A Polynomial Time Attack against Algebraic Geometry Code Based Public Key Cryptosystems

Alain Couvreur, Irene Márquez–Corbella
INRIA Saclay & LIX, CNRS UMR 7161
École Polytechnique
91120 Palaiseau Cedex
Email: alain.couvreur@lix.polytechnique.fr,
irene.marquez-corbella@inria.fr

Ruud Pellikaan
Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven
Email : g.r.pellikaan@tue.nl

***Abstract*—We give a polynomial time attack on the McEliece public key cryptosystem based on algebraic geometry codes. Roughly speaking, this attacks runs in $O(n^4)$ operations in $\mathbb{F}_q$, where n denotes the code length. Compared to previous attacks, the present one allows to recover a decoding algorithm for the public key even for codes from high genus curves.**

I. Introduction

At the end of the seventies, only a couple of years after the introduction of public key cryptography, McEliece proposed an encryption scheme [13] whose security reposes on the difficulty of decoding a random code. Compared to RSA and discrete logarithm based schemes, McEliece has the advantage to resist to quantum attacks so far. In addition, its encryption and decryption are far more efficient. On the other hand, its major drawback is the huge size of the keys required to have a good security level. The original algorithm uses binary Goppa codes. In the sequel, several proposals based on other families of algebraic codes appeared in the literature. For instance, Generalized Reed–Solomon codes are proposed in [17], subcodes of them in [1] and Binary Reed–Muller codes in [23]. All of these schemes are subject to polynomial or sub-exponential time attacks [15], [24], [27].

Another attempt, suggested by Janwa and Moreno [8] was to introduce Algebraic geometry codes. Due to Faure and Minder, this scheme was broken for codes on curves of genus $g \leq 2$, [6], [14]. However, this attack has several drawbacks which makes it impossible to extend to higher genera. Indeed, their attack requires the curve to be hyperelliptic, which is non generic for genus higher than 2. Moreover, even for hyperelliptic curves, the first step of their attack consists of the computation of minimum weight codewords and such a computation is exponential in the curve's genus. Another attempt of breaking this scheme for arbitrary genus appeared in [11], [12] where the authors describe an algorithm for retrieving an equivalent representation of the code $\mathcal{C}$ from the single knowledge of the public key. Unfortunately, the efficient construction of a decoding algorithm from this code's representation is still lacking. Indeed, the obtained embedding of the curve lies in a high dimensional projective space making difficult the computation of Riemann Roch spaces.

In this article, we use another approach based on the use of the Schur product, that is the component wise product of vectors. Our attack is inspired from the the attacks developed in [3], [4]. Thanks to Schur products of codes, we are able to compute an *Error Correcting Pair* [19] in $O(n^4)$ operations in $\mathbb{F}_q$, allowing us to decrypt any encrypted message in $O(n^3)$ under the assumption that the users also use error correcting pairs. Compared to Faure and Minder's attack, ours does not require the computation of minimum weight codewords and its complexity is polynomial in the code length with no exponential contribution of the genus. This allows us to break schemes based on high genus algebraic geometry codes. It should be pointed out that our attack is neither a generic decoding attack like *Information Set Decoding*, nor a structural attack as the structure of the code is not retrieved.

This alternative attack has been implemented in Magma [2] and broke for instance a $[729, 404]$ 126–error correcting Hermitian code (with genus 36) over $\mathbb{F}_{81}$ which had 182-bits security with respect to ISD attacks. Using an Intel ® CoreTM 2 Duo 2.8 GHz, the attack ran in 21 minutes.

II. Algebraic Geometry Codes

For basic notions on algebraic curves and algebraic geometry (AG) codes, such as curves, function fields, valuations, divisors and Riemann–Roch spaces we refer the reader to [25], [26].

