Roth-Ruckenstein Factorization Algorithm

1 Basic Algorithm

Algorithm 3 determines the coefficients of the $f(x)$ polynomials in an iterative way, by computing the roots of univariate polynomials in F_q. Suppose that

$$f(x) = \phi_0 + \phi_1 x + \cdots + \phi_{k-1} x^{k-1} \tag{1}$$

is a polynomial such that $(y - f(x)) | Q(x, y)$, then the coefficients $\phi_0, \phi_1, \ldots, \phi_{k-1}$ can be determined one by one.

Algorithm 1 Roth-Ruckenstein factorization algorithm

Input: Bivariate Polynomial $Q(x, y) \in F_q[x, y]$, integer values k and i.
Output: Polynomials $f(x)$ such that $(y - f(x)) | Q(x, y)$ with $degr(f) < k$.

{Initial call is done with $Q(x, y) \neq 0$, $k > 0$ and $i = 0$}

$(\phi_0, \phi_1, \ldots, \phi_{k-1})$ {Global variable}

$M(x, y) \leftarrow \langle\langle Q(x, y) \rangle\rangle$
{Lemma 1 and Corollary 1}
Search for all roots of $M(0, y)$ in F_q.
for all γ root of $M(0, y)$ **do**
  $\phi_i \leftarrow \gamma$
  if $i = k - 1$ **then**
    return $\phi_0 + \phi_1 x + \cdots + \phi_{k-1} x^{k-1}$
  else
    RothRuckenstein($M(x, xy + \gamma)$, k, $i + 1$)
  end if
end for

We define the mapping $\langle\langle\ \rangle\rangle$,

$$\begin{array}{rccl} \langle\langle\ \rangle\rangle : & F_q[x, y] & \rightarrow & F_q[x, y] \\ & Q(x, y) & \mapsto & \langle\langle Q(x, y) \rangle\rangle = \frac{Q(x,y)}{x^m}, \end{array}$$

where m is the largest integer such that $Q(x, y)/x^m$ divides $Q(x, y)$.

We define the polynomial sequence $i \geq 1$, where initially $f_0(x) = f(x)$ y $Q_0(x, y) = \langle\langle Q(x, y) \rangle\rangle$,

$$f_i(x) = \big(f_{i-1}(x) - f_{i-1}(0)\big)/x = \phi_i + \cdots + \phi_{k-1} x^{k-1-j} \tag{2}$$

$$\widetilde{M}_i(x, y) = Q_{i-1}(x, xy + \phi_{i-1}) \tag{3}$$

$$Q_i(x, y) = \langle\langle \widetilde{M}_i(x, y) \rangle\rangle. \tag{4}$$

The reason to normalize is to avoid $Q_i(0, y)$ be the all-zero polynomial

Theorem 1 *Given $f(x) \in F_q[x]$ with $degr(f) < k$ y $Q(x,y) \in F_q[x,y]$, then from the definitions* (2) – (4),

$$(y - f(x)) | Q(x,y) \text{ if and only if } (y - f_i(x)) | Q_i(x,y), \forall i \geq 1$$

Determine ϕ_0 of $f(x) = \phi_0 + \phi_1 x + \cdots + \phi_{k-1} x^{k-1}$

> **Lemma 1** *If $(y - f(x)) | Q(x,y)$, then $y = f(0) = \phi_0$ is a root of the equation*
>
> $$Q_0(0,y) = 0,$$
>
> *where $Q_0 = \langle\langle Q(x,y) \rangle\rangle$.*

Coefficients of $f(x) = \phi_0 + \phi_1 x + \cdots + \phi_{k-1} x^{k-1}$

> **Corollary 1** *If $(y - f(x)) | Q(x,y)$, then the coefficient ϕ_i is a root of*
>
> $$Q_i(0,y),$$
>
> *for $i = 1, 2, \ldots, k-1$.*

2 Stopping Rule

Stopping Rule

> **Corollary 2** *If $y | Q_k(x,y)$, i.e., if $Q_k(x,0) = 0$ then $f(x) = \phi_0 + \phi_1 x + \cdots + \phi_{k-1} x^{k-1}$ is a y-root of $Q(x,y)$.*

Algorithm 2 Roth-Ruckenstein factorization algorithm

Input: Bivariate Polynomial $Q(x,y) \in F_q[x,y]$, integer values k and i.
Output: Polynomials $f(x)$ such that $(y - f(x)) | Q(x,y)$ with $degr(f) < k$.

```
{Initial call is done with Q(x,y) ≠ 0, k > 0 and i = 0}

(φ_0, φ_1, ..., φ_{k-1}) {Global variable}

M(x,y) ← ⟨⟨Q(x,y)⟩⟩
{Lemma 1 and Corollary 1}
Search for all roots of M(0,y) in F_q.
for all γ root of M(0,y) do
  φ_i ← γ
  if i = k − 1 then
    {Corollary 2}
    M(x,y) < −⟨⟨M(x, xy + γ)⟩⟩
    if M(x,0) = 0 then
      return  φ_0 + φ_1 x + ··· + φ_{k-1} x^{k-1}
    end if
  else
    RothRuckenstein(M(x, xy + γ), k, i + 1)
  end if
end for
```

