Cryptanalysis of the Niederreiter Public Key Scheme Based on GRS Subcodes

Christian Wieschebrink

Federal Office for Information Security (BSI),
Godesberger Allee 185-189, 53175 Bonn, Germany
christian.wieschebrink@bsi.bund.de

Abstract. In this paper a new structural attack on the McEliece/Niederreiter public key cryptosystem based on subcodes of generalized Reed-Solomon codes proposed by Berger and Loidreau is described. It allows the reconstruction of the private key for almost all practical parameter choices in polynomial time with high probability.
Keywords. Public key cryptography, McEliece encryption, Niederreiter encryption, error-correcting codes, generalized Reed-Solomon codes, Sidelnikov-Shestakov attack

1 Introduction

Public key cryptosystems based on the difficulty of the (syndrome) decoding problem for linear codes have been discussed since the work by McEliece [1] in 1977. Although the McEliece cryptosystem remains unbroken till today (for suitable parameter choices) in practice it could not stand up to encryption schemes such as RSA or schemes based on the discrete logarithm problem. This is partly due to the large (public) key sizes needed in the McEliece scheme. For example in terms of security a RSA public key of size 1024 bit is comparable to a 69 kB key in the McEliece scheme [2].

In order to reduce key sizes several alternative approaches for code based cryptography were proposed. In most of these approaches the Goppa code which is used in the McEliece cryptosystem is replaced by other codes which allow polynomial-time bounded distance decoding such as Reed-Muller codes, Gabidulin codes or generalized Reed-Solomon codes. However most of these basic variants turn out to be insecure [3–5].

Recently Berger and Loidreau presented a public key scheme based on subcodes of generalized Reed-Solomon codes [6]. It was partially cryptanalyzed in [7] where it was shown that the secret key can be recovered in manageable time if the subcode is chosen too large. However the attack quickly becomes infeasible for smaller subcodes. In the present paper we describe a new structural attack on the Berger-Loidreau scheme which works for almost all practical parameter choices. It is shown that even if relatively few (linear independent) codewords of a generalized Reed-Solomon code are given the complete code can be recovered with high probability which allows the reconstruction of the secret code parameters.

In the subsequent sections 2 and 3 we introduce some basic properties of generalized Reed-Solomon codes and the cryptosystems using those. In section 4 we review some known attacks and in section 5 we continue with the description of the new attack. A (preliminary) experimental analysis is given in section 6.

2 Basic facts about generalized Reed-Solomon codes

Let $\mathbb{F}$ be a finite field with q elements. We will always work with a field of characteristic 2, i.e. $q = 2^e$. For a matrix M let $<M>$ denote the linear code generated by the rows of M. Let $k, n \in \mathbb{N}$, $k \leq n$, $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}^n, x = (x_1, \ldots, x_n) \in (\mathbb{F} \backslash \{0\})^n$, where the α_i are pairwise distinct. The *generalized Reed-Solomon code (or GRS code)* $GRS_{n,k}(\alpha, x)$ is a linear code of length n and dimension k over $\mathbb{F}$ given by the generator matrix

$$G_{\alpha,x} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ x_1\alpha_1 & x_2\alpha_2 & \ldots & x_n\alpha_n \\ \vdots & & \ddots & \\ x_1\alpha_1^{k-1} & x_2\alpha_2^{k-1} & \cdots & x_n\alpha_n^{k-1} \end{pmatrix}, \tag{1}$$

i.e. $GRS_{n,k}(\alpha, x) = <G_{\alpha,x}>$. Typically we will assume that $GRS_{n,k}(\alpha, x)$ has full length, i.e. $n = q$. It is easy to see that $GRS_{n,k}(x, \alpha)$ consists exactly of those codewords $c \in \mathbb{F}^n$ for which a (unique) polynomial $f_c \in \mathbb{F}[x]$ of degree at most $k-1$ exists such that

$$c = (x_1 f_c(\alpha_1), x_2 f_c(\alpha_2), \ldots, x_n f_c(\alpha_n)).$$

We call f_c the *polynomial associated to* c. GRS codes allow efficient error correction. Up to $\lfloor \frac{n-k}{2} \rfloor$ errors can be corrected using the Berlekamp-Welch algorithm [8]. By applying so called list-decoding techniques even up to $n - \sqrt{(k-1)n}$ errors can be corrected [9] in polynomial time.