A. Notation

Let $\mathcal{X}$ denote a smooth projective geometrically connected curve over a finite field $\mathbb{F}_q$. The function field of $\mathcal{X}$ is denoted by $\mathbb{F}_q(\mathcal{X})$ and for all point $P \in \mathcal{X}$ the valuation at P is denoted by v_P. Given an $\mathbb{F}_q$–divisor E on $\mathcal{X}$, the corresponding Riemann Roch space is denoted by $L(E)$. Given an n–tuple $\mathcal{P} = (P_1, \ldots, P_n)$ of pairwise distinct $\mathbb{F}_q$–points of $\mathcal{X}$, we denote by $D_{\mathcal{P}}$ the divisor $D_{\mathcal{P}} := P_1 + \cdots + P_n$. For $f \in \mathbb{F}_q(\mathcal{X})$, the divisor of f is denoted by (f). Given a divisor E with support disjoint from that of $D_{\mathcal{P}}$, the code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ is defined as

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) := \{(f(P_1), \ldots, f(P_n)) \mid f \in L(E)\}.$$

Finally, from now on the dimension of a linear code $\mathcal{C}$ will be denoted by $k(\mathcal{C})$ and its minimum distance by $d(\mathcal{C})$.

B. Some classical results in algebraic geometry coding theory

Let $\mathcal{X}$, $\mathcal{P}$ and E be respectively a smooth projective geometrically connected curve over $\mathbb{F}_q$, an n–tuple of rational points of $\mathcal{X}$ and an $\mathbb{F}_q$–divisor of degree m on $\mathcal{X}$. Then, we have the following well–known statements.

Theorem 1. *If* $\deg(E) = m < n$ *then*

$$\begin{array}{rcl} k(\mathcal{C}_L(\mathcal{X},\mathcal{P},E)) & \geq & m+1-g \\ d(\mathcal{C}_L(\mathcal{X},\mathcal{P},E)) & \geq & n-m. \end{array}$$

Moreover, if $n > m > 2g-2$ *then* $\mathcal{C}_L(\mathcal{X},\mathcal{P},E)$ *has dimension* $m-g+1$.

Theorem 2. *Let* ω *be a differential form with a simple pole and residue* 1 *at* P_j *for all* $j = 1,\dots,n$. *Let* K *be the divisor of* ω. *Then* $\mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp = \mathcal{C}_L(\mathcal{X},\mathcal{P},E^\perp)$, *where* $E^\perp = D_\mathcal{P} - E + K$ *and* $\deg(E^\perp) = n-m+2g-2$.

Corollary 3. *If* $m > 2g-2$ *then*

$$\begin{array}{rcl} k(\mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp) & \geq & n-m-1+g \\ d(\mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp) & \geq & m-2g+2. \end{array}$$

Moreover, if $n > m > 2g-2$, *then* $\mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp$ *has dimension* $n-m-1+g$.

C. The McEliece encryption scheme

Let $\mathcal{F}$ be any family of linear codes with an efficient decoding algorithm. Every element of this family is represented by the triple $(\mathcal{C},\mathcal{A}_\mathcal{C},t)$ where $\mathcal{A}_\mathcal{C}$ denotes a decoding algorithm for $\mathcal{C} \in \mathcal{F}$ which corrects up to t errors.

The McEliece scheme can be summarized as follows:

Key generation: Consider any element $(\mathcal{C},\mathcal{A}_\mathcal{C},t) \in \mathcal{F}$. Let G be a non structured generator matrix of $\mathcal{C}$. Then the *public key* and the *private key* are given respectively by

$$\mathcal{K}_{pub} = (G,t) \quad \text{and} \quad \mathcal{K}_{secret} = (\mathcal{A}_\mathcal{C})\,.$$

Encryption: $\mathbf{y} = \mathbf{m}G + \mathbf{e}$ where $\mathbf{m}$ is the message and $\mathbf{e}$ is a random error vector of weight at most t.

Decryption: Using $\mathcal{K}_{secret}$, the receiver obtains $\mathbf{m}$.

