codes

- Fix a prime power q; a positive integer m, a positive integer $n \leq q^m$; a positive integer t; distinct $a_1, \ldots, a_n \in \mathbf{F}_{q^m}$;

- 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

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

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 $\deg g + 1 = t + 1$.
- However, a Goppa code over $\mathbf{F}_2$ has minimum distance at least $2t + 1$.
- Patterson decoding efficiently decodes t errors in the binary case whereas Berlekamp's algorithm corrects only $t/2$ errors for any q-ary Goppa code.

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.

Proposal: Wild McEliece Incognito

Beelen: hide wildness by using an extra factor.

This paper:

Use the McEliece cryptosystem with Goppa codes of the form

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

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

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

Features

- If $g = 1$ then $\Gamma_q(a_1, \ldots, a_n, fg^{q-1})$ is the squarefree Goppa code $\Gamma_q(a_1, \ldots, a_n, f)$.

- If $f = 1$ then $\Gamma_q(a_1, \ldots, a_n, fg^{q-1})$ is the *wild Goppa code* $\Gamma_q(a_1, \ldots, a_n, g^{q-1})$ (BLP10).

- The Goppa code with polynomial fg^{q-1} has dimension at least $n - m(s + (q-1)t)$, where s is the degree of f and t is the degree of g.

Minimum distance of wild Goppa codes

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

$$\Gamma_q(a_1, \dots, a_n, fg^{q-1}) = \Gamma_q(a_1, \dots, a_n, fg^q)$$

for coprime monic <u>*squarefree*</u> *polynomials* $g(x)$ *and* $f(x)$ *in* $\mathbf{F}_{q^m}[x]$ *of degree* t *and degree* s*, respectively.*

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

- Alternant decoder efficiently corrects $\lfloor (qt+s)/2 \rfloor$ errors for $\Gamma_q(\ldots, fg^{q-1})$.

- Note that the number of efficiently decodable errors increases by up to a factor of $q/(q-1)$ while the dimension stays the same.

Decoding

- Can use any Reed–Solomon decoder to correct $\lfloor s + qt/2 \rfloor$ errors for $\Gamma_q(a_1, \ldots, a_n, fg^{q-1})$.

Illustration of the following sequence of standard transformations:

$$\text{Reed–Solomon decoder} \Rightarrow \text{generalized Reed–Solomon decoder}$$
$$\Rightarrow \text{alternant decoder} \Rightarrow \text{Goppa decoder.}$$

In particular, can use list decoding:

- simplest case: Guruswami–Sudan list-decoding for Reed-Solomon codes.
- More sophisticated list-decoding algorithms can correct more errors and are faster.

Note: this talk will focus on deterministic decoding.

1. Background

2. Wild McEliece Incognito

3. Attacks and defenses

4. Parameters

5. Challenges

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

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

- 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}.

- Knowing g^{q-1} an attacker can try to apply Sendrier's "support-splitting algorithm" to compute a permutation-equivalent code
  - requires knowledge of the support elements $\{a_1, \ldots, a_n\}$ in order to start the algorithm.

Defenses for the wild system

1. Make polynomial searching hard:
   - Keep q^{mt}/t extremely large, so that guessing g^{q-1} has negligible chance of success.

2. Give up traditional support length $n = q^m$:
   - 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?

Incognito factor f

Completely avoid the potential problem of polynomial-searching attacks by using a code $\Gamma_q(a_1, \ldots, a_n, fg^{q-1})$:

- Choose f so that finding fg^{q-1} takes at least as much time as generic decoding.
- For the extremely paranoid: choose parameters such that there are at least 2^{2b} possible polynomials fg^{q-1} when aiming at b-bit security against ISD.
- Note that factorizability of fg^{q-1} is not analogous to the concatenated structure attacked by Sendrier in 1994.

Masking the structure further

Add extra protection against structural attacks using an idea by Berger and Loidreau (2005).