A useful fact about GRS codes is stated in the following

Proposition 1. *Let α, x be defined as above. Then*

$$GRS_{n,k}(\alpha, x) = GRS_{n,k}((a\alpha_1 + b, \ldots, a\alpha_n + b), (cx_1, \ldots, cx_n))$$

for all $a, b, c \in \mathbb{F}$, $a, c \neq 0$.

A proof can be found in [10]. It follows for example that α_1 and α_2 can be fixed to arbitrary distinct values in $\mathbb{F}$.

The dual of a GRS code is also a GRS code:

Proposition 2. *Let α, x be defined as above and $u := (u_1, \ldots u_n)$ where $u_i := x_i^{-1} \prod_{j \neq i} (\alpha_i - \alpha_j)^{-1}$. Then the dual code of $GRS_{n,k}(\alpha, x)$ is given by*

$$GRS_{n,k}(\alpha, x)^{\perp} = GRS_{n,n-k}(\alpha, u) \ .$$

Proof. See [10]. □

3 Cryptosystems based on GRS codes

Niederreiter was the first to suggest a public-key scheme based on GRS codes [11]. It can be described as follows:

Key generation. Given n, k ($k < n$) randomly choose α, x with above properties and let $G_{\alpha,x}$ be the generator matrix (1) of the corresponding GRS code. Furthermore choose a random nonsingular $k \times k$-matrix H over $\mathbb{F}$ and compute $M := H \cdot G_{\alpha,x}$. Let $t := \lfloor \frac{n-k}{2} \rfloor$. The public key ist given by (M, t), the private key by (α, x).

Encryption. Suppose Alice wants to send a message $b \in \mathbb{F}^k$ to Bob using his public key (M, t). Therefore she chooses a random $e \in \mathbb{F}^n$ with Hamming weight at most t and computes the ciphertext $v := b \cdot M + e$.

Decryption. Using α, x Bob applies the Berlekamp-Welch algorithm to the received ciphertext v obtaining $b' := b \cdot M$. Let M^{-1} be a right side inverse on M. The plaintext is given by $b = b' \cdot M^{-1}$.

As we will see below the Niederreiter scheme is insecure due to the Sidelnikov-Shestakov attack.

The Berger-Loidreau cryptosystem [6] is a variant of the Niederreiter scheme which resists the Sidelnikov-Shestakov attack:

Key generation. Let n, k, α, x and $G_{\alpha,x}$ be as above and $l \in \mathbb{N}^{\leq k}$. Now choose a random $(k-l) \times k$-matrix H over $\mathbb{F}$ of rank $k-l$ and compute $M := H \cdot G_{\alpha,x}$. Let $t := \lfloor \frac{n-k}{2} \rfloor$. The public key ist given by (M, t), the private key by (α, x).

Encryption. The plaintext $b \in \mathbb{F}^{k-l}$ is encrypted by choosing a random $e \in \mathbb{F}^n$ with Hamming weight at most t and computing the ciphertext $v := b \cdot M + e$.

Decryption. Decrpytion works the same way as in the Niederreiter scheme. The Berlekamp-Welch algorithm is applied to v giving $b' := b \cdot M$. Finally b can be calculated from b' as above.

Obviously in the Berger-Loidreau scheme the public matrix M is the generator matrix of a subcode of $GRS_{n,k}(\alpha, x)$. In [6] the example parameters $(n, k, l) = (255, 133, 4)$ are given. In this case the work factor of a decoding attack is $> 2^{100}$.

4 Existing Attacks

4.1 The Sidelnikov-Shestakov attack

The Niederreiter cryptosystem based on GRS codes was broken by Sidelnikov and Shestakov [5]. They show that the parameters α, x of the chosen GRS code can be recovered from the public key in polynomial time. The basic idea of their attack can be described as follows. Let $M = HG_{\alpha,x}$ be the public key. In a first step α is reconstructed. Compute the echelon form $E(M)$ of M:

$$E(M) = \begin{pmatrix} 1 & 0 & \cdots & 0 & b_{1,k+1} & \cdots & b_{1,n} \\ 0 & 1 & \cdots & 0 & b_{2,k+1} & \cdots & b_{2,n} \\ & & \ddots & & \vdots & & \vdots \\ 0 & \cdots & 0 & 1 & b_{k,k+1} & \cdots & b_{k,n} \end{pmatrix}$$

