Algebraic-Geometric Codes for Code-based Cryptography

Christiane Peters

HGI Seminar

Bochum – January 16, 2013

joint work with Daniel J. Bernstein and Tanja Lange

Outline

Code-based encryption scheme (Niederreiter version)

- Public system parameters are n, r, w.
- Public key: a random-looking $r \times n$ matrix $\mathbf{H}_{pub}$ with entries in $\mathbb{F}_q$.
- Input: a message $x \in \mathbb{F}_q^n$ of Hamming weight w.
- Encryption: compute the ciphertext $s = x \cdot \mathbf{H}_{pub}^t$.

Secret key

The public key $\mathbf{H}_{pub}$ has a hidden algebraic structure allowing fast decoding.

Decryption:

- Use linear algebra to undo the conversion from the public code $\mathcal{C}_{pub}$ to the secret code $\mathcal{C}$ and
- make use of the fast decoding algorithm for $\mathcal{C}$ to find low-weight message x.

Note: $\mathbf{H}_{pub}$ is related to a matrix H with

- $c \cdot H^t = 0$ for all codewords $c \in \mathcal{C}$
  (H is a parity-check matrix for $\mathcal{C}$.)

Attacks

There are basically two types of attacks in code-based cryptography.

1. Structural attacks
   - Find the secret code given $\mathbf{H}_{pub}$.

2. Decrypt a single ciphertext
   - Use a generic decoding algorithm (best known algorithms rely on information-set decoding).

Design goals

Public-key size

- Store redundancy part of a generator matrix in systematic form: $r(n-r)$ bits for an $[n, n-r]$ code.

Design goals

Public-key size

- Store redundancy part of a generator matrix in systematic form: $r(n-r)$ bits for an $[n, n-r]$ code.

Thwart structural attacks

- by carefully choosing the hidden code.

Design goals

Public-key size

- Store redundancy part of a generator matrix in systematic form: $r(n-r)$ bits for an $[n, n-r]$ code.

Thwart structural attacks

- by carefully choosing the hidden code.

Assuming that a structural attack is infeasible

- choose parameters n, r, w so that ISD takes at least 2^b bit ops to correct w errors in one single ciphertext (b-bit security).

Reed–Solomon codes

- Fix a prime power q;
- an integer $0 \leq t < q$;
- a primitive element $\alpha \in \mathbb{F}_q$.

The Reed–Solomon code

$$\left\{(f(0), f(1), f(\alpha), \ldots, f(\alpha^{q-2})) : f \in \mathbb{F}_q[x], \deg f < q - t\right\}$$

- has length q, dimension $q - t$, and
- minimum distance $t + 1$ (MDS code).
- Berlekamp's algorithm decodes $t/2$ errors in $O(q^2)$.

This is an example of an evaluation code.

Hidden algebraic structure allowing fast decoding?

- No.

Need to modify the code

- add certain defenses against structural attacks while maintaining good error-correction.

Defenses

Scaling

- Pick q elements $\gamma_1, \ldots, \gamma_q \in \mathbb{F}_q^*$ to produce codewords $(\gamma_1 c_1, \ldots, \gamma_q c_q)$.

Defenses

Scaling

- Pick q elements $\gamma_1, \ldots, \gamma_q \in \mathbb{F}_q^*$ to produce codewords $(\gamma_1 c_1, \ldots, \gamma_q c_q)$.

Permuting

- Pick a permutation $\pi \in S_q$ and permute the coordinates of the codewords to get $(c_{\pi(1)}, \ldots, c_{\pi(q)})$.

Defenses

Scaling

- Pick q elements $\gamma_1, \ldots, \gamma_q \in \mathbb{F}_q^*$ to produce codewords $(\gamma_1 c_1, \ldots, \gamma_q c_q)$.

Permuting

- Pick a permutation $\pi \in S_q$ and permute the coordinates of the codewords to get $(c_{\pi(1)}, \ldots, c_{\pi(q)})$.

Puncturing

- Consider the shortened code containing codewords of the form $(c_{i_1}, \ldots, c_{i_n})$ where $1 \leq i_1 < \cdots < i_n \leq q$.