D. Context of the present article

Until the end of this article, $\mathcal{X}$ denotes a smooth projective geometrically connected curve over $\mathbb{F}_q$ of genus g, $\mathcal{P} = (P_1,\dots,P_n)$ denotes an n-tuple of mutually distinct $\mathbb{F}_q$-rational points of $\mathcal{X}$, $D_\mathcal{P}$ denotes the divisor $D_\mathcal{P} := P_1 + \cdots + P_n$ and E denotes an $\mathbb{F}_q$–divisor of degree $m \in \mathbb{Z}$ with $m > 3g-1$ (see Remark 1 further) and support disjoint from that of $D_\mathcal{P}$.

We assume that our public key is a generator matrix $\mathbf{G}$ of the public code $\mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp$ and the largest number t of errors introduced during the encryption step.

We take $t = \lfloor (d^*-g-1)/2 \rfloor$ where $d^* = m-2g+2$ is called the *designed minimum distance* of the public code $\mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp$. This correction capability seems reasonable if the secret key of the scheme is a decoding algorithm based on the so-called *error correcting pairs* (ECP). However, this value is smaller than the actual error-correction capability of $\mathcal{C}$ which is defined as $\lfloor (d^*-1)/2 \rfloor$. This case will be considered in a longer version of this article.

Thus,

$$\mathcal{C}_{pub} \quad : \quad \mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp \text{ with } t = \left\lfloor \frac{d^*-g-1}{2} \right\rfloor.$$

Our attack will consist in the computation of an ECP in order to decode $\mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp$. The following section is devoted to the theory of error correcting pairs.

Remark 1. The lower bound $m > 3g-1$ is chosen in order to have $t > 0$.

III. Decoding by Error Correcting Pairs

Given two elements $\mathbf{a}$ and $\mathbf{b}$ in $\mathbb{F}_q^n$, the *Schur product* is defined by coordinatewise multiplication, that is

$$\mathbf{a} * \mathbf{b} = (a_1b_1,\dots,a_nb_n)$$

while the *standard inner product* is defined by $\mathbf{a}\cdot\mathbf{b} = \sum_{i=1}^n a_ib_i$. In general, for two subsets A and B of $\mathbb{F}_q^n$ the set $A * B$ is given by

$$A * B := \langle\ \mathbf{a} * \mathbf{b} \mid \mathbf{a} \in A \text{ and } \mathbf{b} \in B\ \rangle\,.$$

For $B = A$, then $A * A$ is denoted as $A^{(2)}$. Furthermore, we denote by $A \perp B$ if $\mathbf{a}\cdot\mathbf{b} = 0$ for all $\mathbf{a} \in A$ and $\mathbf{b} \in B$.

Definition 1. Let $\mathcal{C}$ be a linear code in $\mathbb{F}_q^n$. A pair (A,B) of linear codes over $\mathbb{F}_q$ of length n is called a *t-error correcting pair* (ECP) for $\mathcal{C}$ if the following properties hold:

E.1 $(A*B) \perp \mathcal{C}$,
E.2 $k(A) > t$,
E.3 $d(B^\perp) > t$,
E.4 $d(A) + d(\mathcal{C}) > n$.

The notion of error correcting pair for a linear code was introduced by Pellikaan [18], [19] and independently by Kötter [10]. It is shown that a linear code in $\mathbb{F}_q^n$ with a t-error correcting pair has a decoding algorithm which corrects up to t errors with complexity $\mathcal{O}\left(n^3\right)$.

The existence of ECP's for GRS and AG codes was shown in [18], [19]. For many cyclic codes Duursma and Kötter in [5], [10] have found ECP's which correct beyond the designed BCH capacity.

The Schur product is also used for cryptanalytic applications [3], [4], [11], [27], multiparty computation, secret sharing, oblivious transfer or construction of lattices. See [22, §4] for a summary of these applications.

A. ECP for AG codes

Theorem 4 ([19, Theorem 3.3])**.** *In the context of §II-D, the pair of codes* (A,B) *defined by*

$$A = \mathcal{C}_L(\mathcal{X},\mathcal{P},F) \quad \textit{and} \quad B = \mathcal{C}_L(\mathcal{X},\mathcal{P},E-F)$$

with $m > \deg(F) \geq t+g$ *is a t-ECP for* $\mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp$. *Such a pair* (A,B) *for* $\mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp$ *always exists whenever* $m > 2g-2$.