Consider the i-th row b_i of $E(M)$ and the associated polynomial f_{b_i}. Since the entries $b_{i,1}, \ldots, b_{i,i-1}$ and $b_{i,i+1}, \ldots, b_{i,k}$ of b_i are equal to zero and f_{b_i} has degree at most $k-1$ the polynomial must have the form

$$f_{b_i}(y) = c_{b_i} \cdot \prod_{j=1, j\neq i}^{k} (y - \alpha_j) \tag{2}$$

with $c_{b_i} \in \mathbb{F}\backslash\{0\}$. Now pick two arbitrary rows of $E(M)$, for example b_1 and b_2, and divide the entries of the first row by the corresponding entries in the second row as long as these are different from zero. Using (2) we get

$$\frac{b_{1,j}}{b_{2,j}} = \frac{x_j \cdot f_{b_1}(\alpha_j)}{x_j \cdot f_{b_2}(\alpha_j)} = \frac{c_{b_1}(\alpha_j - \alpha_2)}{c_{b_2}(\alpha_j - \alpha_1)} \tag{3}$$

for $j = k+1, \ldots, n$. By Proposition 1 we can assume that $\alpha_1 = 0$ and $\alpha_2 = 1$. Since the $\frac{b_{1,j}}{b_{2,j}}$ are known, the α_j can uniquely be reconstructed from (3), if $\frac{c_{b_1}}{c_{b_2}}$ is guessed correctly. It remains to find $\alpha_3, \ldots, \alpha_k$. Therefore we replace the row b_2 by b_i $(i = 3, \ldots, k)$ in the above equation (3) and get

$$\frac{b_{1,j}}{b_{i,j}}(\alpha_j - \alpha_1) = \frac{c_{b_1}}{c_{b_i}}(\alpha_j - \alpha_i). \tag{4}$$

Here $\frac{c_{b_1}}{c_{b_i}}$ and α_i are unknown, but by letting $j = k+1, k+2$ for example, those values can uniqely be reconstructed by solving a system of two linear equations.

In total α can be calculated using $O(k^2 n)$ arithmetic operations in $\mathbb{F}$.

Now in a second step x (and the matrix H as a byproduct) can be recovered. First find a non-trivial solution $c = (c_1, \ldots, c_{k+1})$ of the linear system

$$M' \cdot c = 0,$$

where M' is the $k \times (k+1)$-matrix consisting of the $k+1$ leftmost columns of the public key M. Let G' be the $k \times (k+1)$-matrix consisting of the $k+1$ leftmost columns of $G_{\alpha,x}$. Because of $M' = HG'$ c also solves

$$G' \cdot c = 0$$

and therefore the first $k+1$ entries $x_1, \ldots, x_{k+1}$ of x solve

$$\begin{pmatrix} c_1\alpha_1^0 & c_2\alpha_2^0 & \cdots & c_{k+1}\alpha_{k+1}^0 \\ c_1\alpha_1^1 & c_2\alpha_2^1 & \cdots & c_{k+1}\alpha_{k+1}^1 \\ \vdots & \ddots & & \vdots \\ c_1\alpha_1^{k-1} & c_2\alpha_2^{k-1} & \cdots & c_{k+1}\alpha_{k+1}^{k-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{k+1} \end{pmatrix} = 0.$$

By assuming $x_1 = 1$ the solution is uniquely determined. Now the matrix G' is completely known. Let G'' be the matrix consisting of the first k columns of G' and M'' the matrix consisting of the first k columns of M. We have $H = M''(G'')^{-1}$. Finally $G = H^{-1}M$ (and thereby the remaining x_i) can be computed. The second step can be completed with $O(k^3 + k^2 n)$ operations in $\mathbb{F}$. The attack works if $2 \leq k \leq n-2$.

The above attack proves the following

Proposition 3. *Let $2 \leq k \leq n-2$ and α, x be as above. There are at most $q(q-1)^2$ pairwise distinct vectors $\alpha' \in \mathbb{F}^n$ for which a $x' \in \mathbb{F}^n$ exists, s.t.*

$$GRS_{n,k}(\alpha, x) = GRS_{n,k}(\alpha', x'). \tag{5}$$

If α_1 and α_2 are fixed, the number of different α' for which x' with (5) exists is upper bounded by $(q-1)$.