Generalized Reed–Solomon code

- Fix integers n, t with $0 \leq t < n \leq q$;
- an ordered set of distinct elements $\{\alpha_1, \dots, \alpha_n\} \subseteq \mathbb{F}_q$;
- $\gamma_1, \dots, \gamma_n \in \mathbb{F}_q^*$ (not necessarily distinct).

The Generalized Reed–Solomon code

$$\{(\gamma_1 f(\alpha_1), \dots, \gamma_n f(\alpha_n)) : f \in \mathbb{F}_q[x], \deg f < n - t\}$$

- has length n, dimension $n - t$, and
- minimum distance $t + 1$ (MDS code).
- Can apply RS decoders to the punctured code after undoing the scaling and permuting.

A GRS parity-check matrix

A parity–check matrix of the Generalized Reed–Solomon code with parameters q, n, t and support $\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subseteq \mathbb{F}_q$ and scalars $\{\gamma_1, \ldots, \gamma_n\} \subseteq \mathbb{F}_q^*$ is given by

$$H = \begin{pmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_n \\ \gamma_1\alpha_1 & \gamma_2\alpha_2 & \cdots & \gamma_n\alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1\alpha_1^{t-1} & \gamma_2\alpha_2^{t-1} & \cdots & \gamma_n\alpha_n^{t-1} \end{pmatrix}$$

Structural attacks

Sidelnikov–Shestakov attack (1991) recovers private key (the α_i's and the γ_i's) from public key in polynomial time.

- Reconstruct codewords of weight $t+1$ from the rows of the systematic generator matrix of the public code (MDS code).

Structural attacks

Sidelnikov–Shestakov attack (1991) recovers private key (the α_i's and the γ_i's) from public key in polynomial time.

- Reconstruct codewords of weight $t+1$ from the rows of the systematic generator matrix of the public code (MDS code).

Fix: Berger–Loidreau (2005): add ℓ parity checks to the matrix to hide the GRS code.

- Fake parity checks decrease the dimension of the public code (no longer MDS) and thus remove codewords needed for Sidelnikov–Shestakov attack.

Structural attacks

Sidelnikov–Shestakov attack (1991) recovers private key (the α_i's and the γ_i's) from public key in polynomial time.

- Reconstruct codewords of weight $t + 1$ from the rows of the systematic generator matrix of the public code (MDS code).

~~Fix~~: Berger–Loidreau (2005): add ℓ parity checks to the matrix to hide the GRS code.

- Fake parity checks decrease the dimension of the public code (no longer MDS) and thus remove codewords needed for Sidelnikov–Shestakov attack.

Wieschebrink (2006, 2010): apply Sidelnikov–Shestakov to the square of the public code (likely to be a GRS code containing minimum-weight word of the desired form).

Subfield subcodes

- Let $q = 2^m$;
- fix n, k with $0 \leq k < n \leq q$;
- consider a linear code $\mathcal{C}$ over $\mathbb{F}_q$.

The subfield subcode $\mathcal{C}|_{\mathbb{F}_2}$ of $\mathcal{C}$ is the restriction of $\mathcal{C}$ to $\mathbb{F}_2$.

$$\mathcal{C}|_{\mathbb{F}_2} = \{(c_1, \ldots, c_n) \in \mathcal{C} \mid c_i \in \mathbb{F}_2 \text{ for } i = 1, \ldots, n\}.$$

Properties

- Dimension: $\dim(\mathcal{C}|_{\mathbb{F}_2}) \geq n - m(n - \dim \mathcal{C})$.
- Minimum distance: $d(\mathcal{C}|_{\mathbb{F}_2}) \geq d(\mathcal{C})$.

A family of GRS codes

Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{2^m}$, $h = \prod_{i=1}^{n}(x - \alpha_i)$, and g a degree-t polynomial in $\mathbb{F}_{2^m}[x]$ with $g(\alpha_i) \neq 0$.