3 Roth-Ruckenstein Magic

Since if

$$(y - (\phi_0 + \phi_1 x + \phi_2 x^2 + \cdots))|(Q(x, y)$$

then the first coefficient ϕ_0 is a root of $\langle\langle Q(0, y)\rangle\rangle$

Now how do we find ϕ_1?

Magic: Replace y by $xy + \phi_0$!!!

$$(xy + \phi_0 - (\phi_0 + \phi_1 x + \phi_2 x^2 + \cdots))|(Q(x, xy + \phi_0)$$

$$x(y - (\phi_1 + \phi_2 x + \phi_3 x^2 + \cdots))|(Q(x, xy + \phi_0)$$

$$(y - (\phi_1 + \phi_2 x + \phi_3 x^2 + \cdots))|\langle\langle(Q(x, xy + \phi_0)\rangle\rangle = Q_1(x, y)$$

We have "**killed**" ϕ_0 and now we are at the same situation!!. And can repeat the same procedure to pick ϕ_1, by computing the roots of

$$Q_1(x, y) = \langle\langle(Q(x, xy + \phi_0)\rangle\rangle$$

and so on. This was stated in Theorem 1.

4 Code

```
public static Vector<GFPolynomial> reconstruir(GFBiPolynomial Q, int k)
                 throws KoetterVardyException {

        GaloisField.Element[] coefs = new GaloisField.Element[k];
        Vector<GFPolynomial> factors = new Vector<GFPolynomial>();

        _factor(Q, k, 0, coefs, factors);

        return factors;
 }
```

```
private static void _factor(GFBiPolynomial Q, int k, int i,
                 GaloisField.Element[] coefs, Vector<GFPolynomial> factors){

    GFBiPolynomial M, M_shift, M_change_xy, xBiPoly, xyBiPoly;
    int  numRoots;
    GaloisField field;
    GaloisField.Element gamma;
    GFPolynomial zeroPoly;

    //  M(x,y) = <<Q(x,y)>> <- Q(x,y)/x^r

    //  find all the roots in F of the univariate polynomial M(0,y);
    Vector<GaloisField.Element> roots = (M.xEval(field.zeroElement())).roots();

    //  for each of the distinct roots gamma of M(0,y) do {
    for (int j = 0; j < numRoots; j++) {
        // coefs[i] = gamma;

        // if i = k-1 and satisfies Corollary 12
        // then output coefs[0],..., coefs[k-1];
        if (i == k - 1) {

            // McElice Corollary 12 pag 35. Exit condition
            // M(x, y) <- <<M(x, xy+gamma)>>

            GFPolynomial eval = M.yEval(field.zeroElement());
```

```
                if(M.yEval(field.zeroElement()).equals(zeroPoly)){
                    factors.addElement(new GFPolynomial(coefs, field));
                }
            // Pick up remaining coeficients
            } else {
                // M_{i+1}(x, y) <- M(x, xy+gamma)
                // Reconstruct(M~(x,y), k, i+1);
                _factor(M_i+1, k, i + 1, coefs, factors);
            }
        }
    }
```

Algorithm 3 Roth-Ruckenstein factorization algorithm

Input: Bivariate Polynomial $Q(x, y) \in F_q[x, y]$, integer values k and i.
Output: Polynomials $f(x)$ such that $(y - f(x))|Q(x, y)$ with $degr(f) < k$.

{Initial call is done with $Q(x, y) \neq 0$, $k > 0$ and $i = 0$}

$(\phi_0, \phi_1, \ldots, \phi_{k-1})$ {Global variable}

$r \leftarrow \max\{r' : Q(x, y)/x^{r'} \in F_q[x, y]\}$
$M(x, y) \leftarrow Q(x, y)/x^r$
Search for all roots of $M(0, y)$ in F_q.
for all γ root of $M(0, y)$ **do**
  $\phi_i \leftarrow \gamma$
  if $i = k - 1$ **then**
    return $\phi_0 + \phi_1 x + \cdots + \phi_{k-1} x^{k-1}$
  else
    $\widehat{M}(x, y) \leftarrow M(x, y + \gamma)$
    $\widetilde{M}(x, y) \leftarrow \widehat{M}(x, xy)$
    RothRuckenstein($\widetilde{M}(x, y)$, k, $i + 1$)
  end if
end for