Proof. Equation (3) shows that $\alpha_{k+1}, \ldots, \alpha_n$ are uniquely determined if $\alpha_1, \alpha_2, \frac{c_{b_1}}{c_{b_2}}$ are given (since the GRS code has minimum weight $n-k+1$ we know that $\frac{b_{1,j}}{b_{2,j}} \neq 0$). Furthermore, if $\alpha_{k+1}, \alpha_{k+2}$ are uniquely determined, so are $\alpha_3, \ldots, \alpha_k$ according to equation (4). The proof is complete by noting that there are at most $q(q-1)^2$ different choices for $(\alpha_1, \alpha_2, \frac{c_{b_1}}{c_{b_2}})$.

4.2 An Attack on the Berger-Loidreau cryptosystem

The attack on the Berger-Loidreau cryptosystem presented in [7] can be considered as an extension of the above method by Sidelnikov and Shestakov. We give a brief overview. Let $E(M) = [1_{k-l}|B] = (t_{i,j})$ be the echelon form of the public matrix M of the Berger-Loidreau scheme. Generalizing the above argument for every pair $(c,d) \in \{1, \ldots, k-l\}^2$ there exist polynomials $P_c, P_d \in \mathbb{F}[x]$ of degree $\leq l$ such that

$$\frac{t_{c,j}}{t_{d,j}} = \frac{(\alpha_j - \alpha_d)P_c(\alpha_j)}{(\alpha_j - \alpha_c)P_d(\alpha_j)} \tag{6}$$

for all $j = k-l+1, \ldots, n$ with $t_{d,j} \neq 0$. Now let $d = k-l =: m$ and define

$$\tilde{P}_c(x) := (x - \alpha_m)P_c(x), \tilde{Q}_c(x) := (x - \alpha_c)P_m(x),$$

s.t. (6) becomes

$$\frac{t_{c,j}}{t_{d,j}} = \frac{\tilde{P}_c(\alpha_j)}{\tilde{Q}_c(\alpha_j)}.$$

Suppose (for a moment) that $\alpha_{m+1}, \ldots, \alpha_{m+2l+3}$ are known. In this case the polynomials $\tilde{P}_c, \tilde{Q}_c$ can be calculated by solving a linear system. (The polynomials are uniqely determined if we assume that they are relatively prime and $\tilde{Q}_c$ is monic.) By extracting the linear factors of $\tilde{Q}_c$, where c ranges over $1, \ldots, m-1$ the $\alpha_1, \ldots, \alpha_{m-1}$ can be recovered. By appropriately permuting the columns of M and applying the just described method to the permuted matrix the remaining $\alpha_m, \alpha_{m+2l+4}, \ldots, \alpha_n$ can be found easily. Once α is found, x can easily be determined: since $<M>$ is a subcode of $GRS_{n,k}(\alpha, x)$ it follows from Proposition 2 that

$$M_{i,1}\alpha_1^j u_1 + \cdots + M_{i,n}\alpha_n^j u_n = 0$$

for all $i = 1, \ldots, k-l$ and $j = 0, \ldots, n-k-1$, where $M = (M_{i,j})$. So the values $u_1, \ldots, u_n$ can be recovered by solving a system of $(k-l)(n-k)$ linear equations. Typically $(k-l)(n-k) > n$ so $(u_1, \ldots, u_n)$ is expected to be uniqely determined if we require $u_1 = 1$. Finally x can be computed from $(u_1, \ldots, u_n)$.

However since $\alpha_{m+1}, \ldots, \alpha_{m+2l+3}$ are unknown (we can assume $\alpha_{m+1} = 0$, $\alpha_{m+2} = 1$) all $(q-2)\cdot\ldots\cdot(q-2l-2)$ possible assignments have to be checked. In total the procedure to reconstruct α can be completed with $O(m^2n + q^{2l+1}ml^3)$ arithmetic operations in $\mathbb{F}$, so is feasible if l and q are small. However if for example $q \geq 64$ and $l \geq 8$ the attack becomes infeasible in practice.

The described method can be improved by finding two codewords in $<M>$ which have many (i.e. more than $m-2$) zero entries in common positions. For details we refer to [7].

5 An improved attack on the Berger-Loidreau cryptosystem

Let $M = H \cdot G_{\alpha,x}$ be the public matrix of the Berger-Loidreau cryptosystem, which is the generator matrix of a $(k-l)$-dimensional subcode of $GRS_{n,k}(\alpha, x)$. We present an algorithm to recover the secret parameters α, x from M which is feasible even for larger l).