Corollary 5. *Let us define* $A_0 = (B * \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp)^\perp$. *Then* (A_0, B) *is a* t*-ECP for* $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$.

Remark 2. The above corollary is central to our attack. It asserts that, it is sufficient to compute a generator matrix of a code of the type $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ for some divisor F with $\deg(F) \geq t + g$, in order to determine a t-ECP for the code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$.

IV. The product of the spaces $L(G)$ and $L(H)$ in $L(G + H)$

Proposition 6. *Let* G, H *be two divisors on* $\mathcal{X}$ *such that* $\deg(G) \geq 2g$ *and* $\deg(H) \geq 2g + 1$. *Then*

$$\langle\ gh \mid g \in L(G),\ h \in L(H)\ \rangle = L(G + H).$$

Proof: See [16, Theorem 6]. ∎

Corollary 7. *Let* G, H *be two divisors on* $\mathcal{X}$ *such that* $\deg(G) \geq 2g$ *and* $\deg(H) \geq 2g + 1$. *Then,*

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, G) * \mathcal{C}_L(\mathcal{X}, \mathcal{P}, H) = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G + H).$$

From the single knowledge of a generator matrix of $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$, one can compute $\deg(F) = m$ and the genus g of $\mathcal{X}$ using the following statement.

Proposition 8 ([12, Proposition 18])**.** *If* $2g+1 \leq m < \frac{n}{2}$. *Let* k_1 *and* k_2 *be the dimensions of* $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ *and* $\mathcal{C}^{(2)}$ *respectively. Then,* $m = k_2 - k_1$ *and* $g = k_2 - 2k_1 + 1$.

V. The P–Filtration

Let P be a point of the n-tuple $\mathcal{P}$. We focus on the sequence of codes

$$\mathcal{B}_i := (\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - iP))_{i \in \mathbb{N}}.$$

This sequence provides a filtration of $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$. The first step of our attack consists of the computation of some elements of this filtration.

Remark 3. Notice that for $i > 0$, the codes $\mathcal{B}_i$ are degenerated.

A. Which elements of the sequence do we know?

From a generator matrix of $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$, one can compute $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ by Gaussian elimination. Then, $\mathcal{B}_0$ is nothing but the code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ and $\mathcal{B}_1$ is the set of codewords of the code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$, which are zero at position P which can also be computed by Gaussian elimination. Thus, from now on, we assume that $\mathcal{B}_0$ and $\mathcal{B}_1$ are known.

B. Effective computations

The only information available to the attacker is exactly a generator matrix of $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$ and its error correcting capability t.

From Remark 2, attacking the scheme reduces to compute a generator matrix of a code of the form $B = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ for some F of degree $t + g$ and disjoint support from $\mathcal{P}$.

In this section we present a polynomial time method to compute a generator matrix of B. Then, a t-ECP (A, B) can be deduced from B and the public code using Corollary 5.

Definition 2. Let G be a divisor and P be a rational point on the curve $\mathcal{X}$. An integer $\gamma \geq -\deg(G)$ is called a G gap at P if $L(G + \gamma P) = L(G + (\gamma - 1)P)$.

Theorem 9. *If* $s \geq 1$ *and* $\frac{n}{2} > m \geq 2g + s + 1$, *then* $\mathcal{B}_{s+1}$ *is the solution space of the following problem*

$$\mathbf{z} \in \mathcal{B}_s \quad \textit{and} \quad \mathbf{z} * \mathcal{B}_{s-1} \subseteq (\mathcal{B}_s)^{(2)}. \tag{1}$$

