Cryptanalysis of the McEliece cryptosystem over hyperelliptic codes

ACCT workshop
June 16-22, 2008, Pamporovo

Cedric Faure and Lorenz Minder
INRIA Rocquencourt, UC Berkeley

McEliece cryptosystem

Public key : $G_{\text{pub}} = SG_0P$
Secret key : G_0, S, P

Encryption : $\mathbf{c} = \mathbf{m}G_{\text{pub}} + \mathbf{e}$

Attack either on the ciphertext (decoding problem) or the public key (code identification problem)

History

Algebraic geometry codes are fast, with good correction capability. Why not use them for McEliece cryptosystem ?

Genus 0 : Generalized Reed-Solomon codes, broken by Sidelnikov and Shestakov in 1992.

Genus 1 : Elliptic codes, broken by Minder and Shokrollahi in 2007.

Genus 2 : Hyperelliptic codes, proposed by Janwa and Moreno in 1996, unattacked until today.

Outline of the talk

Mathematical definitions

Presentation of our algorithm

Algebraic geometry

Let $\mathcal{X}$ be a hyperelliptic curve of genus $g = 2$ over $\mathbb{A}_2(\mathbb{F}_q)$, defined by the equation :

$$y^2 + G(x)y = F(x), \text{ with } \deg(F) = 2g + 1, \text{ and } \deg(G) \leq g.$$

A divisor Δ over $\mathcal{X}$ is a formal finite sum of points of $\mathcal{X}$

$$\Delta = \sum_{P \in \mathcal{X}} n_P \langle P \rangle, \ \deg(\Delta) = \sum_{P \in \mathcal{X}} n_P, \ n_P \in \mathbb{Z}.$$

Jacobian group

Any rational function f over $\mathcal{X}$ has an associated divisor $\mathrm{div}(f)$:

$$\mathrm{div}(f) = \sum_{P \in \mathcal{X}} \mathrm{ord}_P(f)\langle P \rangle.$$

$$\deg(\mathrm{div}(f)) = 0$$

$$\mathrm{Jac}(\mathcal{X}) = \text{Divisors of degree } 0/\text{divisors of rational functions}$$

$$\mathrm{Jac}(\mathcal{X}) \simeq \mathcal{G} = \frac{\mathbb{Z}}{d_1\mathbb{Z}} \times \cdots \times \frac{\mathbb{Z}}{d_{2g}\mathbb{Z}}, \text{ with } d_1 | \ldots | d_{2g},\ d_1 | q-1$$

Geometric codes

Let Δ be a divisor of degree $k+1 \geq 2$ over $\mathcal{X}$.

$$\mathcal{L}(\Delta) = \{f \in \mathbb{F}_q(\mathcal{X}) | \text{div}(f) + \Delta \geq 0\} \cup \{0\}$$

is a vector space of dimension k.

$$\text{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n)) = \{(f(P_1), \ldots, f(P_n)) | f \in \mathcal{L}(\Delta)\}$$

If $(P_1, \ldots, P_n)$ are distinct, this is a linear code of length n, dimension k, and minimal distance $d \geq n - k - 1$.

For $c_i \in \mathbb{F}_q^*$, $\text{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n), (c_1, \ldots, c_n))$ is a directional scaling of the former code.

Our goal

Given $\mathcal{C} = \mathrm{AGC}(\mathcal{X}', \Delta', (P_1', \ldots, P_n'))$, where $\mathcal{X}', \Delta', (P_1', \ldots, P_n')$ are unknown,

we recover in polynomial (quartic) time $\mathcal{X}, \Delta, (P_1, \ldots, P_n), (c_1, \ldots, c_n)$ such that

$$\mathcal{C} = \mathrm{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n), (c_1, \ldots, c_n))$$

Assumptions

$$n \approx \mathbb{F}_q(\mathcal{X})$$

$\mathrm{gcd}(k+1, |\mathcal{G}|) = 1$, so $\Delta = (k+1)\Delta_0$.

Codewords of weight $n-k-1$ are easy to generate.

Outline of the attack

Recovering the Jacobian group structure

Recovering the curve equation

Recovering the coordinates of the evaluation points

Computing the scaling coefficients

Recovering the Jacobian structure

$$\mathrm{Jac}(\mathcal{X}) \overset{\varphi}{\simeq} \mathcal{G} = \frac{\mathbb{Z}}{d_1\mathbb{Z}} \times \cdots \times \frac{\mathbb{Z}}{d_{2g}\mathbb{Z}}$$

$$\tilde{z}_i = \varphi(\langle P_i \rangle - \Delta_0) \in \mathcal{G}$$

Let $\mathbf{x} \in \mathcal{C}$ be a codeword of weight $n-k-1$, with zero postions on $i_1, \ldots, i_{k+1}$. Then

$$\sum_{j=0}^{k+g-1} \tilde{z}_{i_j} = 0$$

Recovering the Jacobian structure

With slightly more than n equations, we recover the d_i and the $\tilde{z}_i$ in $O(n^4)$.

A statistical test on opposite points allow us to recover the value of $\delta_0 = \varphi(\Delta_0 - \langle \mathcal{O} \rangle)$ in $O(n^2)$ operations.

Recovering the curve equation

We generate (in $O(n^3)$) $\mathbf{v}, \mathbf{w} \in \mathcal{C}$ of weight $(n-k-1)$, with exactly $k-1$ zero position in common, and the remaining zeros on a pair of opposite points.

$$\frac{v_i}{w_i} = \frac{f_1}{f_2}(P_i) = \frac{ax_i + b}{cx_i + d}$$

where $a, b, c, d \in \mathbb{F}_q$ are unknown constants, and x_i is the X-coordinate of P_i.

Recovering the curve equation

$$\frac{v_i}{w_i} = \frac{f_1}{f_2}(P_i) = \frac{ax_i + b}{cx_i + d}$$

We guess the coordinates of 3 points $P_{k_1}, P_{k_2}, P_{k_3}$.
We recover the constants a, b, c, d.
We recover the X-coordinates of many P_i. (We use colinearity equations for Y-coordinates)

We need $O(n)$ guesses to recover the curve equation.

Recovering all the evaluation points

We know all the $\tilde{z}_i = \varphi(\langle P_i \rangle - \Delta_0) \in \mathcal{G}$

We know the curve equation, and the coordinates (x_i, y_i) of a quite large number of P_i.

The coordinates of the remaining P_i are computed by decomposition in $\mathcal{G}$ and point arithmetics over the curve, in $O(n \log n)$.

ecovering the distortion coefficients

$$\mathcal{C} = \mathsf{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n), (c_1, \ldots, c_n))$$

$c_1, \ldots, c_n \in \mathbb{F}_q$ are the only unknowns, we compute them in $O(n^3)$ by a simple matrix inversion.

Conclusions

Under reasonable assumptions, our attack breaks McEliece cryptosystem over hyperelliptic codes of genus 2, in time $O(n^4)$.

Over superior genus, this attack could work, with very low but non-zero probability.