Let $r_1, \ldots, r_m$ be the rows of M and $f_1, \ldots, f_m$ ($m = k-l$) be the polynomials associated to those rows. For two row vectors $a, b \in \mathbb{F}^n$ we define the componentwise product $a * b \in \mathbb{F}^n$ to be

$$a * b := (a_1 \cdot b_1, a_2 \cdot b_2, \ldots, a_n \cdot b_n).$$

For our attack we distinguish two cases. First we consider the case $2k-1 \leq n-2$. Then the attack works as follows. Calculate $r_i * r_j$ for all $i, j \in \{1, \ldots, m\}, i \leq j$. Obviously $r_i * r_j$ has the form

$$r_i * r_j = (x_1^2 f_i(\alpha_1) \cdot f_j(\alpha_1), \ldots, x_n^2 f_i(\alpha_n) \cdot f_j(\alpha_n)),$$

and since $\deg f_i \cdot f_j \leq 2k-2$ the code C generated by the $r_i * r_j$ is a subcode of $GRS_{n,2k-1}(\alpha, x')$, where $x' = (x_1^2, \ldots, x_n^2)$. If $C = GRS_{n,2k-1}(\alpha, x')$ then the Sidelnikov-Shestakov attack can be applied to an generator matrix of C returning x', α. If $\mathrm{char}\ \mathbb{F} = 2$ the vector x can be computed from x' directly (by applying the inverse Frobenius operator), otherwise x can be recovered from M with the method described in section 4.2.

Otherwise if $C \neq GRS_{n,2k-1}(\alpha, x')$ we consider the attack to have failed. Since the running time of the attack of section 4.2 largely depends on $l = k-m$ at least we can apply 4.2 to a generator matrix of C if $0 < 2k-1-\dim C < l$. Note however that for not too large l the probability that C equals $GRS_{n,2k-1}(\alpha, x')$ seems to be very high (see section 6).

For typical instances of the Berger-Loidreau cryptosystem we may have the case $2k-1 > n-2$. The above attack does not work here in general since the Sidelnikov-Shestakov algorithm cannot be applied or the code generated by the $r_i * r_j$ may be equal to $\mathbb{F}^n$. However the idea of multiplying codewords componentwise can be applied to a shortened code of $<M>$.

Definition 1. *Let $C \subset \mathbb{F}^n$ be a linear code of length n and dimension k and let $d \in \mathbb{N}^{\leq k}$. The* shortened code $S_d(C)$ *consists of all codewords $(s_1, \ldots, s_{n-d}) \in \mathbb{F}^{n-d}$ such that*

$$(\underbrace{0, \ldots, 0}_{d\ times}, s_1, \ldots, s_{n-d}) \in C.$$

Given the generator matrix $G_C = [1_k|T]$ of C in echelon form (where T denotes a $k \times (n-k)$-matrix) a basis of $S_d(C)$ can easily be obtained by extracting the $n-d$ rightmost components of the last $k-d$ rows of G_C.

Now let M again be the public $(m \times n)$-matrix of the Berger-Loidreau cryptosystem and S be a generator matrix of $S_d(<M>)$. For a row $s = (s_1, \ldots s_{n-d})$ of S we have

$$(0, \ldots, 0, s_1, \ldots, s_{n-d}) \in GRS_{n,k}(\alpha, x),$$

so s can be written

$$s = (x_{d+1}f(\alpha_{d+1}), \ldots, x_n f(\alpha_n)),$$

where $f(x) \in \mathbb{F}[x]$ has the form

$$f(x) = g(x) \prod_{j=1}^{d} (x - \alpha_j)$$

with $\deg g(x) \leq k-d-1$. Letting $z := (x_{d+i} \prod_{j=1}^{d} (\alpha_{d+i} - \alpha_j))_{i=1,\ldots n-d}$ and $\alpha' := (\alpha_{d+1}, \ldots, \alpha_n)$ obviously we have

$$S_d(<M>) = <S> \subset GRS_{n-d,k-d}(\alpha', z).$$

if d can be chosen such that $d \leq m-1$ and $2(k-d)-1 \leq n-d-2$ the above algorithm can be applied to S which in case of success delivers at most $q(q-1)^2$ candidates for α' according to proposition 3 ($q-1$ candidates at most if we require $\alpha'_{n-1} = 1, \alpha'_n = 0$ for example). Let T be the set of these solutions. Once T is known the remaining $\alpha_1, \ldots, \alpha_d$ can be computed with similiar methods as described in [7]. In the following an alternative approach is given.