Proof: From Corollary 7, every $\mathbf{z} \in \mathcal{B}_{s+1}$ satisfies (1). Conversely, assume the existence of $\mathbf{c} \in \mathcal{B}_s \setminus \mathcal{B}_{s+1}$ satisfying (1). Then there exists $f \in L(E - sP) \setminus L(E - (s+1)P)$ such that $(f(P_1), \ldots, f(P_n)) = \mathbf{c}$, i.e. $\mathrm{v}_P(f) = s$. From Riemann-Roch Theorem, if $\deg(E) - s \geq 2g + 1$, then any integer $\gamma \geq -s$ is an E non-gap at P. Thus, there exists $g \in L(E - (s-1)P) \setminus L(E - sP)$, i.e. such that $\mathrm{v}_P(g) = s - 1$. Since $\mathbf{c}$ satisfies (1), we have $\mathrm{ev}_{\mathcal{P}}(fg) \in \mathcal{B}_s^{(2)} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, 2E - 2sP)$. Moreover, since $m < \frac{n}{2}$, then the evaluation map from $L(2E - 2sP)$ to $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, 2E - 2sP)$ is injective. Thus, $fg \in L(2E - 2sP)$ but $\mathrm{v}_P(fg) = 2s - 1 < 2s$, which yields a contradiction. ∎

This result gives rise to Algorithm 1 for determining the code $\mathcal{B}_{t+g}$, which consists of $(t + g)$ repeated applications of Theorem 9.

Algorithm 1: Let $\frac{n}{2} > m \geq 3g + t + 1$

Data: Generator matrices for the codes $\mathcal{B}_0$ and $\mathcal{B}_1$
Result: A generator matrix for the code $\mathcal{B}_{t+g}$
for $s = 2, \ldots, t + g$
  Compute $\mathcal{B}_s$ from the codes $\mathcal{B}_{s-1}$ and $\mathcal{B}_{s-2}$ using Theorem 9.
endfor

Algorithm complexity: We solve $(t + g)$ systems of linear equations of type (1).

Actually, we can do better by decreasing the number of iterations of the above algorithm and relaxing the parameters conditions. The following theorem yields to a nice improvement giving rise to Algorithm 2. We omit its proof which is very similar to that of Theorem 9.

Theorem 10. *If* $\frac{n}{2} > m \geq 2g + \left\lfloor \frac{s+1}{2} \right\rfloor + 1$, *then* $\mathcal{B}_s$ *is the solution space of the following problem*

$$\mathbf{z} \in \mathcal{B}_{\lfloor (s+1)/2 \rfloor} \quad \textit{and} \quad \mathbf{z} * \mathcal{B}_0 \subseteq \mathcal{B}_{\lfloor s/2 \rfloor} * \mathcal{B}_{\lfloor (s+1)/2 \rfloor}. \tag{2}$$

C. Extending the attack

We have been working under the assumption that $m < \frac{n}{2}$. In the remainder of this section we will see how by shortening arguments this condition can be weakened.

Definition 3. Consider the code $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$. Let I be a subset of $\{1, \ldots, n\}$ and E_I the divisor $E - \sum_{j \in I} P_j$. We define as $\mathcal{C}(I)$ the code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E_I)$.

Lemma 11. *Let* $m \geq \frac{n}{2}$. *Let* I *be a subset of* $\{1, \ldots, n\}$ *with* $|I| > 2m - n + 1$. *Then* $\deg(E_I) < \frac{n}{2}$.

Algorithm 2: Let $\frac{n}{2} > m \geq \frac{5g+t}{2} + 1$

Data: Generator matrices for the codes $\mathcal{B}_0$ and $\mathcal{B}_1$
Result: A generator matrix for the code $\mathcal{B}_{t+g}$
Let $b = \lfloor \log_2 n \rfloor + 1$, then n satisfies: $2^{b-1} \leq n < 2^b$.
Therefore, $2 \leq n/2^{b-2} < 4$.

Compute $\mathcal{B}_2$ and $\mathcal{B}_3$ using Proposition 9;
for $s = b-2, \ldots, 1$
  Compute the codes $\mathcal{B}_{\lfloor (t+g)/2^{s-1} \rfloor}$ and $\mathcal{B}_{\lfloor (t+g+1)/2^{s-1} \rfloor}$ from the codes $\mathcal{B}_{\lfloor (t+g)/2^{s} \rfloor}$ and $\mathcal{B}_{\lfloor (t+g+1)/2^{s} \rfloor}$ by applying twice Theorem 10.
endfor

Algorithm complexity: We solve $2\lceil \log_2(t+g) \rceil + 2$ systems of linear equations of type (1) and (2).