- The words $c = (c_1, \ldots, c_n)$ in $\mathbb{F}_{2^m}^n$ with

$$\left\{ \left(\frac{fg}{h'}(\alpha_1), \ldots, \frac{fg}{h'}(\alpha_n) \right) : f \in \mathbb{F}_{2^m}[x],\ \deg(f) < n - t \right\}$$

  form a linear $[n, n-t]$ code in $\mathbb{F}_{2^m}^n$, denoted as $\Gamma_{2^m}(g) = \Gamma_{2^m}(\alpha_1, \ldots, \alpha_n, g)$.

Properties of $\Gamma_{2^m}(g)$

- Minimum distance $d(\Gamma_{2^m}(g)) \geq t + 1$.
- Use Berlekamp's algorithm for decoding up to half the minimum distance.

Goppa codes

The restriction $\Gamma_2(g)$ of $\Gamma_{2^m}(g)$ to the field $\mathbb{F}_2$ is called a Goppa code.

Properties of $\Gamma_2(g)$

- Dimension $k \geq n - mt$.
- Minimum distance $\geq t + 1$.

q-ary Goppa codes

Let q be an arbitrary prime power.

The restriction $\Gamma_q(g)$ of $\Gamma_{q^m}(g)$ to the field $\mathbb{F}_q$ is called a Goppa code.

Properties of $\Gamma_q(g)$

- Dimension $k \geq n - mt$.
- Minimum distance $\geq t + 1$.

Wild Goppa codes

Let q be an arbitrary prime power and g squarefree in $\mathbb{F}_q[x]$.

The restriction $\Gamma_q(g)$ of $\Gamma_{q^m}(g)$ to the field $\mathbb{F}_q$ is called a Goppa code.

Properties of $\Gamma_q(g^{q-1})$

- Dimension $k \geq n - mt$.
- Minimum distance $\geq qt + 1$ since $\Gamma_q(g^q) = \Gamma_q(g^{q-1})$ for squarefree g.

Goppa codes of the form $\Gamma_q(g^{q-1})$ are called wild Goppa codes.

Structural security

Many possible codes for a given parameter set m, n, k.

- Guessing the Goppa polynomial g or the support set $\{\alpha_1, \ldots, \alpha_n\}$ is made infeasible.

Structural security

Many possible codes for a given parameter set m, n, k.

- Guessing the Goppa polynomial g or the support set $\{\alpha_1, \ldots, \alpha_n\}$ is made infeasible.

Wieschebrink's version of Sidelnikov–Shestakov attack for subcodes not applicable

- square code is not GRS.

Structural security

Many possible codes for a given parameter set m, n, k.

- Guessing the Goppa polynomial g or the support set $\{\alpha_1, \ldots, \alpha_n\}$ is made infeasible.

Wieschebrink's version of Sidelnikov–Shestakov attack for subcodes not applicable

- square code is not GRS.

Faugère et al. (2010): distinguish hidden Goppa-code matrix from random matrix for high-rate Goppa codes.

- No key recovery.

Key sizes

Typical key sizes for binary Goppa codes:

- $187kB$ for 128-bit security against ISD

Typical key sizes for q-ary Goppa codes:

- 88kB for $\Gamma_{31}(g)$ (small subfield $m = 2$, secure?).
  (P., PQCrypto 2010).

Typical key sizes for wild Goppa codes:

- 88kB for $\Gamma_{31}(g^{30})$ (extra structural security "incognito")
  (Bernstein, Lange, P., SAC 2010).

Want new designs. Still following paranoid design strategy.

The idea behind algebraic-geometric codes

- Recall GRS codes:

$$\{(\gamma_1 f(\alpha_1), \ldots, \gamma_n f(\alpha_n)) : f \in \mathbb{F}_q[x], \deg f < n - t\}.$$

Aim: use bigger class of such evaluation codes.

The idea behind algebraic-geometric codes

- Recall GRS codes:

$$\{(\gamma_1 f(\alpha_1), \ldots, \gamma_n f(\alpha_n)) : f \in \mathbb{F}_q[x], \deg f < n - t\}.$$

Aim: use bigger class of such evaluation codes.

- Replace the set $\{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_q$ by a set of points $\{P_1, \ldots, P_n\}$ on an algebraic curve $\mathcal{H}$ over $\mathbb{F}_q$.