Let $m^{(1)}, \ldots, m^{(n)}$ be the column vectors of matrix M and $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ be a permutation. Let M_π denote the matrix obtained from M by permuting the columns according to π, i.e. $M_\pi = (m^{(\pi(1))}, \ldots, m^{(\pi(n))})$. Similiarly for $y := (y_1, \ldots, y_n) \in \mathbb{F}^n$ we define $y_\pi := (y_{\pi(1)}, \ldots, y_{\pi(n)})$. Obviously we have

$$<M_\pi> \subset GRS_{n,k}(\alpha_\pi, x_\pi). \tag{7}$$

For simplicity we assume $2d \leq n-3$ (for typical instances d can be chosen this way, however the following method can easily be extended to the general case). Now let $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ be given by

$$\pi(i) = \left\{ \begin{array}{l} d+i,\ 1 \leq i \leq d \\ i-d,\ d+1 \leq i \leq 2d \\ i, \quad 2d+1 \leq i \leq n. \end{array} \right\}$$

Apply the above algorithm to M_π, i.e. compute a generator matrix S' of $S_d(<M_\pi>)$ and multiply every pair of rows of S' componentwise. Because of (7) we find a set T' of candidates for $(\alpha_{\pi(d+1)}, \ldots, \alpha_{\pi(n)})$. A solution for α can be reconstructed by finding $\sigma \in T$ and $\tau \in T'$ with $\sigma_{d+1} = 0, \sigma_{d+2} = 1$ and $\sigma_i = \tau_i$ for $i = d+1, \ldots, n-d$ and setting

$$\begin{aligned} &\alpha_i = \tau_i && \text{for } 1 \le i \le d, \\ &\alpha_i = \sigma_{i-d} && \text{for } d+1 \le i \le 2d, \\ &\alpha_i = \sigma_{i-d} = \tau_{i-d} && \text{for } 2d+1 \le i \le n. \end{aligned}$$

The above algorithm is summarized in Algorithms 1 and 2. The procedure *SidelnikovShestakovAlpha(P)* in line 13 of algorithm 1 represents the Sidelnikov-Shestakov algorithm and returns the set of all possible α for a given generator matrix P of a GRS code.

Algorithm 1 *partAlpha(d,M)*

Input: d, generator matrix M with $m = k - l$ rows s.t. $<M> \subset GRS_{n,k}(\alpha, x)$
Output: Set A of candidates for $(\alpha_{d+1}, \ldots, \alpha_n)$

1: $G = (G_{i,j}) \leftarrow$ echelon form of M
2: $S \leftarrow (G_{i,j})_{\substack{i=d+1,\ldots,m \\ j=d+1,\ldots,n}}$
3: $s_1, \ldots, s_{m-d} \leftarrow$ rows of S
4: $r \leftarrow 1$
5: **for** $i \leftarrow 1, \ldots, m-d$ **do**
6: **for** $j \leftarrow i, \ldots, m-d$ **do**
7: $p_r \leftarrow s_i * s_j$
8: $r \leftarrow r + 1$
9: **end for**
10: **end for**
11: $P \leftarrow$ generator matrix of the code spanned by the p_r
12: **if** dim $<P>= 2(k-d) - 1$ **then**
13: $A \leftarrow$ *SidelnikovShestakovAlpha(P)*
14: **return** A
15: **else**
16: **return** FAIL
17: **end if**

6 Analysis and Experimental Results

Let us consider the case $2k \le n-1$. The above attack is successful if the products $r_i * r_j$ generate the code $GRS_{n,2k-1}(\alpha, x')$. The number of different products is at most $\frac{m(m+1)}{2}$. If l is chosen small (as it is suggested in [6]) then $\frac{m(m+1)}{2} > 2k-1$. For randomly chosen subcodes it is expected that indeed the above GRS code is generated. In this case the algorithm takes $O(m^4 n + k^2 n + m^2(n-k)^2 n)$

Algorithm 2 Reconstruction of α and x

Input: Public generator matrix M of the Berger-Loidreau scheme with $m = k-l$ rows and n columns