Lemma 12. *Let $I_1, \ldots, I_s$ be different subsets of $\{1, \ldots, n\}$ such that*

$$\cap_{j=1}^{s} I_j = \emptyset \quad \text{and} \quad k(\mathcal{C}) - |I_j| \geq |\cap_{i=j+1}^{s} I_i| - |\cap_{i=j}^{s} I_i|$$

Then $\mathcal{C} = \mathcal{C}(I_1) + \cdots + \mathcal{C}(I_s) = \sum_{j=1}^{s} \mathcal{C}(I_j)$.

Remark 4. Suppose $m \geq \frac{n}{2}$. Then, to compute $\mathcal{B}_{t+g}$, it suffices to find different subsets $I_1, \ldots, I_s$ of $\{2, \ldots, n\}$ with at least $i > 2m - n + 1$ elements and each satisfying the assumptions of Lemma 12. Let us use the following notation:

$$\mathcal{B}_l(I_j) := \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - l(P_1 + \sum_{i \in I_j} P_i)) \text{ with } l \in \mathbb{N}.$$

Then, Algorithm 1 or 2 will provide the codes $\mathcal{B}_{t+g}(I_j)$ with $j = 1, \ldots, s$ from which we obtain the desired code using Lemma 12.

D. From degenerate to non degenerate codes

In summary, from the single knowledge of $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$ we are able to compute a subcode $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ of $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ for some positive divisor E. Unfortunately, since E is supported by elements of $\mathcal{P}$, the code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ is degenerated and hence not suitable for the construction of an ECP using Corollary 5. In what follows, we explain how to compute another code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F')$, where F' is linearly equivalent to $F$and has disjoint support with $D_{\mathcal{P}}$. It should be pointed out that we do not need to compute h but just prove its existence.

On the following, we explain how to compute a generator matrix of $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t+g)P - (h))$ knowing generator matrices of $\mathcal{B}_{t+g}$ and $\mathcal{B}_{t+g+1}$.

Theorem 13. *Let $\mathbf{G}$ be a generator matrix of $\mathcal{B}_{t+g}$ of the form*

$$\mathbf{G} = \left(\begin{array}{c|c} 0 & \mathbf{c}_1 \\ \hline (0) & \mathbf{G}_1 \end{array} \right),$$

where $\mathbf{c}_1 \in \mathbb{F}_q^{n-1}$ and $\left(\begin{array}{c|c} 0 & \mathbf{c}_1 \end{array} \right) \in \mathcal{B}_{t+g} \setminus \mathcal{B}_{t+g+1}$ and $\left(\begin{array}{c|c} (0) & \mathbf{G}_1 \end{array} \right)$ is a generator matrix of $\mathcal{B}_{t+g+1}$. Then, there exists a rational function h on $\mathcal{X}$ such that the matrix

$$\mathbf{G}' := \left(\begin{array}{c|c} 1 & \mathbf{c}_1 \\ \hline (0) & \mathbf{G}_1 \end{array} \right)$$

is a generator matrix for $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t+g)P - (h))$.

Proof: Let P be any P_i, for simplicity take $i = 1$.

Let $f \in L(E-(t+g)P) \setminus L(E-(t+g+1)P)$ be such that $(0 \mid \mathbf{c}_1) = (f(P_1), \ldots, f(P_n))$. By definition, $v_{P_1}(f) = t+g$. From the weak approximation Theorem [25, Theorem 1.3.1], there exists a rational function $h \in \mathbb{F}_q(\mathcal{X})$ such that

(i) $\forall i \geq 2,\ h(P_i) = 1$;
(ii) $v_{P_1}(h) = -t - g$ and $hf(P_1) = 1$.

Such a function h yields the result. Details are left to the reader. ■

VI. The Attack

A. The algorithm

Recall that the attacker knows a generator matrix of the public code $\mathcal{C}_{pub} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$ and the integer t.