The idea behind algebraic-geometric codes

- Recall GRS codes:

$$\{(\gamma_1 f(\alpha_1), \ldots, \gamma_n f(\alpha_n)) : f \in \mathbb{F}_q[x], \deg f < n - t\}.$$

Aim: use bigger class of such evaluation codes.

- Replace the set $\{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_q$ by a set of points $\{P_1, \ldots, P_n\}$ on an algebraic curve $\mathcal{H}$ over $\mathbb{F}_q$.

- Replace the vector space generated by $\langle 1, x, x^2, \ldots, x^{n-t-1} \rangle$ by a linear space containing functions mapping rational points of $\mathcal{H}$ to $\mathbb{F}_q$.

Algebraic-geometric codes for crypto

Consider an algebraic curve $\mathcal{H}$ with

- many points and
- Riemann–Roch spaces ($=$ linear vector spaces of functions on $\mathcal{H}$) with a nice description.

Algebraic-geometric codes for crypto

Consider an algebraic curve $\mathcal{H}$ with

- many points and
- Riemann–Roch spaces ($=$ linear vector spaces of functions on $\mathcal{H}$) with a nice description.

Additionally for crypto:

- Efficient decoding needed for the evaluation code arising from $\mathcal{H}$.

Algebraic-geometric codes for crypto

Consider an algebraic curve $\mathcal{H}$ with

- many points and
- Riemann–Roch spaces (= linear vector spaces of functions on $\mathcal{H}$) with a nice description.

Additionally for crypto:

- Efficient decoding needed for the evaluation code arising from $\mathcal{H}$.

Then we can try to throw in all tricks learned before to achieve good (or even better?) performance.

Hermitian curve

- Let $q = 2^m$.

The Hermitian curve $\mathcal{H}$ is defined over $\mathbb{F}_{q^2}$ by the absolutely irreducible polynomial

$$Y^q + Y - X^{q+1}.$$

Properties:

- For each $\alpha \in \mathbb{F}_{q^2}$ there exist exactly q values $\beta \in \mathbb{F}_{q^2}$ so that $\beta^q + \beta = \alpha^{q+1}$.
- Thus there are q^3 rational points (α, β).

Functions on $\mathcal{H}$

- Let $s \geq 0$.

The polynomials $x^i y^j$ with (i, j) in the set

$$I(s) = \{(i,j) : 0 \leq i, 0 \leq j < q, \text{ and } qi + (q+1)j \leq s\}$$

generate an $\mathbb{F}_{q^2}$-linear vector space $\mathcal{L}_s$.

- Can easily evaluate polynomials in $\mathcal{L}_s$ at the points (α, β): $\alpha^i \beta^j \in \mathbb{F}_{q^2}$.

Hermitian code

- Let $q = 2^m$;
- $0 \le s < q^3$;
- $\{P_1, \ldots, P_{q^3}\}$ on $\mathcal{H}$ where $P_i = (\alpha_i, \beta_i)$ so that $\beta_i^q + \beta_i = \alpha_i^{q+1}$ in $\mathbb{F}_{q^2}$.

The Hermitian code

$$\mathcal{C}_s = \left\{(f(P_1), \ldots, f(P_{q^3})) : f \in \mathcal{L}_s\right\} \subseteq \mathbb{F}_{q^2}^{q^3}$$

- has length q^3, dimension $k = |I(s)|$,
- minimum distance $q^3 - s$,
- $C_s^\perp = C_{q^3+q^2-q-2-s}$
  (note: different formula for dim $C_s^\perp$ if $s \le q(q-1) - 2$)

More variety

Use scaling+permuting+puncturing

- Fix integers n, s with $0 \leq s < n \leq q^3$;
- an ordered set of distinct points $\{P_1, \ldots, P_n\}$ on $\mathcal{H}$ where $P_i = (\alpha_i, \beta_i)$ so that $\beta_i^q + \beta_i = \alpha_i^{q+1}$ in $\mathbb{F}_{q^2}$.
- $\gamma_1, \ldots, \gamma_n \in \mathbb{F}_q^*$ (not necessarily distinct).