Output: Parameters α, x, s.t. $<M> \subset GRS_{n,k}(\alpha, x)$

$d \leftarrow 2k - n + 2$
if $d > m - 1$ **then**
  return FAIL
end if
if $d < 0$ **then**
  $d \leftarrow 0$
end if
$A_1 \leftarrow$ *partAlpha(d,M)*
if $d = 0$ **then**
  $\alpha \leftarrow$ random element of A_1
end if
if $d > 0$ **then**
  for $i \leftarrow 1, \ldots, n$ **do**
    if $1 \leq i \leq d$ **then**
      $\pi(i) \leftarrow d + i$
    end if
    if $d + 1 \leq i \leq 2d$ **then**
      $\pi(i) \leftarrow i - d$
    end if
    if $2d + 1 \leq i \leq n$ **then**
      $\pi(i) \leftarrow i$
    end if
  end for
  $A_2 \leftarrow$ *partAlpha*(d, M_π)
  Find $(\sigma, \tau) \in A_1 \times A_2$ with $\sigma_{d+1} = 0, \sigma_{d+2} = 1, \sigma_i = \tau_i$ for $i = d + 1, \ldots, n - d$
  for $i \leftarrow 1, \ldots, n$ **do**
    if $1 \leq i \leq d$ **then**
      $\alpha_i \leftarrow \tau_i$
    end if
    if $d + 1 \leq i \leq n$ **then**
      $\alpha_i \leftarrow \sigma_{i-d}$
    end if
  end for
end if
$X \leftarrow$ solution space of the linear system

$$M_{i,1}\alpha_1^j u_1 + \cdots + M_{i,n}\alpha_n^j u_n = 0$$

for $i = 1, \ldots, m$ and $j = 0, \ldots, n - k$.
$(u_1, \ldots, u_n) \leftarrow$ random nonzero element of X
for $i \leftarrow 1, \ldots, n$ **do**
  $x_i \leftarrow (u_i \prod_{j \neq i}(\alpha_i - \alpha_j))^{-1}$
end for
return α, x

operations in $\mathbb{F}$ in the worst case, i.e. we have a polynomial running time in the code length n.

In case $2k > n-1$ the attack is sucessful if there is a d such that

$$2k - n + 1 \leq d \leq m - 1. \tag{8}$$

and the $s_i * s_j$, where s_i, s_j are the rows of the generator matrix S of the (permuted) shortened code, generate the codes $GRS_{n-d,k-d}(\alpha', x')$ respectively $GRS_{n-d,k-d}(\alpha_\pi, x_\pi)$. A necessary condition for this is

$$\frac{(m-d)(m-d+1)}{2} \geq 2(k-d) - 1. \tag{9}$$

In this case we need $O((m-d)^4 n + (k-d)^2 n + n^3 + m^2(n-k)^2 n)$ operations for the complete attack (assuming $q = n$).

The above attack was implemented in MAGMA and verified experimentally. To this end 100 random instances of the public key for four different parameter sets were created. It turned out that for all 400 created instances the private key could be reconstructed. The average times t_α and t_x to reconstruct the vectors α and x respectively are given in table 1. (Experiments were made using MAGMA V2.11-6 on a 1.83 GHz Core 2 Duo machine.) The first line in table 1 represents the suggested parameters in [6]. The results clearly show that even if the dimension m of the subcode is small the parameters of the GRS code can easily be obtained.[1] For most practical parameter sets the scheme of [6] is insecure.

Table 1. Running times of the attack

q	n	k	m	t_α (sec)	t_x (sec)
2^8	256	133	129	337	209
2^8	256	126	45	176	105
2^7	128	60	16	23	10
2^7	128	70	34	40	14

However it is not difficult to construct instances of the Berger-Loidreau scheme where the above attack fails in the sense that $GRS_{n,2k-1}(\alpha, x)$ cannot be completely generated. For example let $k < \frac{n-3}{2}$, $b, c \in \mathbb{N}$ with $1 < c < k$ and $\frac{b(b-1)}{2} < c$. Let M be an instance of the public matrix with rows m_i where the polynomials associated to the m_i have the form

$$f_i(x) = a_i(x)g(x) + r_i(x)$$

with $i = 1, \ldots, m$ $(m < k)$ for a fixed $g(x)$ where deg $g(x) = c$, deg $a_i(x) \leq k - c - 1$, deg $r_i(x) < c$ and

$$r_b(x) = r_{b+1}(x) = \ldots = r_m(x) = 0.$$

[1] Note also that small m are insecure due to possible decoding attacks.