- Add ℓ additional rows to parity-check matrix.
- There are $\binom{k}{\ell}_q = \frac{(1-q^k)(1-q^{k-1})\cdots(1-q^{k-\ell+1})}{(1-q)(1-q^2)\cdots(1-q^\ell)}$ subspaces of dimension ℓ in a k-dimensional code over $\mathbf{F}_q$ (already big for $\ell = 1$).
- Mask unsuccessful for GRS codes (Wieschebrink PQC2010); attack not obviously applicable to unmask Goppa codes.
- Small increase in key size: public key has $(n-k+\ell)(k-\ell)$ entries instead of $(n-k)k$ (systematic form).

Parameter choice

- 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, s, t) for Wild McEliece.

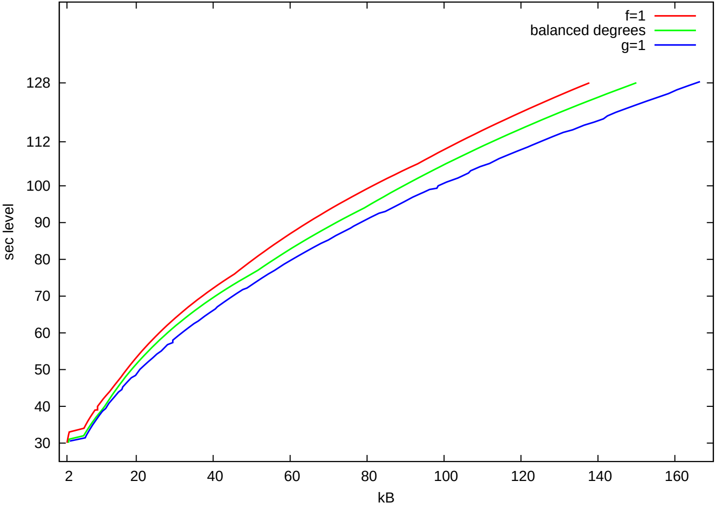
Parameter suggestions for 128-bit security

q	key size	n	k	s	t	w	p
3	186 kB	2136	1492	0	46	69	100%
4	210 kB	2252	1766	0	27	54	100%
5	191 kB	1878	1398	0	24	60	100%
7	170 kB	1602	1186	8	16	60	92%
8	187 kB	1628	1204	8	14	60	92%
9	205 kB	1668	1244	10	12	59	91%
11	129 kB	1272	951	17	9	58	84%
13	142 kB	1336	1033	17	7	54	83%
16	157 kB	1328	1010	16	6	56	85%
17	162 kB	1404	1113	17	5	51	82%
19	169 kB	1336	1015	17	5	56	84%
23	183 kB	1370	1058	16	4	54	85%
25	189 kB	1314	972	18	4	59	84%
27	200 kB	1500	1218	42	2	48	55%
29	199 kB	1390	1081	19	3	53	82%
31	88 kB	856	626	25	3	59	78%
32	89 kB	852	618	24	3	60	79%

Minimized key size against q-ary ISD attacks.

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. Wild McEliece Incognito
3. Attacks and defenses
4. Parameters
5. Challenges

How strong is the (wild) McEliece cryptosystem?

- Are there structural attacks against (wild) McEliece?
- How good/fast are the best generic attacks?
- In particular, measure progress of all sorts of attacks.
- How good are q-ary attacks?
  - Classical ISD vs. generalized statistical decoding (Niebuhr PQC2011).

Website (1)

Our challenges are online at

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

Each of our challenges is labelled by

1. "wild McEliece" for [BLP2010] or "wild McEliece incognito" for this paper;
2. a field size q ($q \geq 2$);
3. a key size expressed in kilobytes.

Website (2)

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

We intend to keep this web page up to date to show

- any solutions (plaintexts) sent to us — with credit to the first solver of each challenge;
- any secret keys sent to us — again with credit to the first solver of each challenge;
- cryptanalytic benchmarks — measurements of the speed of publicly available cryptanalytic software for the smaller challenges;
- predictions — estimates of how difficult the larger challenges will be to break.

Announcement

Code-based Cryptography
DTU, Lyngby
May 9–11, 2012

- Invited talks.
- Talks on recent results.
- Research retreat.

More information soon at

`http://www.agincc.mat.dtu.dk/`

Thank you for your attention!