The code

$$\mathcal{C}_s = \{(\gamma_1 f(P_1), \ldots, \gamma_n f(P_n)) : f \in \mathcal{L}_s\} \subseteq \mathbb{F}_{q^2}^n$$

- has length n, dimension $k = |I(s)|$,
- minimum distance $\geq n - s$.

Decoding

Many efficient list decoders for Hermitian codes available.

E.g.,

- Guruswami–Sudan (1999),
- Shokrollahi–Wasserman (1999),
- Høholdt–Nielsen (1999),
- Lee–O'Sullivan (2006),
- Cohn–Heninger (2010),
- Beelen–Brander (2010),
- Geil–Matsumoto–Ruano (2012)

correct beyond half the minimum distance.

Structural attacks on algebraic-geometric codes

Can generalize Sidelnikov–Shestakov attack to algebraic-geometric codes

- Minder (2007), Minder–Faure (2008), Pellikaan et al (2011).

Structural attacks on algebraic-geometric codes

Can generalize Sidelnikov–Shestakov attack to algebraic-geometric codes

- Minder (2007), Minder–Faure (2008), Pellikaan et al (2011).

As for GRS codes, subfields render those attacks infeasible (see also Janwa–Moreno (1996)).

Structural attacks on algebraic-geometric codes

Can generalize Sidelnikov–Shestakov attack to algebraic-geometric codes

- Minder (2007), Minder–Faure (2008), Pellikaan et al (2011).

As for GRS codes, subfields render those attacks infeasible (see also Janwa–Moreno (1996)).

Problem:

- What is the actual minimum distance of the subfield subcode?
- Need to choose parameters; need better bounds.

Recall Goppa codes

Thanks to the equality

$$\Gamma_2(g) = \Gamma_2(g^2)$$

we know that the dimension is $\geq n - mt$ and $d \geq 2t + 1$.

At a first glance

- $\dim \Gamma_2(g) \geq n - mt$,
- $d(\Gamma_2(g)) \geq t + 1$.

and

- $\dim \Gamma_2(g^2) \geq n - m \cdot (2t) = n - 2mt$,
- $d(\Gamma_2(g^2)) \geq 2t + 1$.

Tighten parameters

Use Stichtenoth bound on the dimension of subfield subcodes (Stichtenoth, 1990):

- Let $q = 2^m$, $n = q^3$, and $\mathcal{C} = \mathcal{C}_{2s}^{\perp} = \mathcal{C}_{q^3+q^2-q-2-2s}$ where s is an integer so that $0 \leq 2s < q^3$.

Then

$$\dim \mathcal{C}|_{\mathbb{F}_2} \geq q^3 - 1 - m\left(|I(2s)| - |I(s)|\right).$$

Compare to trivial bound:

$$\begin{aligned}\dim \mathcal{C}|_{\mathbb{F}_2} &\geq q^3 - m(q^3 - \dim \mathcal{C}) \\ &\overset{(*)}{=} q^3 - m\left(q^3 - |I(q^3 + q^2 - q - 2 - 2s)|\right).\end{aligned}$$

$(*)$ if $s > \frac{1}{2}q(q-1) - 1$

Proposal for AG McEliece

Code-based encryption scheme uses $\mathcal{C} = (\mathcal{C}_{2s}^{\perp})|_{\mathbb{F}_2}$ as secret key.

- Apply usual defenses (scaling+permuting+puncturing).
- With Stichtenoth's bound we have a much better understanding of the code parameters of $\mathcal{C}|_{\mathbb{F}_2}$.
- Decode using your favorite Hermitian decoder.

For 128-bit security:

- $\mathcal{C} = (\mathcal{C}_{2s}^{\perp})|_{\mathbb{F}_2}$ where $q = 16$, $\mathcal{C}_{2s} \subset \mathbb{F}_{256}^{n}$, and $n < 4096$.

Ongoing work

- Speed up Hermitian decoding algorithms.
- Provide concrete instances (parameter optimization).
- Use other curve families: codes from C_{ab} curves (lower genus) which make use of very similar decoders.

Thank you for your attention!