We have

$$f_i \cdot f_j = a_i a_j g^2 + (a_i r_j + a_j r_i)g + r_i r_j.$$

Since there are at most $\frac{b(b-1)}{2}$ different $r_i r_j \neq 0$ the subspace of $\mathbb{F}[x]$ generated by the $f_i f_j$ cannot cover all possible remainders mod g, which means the $f_i f_j$ cannot generate the linear space $\mathbb{F}^{2k-2}[x]$ of all polynomials over $\mathbb{F}$ of degree at most $2k-2$ and thus the $m_i * m_j$ cannot generate $GRS_{n,2k-1}(\alpha, x')$.

7 Conclusion and Future Work

We presented an new attack on the Berger-Loidreau public key cryptosystem which allows the reconstruction of the private key and gave experimental evidence of its correctness. The presented attack is much more efficient than previously known structural attacks. It is possible to construct instances of the scheme which resist the attack however it seems doubtful that these are secure.

Finally we make some remarks on possible implications to the security of the original McEliece scheme [1]. The original McEliece cryptosystem is one of the few unbroken code-based public key schemes. It is analogous to the Niederreiter scheme presented in section 3 where the GRS code is replaced by a binary irreducible Goppa code. This type of code belongs to the class of alternant codes. More specifically let $n := 2^e$, $(\alpha_1, \ldots, \alpha_n)$ a permutation of the elements of $\mathbb{F} := GF(n)$ and $G(x) \in \mathbb{F}[x]$ an irreducible polynomial. Then the binary linear code given by

$$GF(2)^n \cap GRS_{n,k}((\alpha_1, \ldots, \alpha_n), (G(\alpha_1), \ldots, G(\alpha_n)),$$

where $k := n - \deg G$, is called binary irreducible Goppa code. So given a (scrambled) generator matrix M of such a code (which represents the public key of the McEliece scheme), M can be considered as a generator matrix of a subcode of a GRS code. However the attack of section 5 fails in this case, since typically we have for the dimension m of the Goppa code

$$m \approx n - e \cdot \deg G = 2k - n - (e-2)\deg G.$$

As $(e-2)\deg G > 0$ there is no suitable d with (8). Of course another reason is that the component-wise products of the rows of M are elements of $GF(2)^n$, so they cannot generate a non-trivial GRS code in $\mathbb{F}^n$. There seems to be no straightforward way to generalize the attack to this case, however a more detailed analysis of the attack in this respect remains part of future work.

References

1. McEliece, R.: A public-key cryptosystem based on algebraic coding theory. DSN Progress Report, Jet Prop. Lab., California Inst. Tech. **42-44** (1978) 114–116
2. van Tilborg, H.: Encyclopedia of Cryptography and Security. Springer-Verlag (2005)

3. Minder, L., Shokrollahi, A.: Cryptanalysis of the Sidelnikov cryptosystem. In: Eurocrypt 2007. Number 4515 in Lecture Notes in Computer Science, Springer-Verlag (2007)
4. Gibson, K.: The security of the Gabidulin public-key cryptosystem. In: Eurocrypt '96. Lecutre Notes in Computer Science (2005) 212–223
5. Sidelnikov, V., Shestakov, S.: On insecurity of cryptosystems based on generalized Reed-Solomon codes. Discrete Math. Appl. **2**(4) (1992) 439–444
6. Berger, T., Loidreau, P.: How to mask the structure of codes for a cryptographic use. Designs, Codes and Cryptography **35**(1) (2005) 63–79
7. Wieschebrink, C.: An attack on a modified Niederreiter encryption scheme. In: PKC 2006. Number 3958 in Lecture Notes in Computer Science, Springer-Verlag (2006) 14–26
8. Berlekamp, E., Welch, L.: Error correction of algebraic block codes (1986) US Patent No. 4,633,470.
9. Guruswami, V., Sudan, M.: Improved decoding of Reed-Solomon and algebraic-geometric codes. In: Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on. (1998) 28–37
10. MacWilliams, F., Sloane, N.: The Theory of Error-Correcting Codes. North Holland (1997)
11. Niederreiter, N.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory **15** (1986) 159–166