If $\frac{n}{2} > m$, then the attack summarizes as follows. Otherwise we have to apply techniques from §V-C.

Step 1. Determine the values g and m using Proposition 8.
Step 2. Compute $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ by Gaussian elimination.
Step 3. Compute the code $B = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t+g)P_1)$, using one of the algorithms described in §V-B.
Step 4. Deduce from B a non degenerated code $\hat{B} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t+g)P_1 - (h))$ using §V-D.
Step 5. Apply Corollary 5 to deduce an ECP from $\hat{B}$.

B. Complexity

The costly part of the attack is the computation of the code $B = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - P_1)$ such that (A_0, B) forms an t-ECP for $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$. For that purpose we can apply one of the algorithms proposed in §V-B. Take notice that computing a generator matrix of $\mathcal{C}^{(2)}$ and then apply Gaussian elimination to such matrix costs $O\left(k^2 n^2\right)$ operations in $\mathbb{F}_q$. Roughly speaking the cost of our attack is about $O\left((\lambda+1)n^4\right)$ operations in $\mathbb{F}_q$ where λ denotes the number of linear systems to solve depending on the chosen algorithm from §V-B. The "+1" in $\lambda + 1$ is due to Theorem 13.

It seems logical to chose Algorithm 2, which has a better complexity and works for a larger set of possible m. However Algorithm 1 allows to compute a sequence of codes (called *GAP-filtration*)

$$\mathcal{B}_{t+g} \subseteq \mathcal{B}_{t+g-1} \subseteq \ldots \mathcal{B}_1 \subseteq \mathcal{B}_0.$$

In a longer version of this article, we expect to provide an attack allowing the correction of up to $t = \lfloor (d^* - 1)/2 \rfloor$ errors. This attack will use the concept of error correcting arrays [9, Definition 2.1], [20] or well-behaving sequences [7].

VII. Parameters under attack

Our attack has been implemented with MAGMA [2], we summarize in the following tables the average running times for several examples of codes, obtained with an Intel ® CoreTM 2 Duo 2.8 GHz. The table includes for each code its base field size q, its length n, its dimension k and the correction capability t when using Error Correcting pairs. Moreover, the work factor $\mathbf{w}$ of and ISD attack is given. These work factors have been computed thanks to Christiane Peter's Software [21].

Example 1. The *Hermitian curve* $\mathcal{H}_r$ over $\mathbb{F}_q$ with $q = r^2$ is defined by the affine equation $Y^r + Y = X^{r+1}$. This curve has $P_\infty = (0 : 1 : 0)$ as the only point at infinity.

Take $E = mP_\infty$ and let $\mathcal{P}$ be the $n = q\sqrt{q} = r^3$ affine $\mathbb{F}_q$-rational points of the curve. Table I considers different codes of type $\mathcal{C}_L(\mathcal{H}_r, \mathcal{P}, E)^\perp$ with $n > m > 2g - 2$.

q	g	n	k	t	$\mathbf{w}$	key size	time
7^2	21	343	193	54	2^{84}	163 ko	74 s
9^2	36	729	404	126	2^{182}	833 ko	21 min
11^2	55	1331	885	168	2^{311}	2730 ko	67 min

Table I
COMPARISON WITH HERMITIAN CODES

Example 2. The *Suzuki curves* are curves $\mathcal{X}$ defined over $\mathbb{F}_q$ by the following equation $Y^q - Y = X^{q_0}(X^q - X)$ with $q = 2q_0^2 \geq 8$ and $q_0 = 2^r$ This curve has exactly q^2+1 rational places and a single place at infinity P_∞. Let $E = mP_\infty$ and $\mathcal{P}$ be the q^2 rational points of the curve. Table II considers a code of type $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$ with $n > m > 2g - 2$.

q	g	n	k	t	$\mathbf{w}$	key size	time
2^5	124	1024	647	64	2^{110}	1220 ko	30 min

Table II
COMPARISON WITH SUZUKI CODES

VIII. Conclusion

We constructed a polynomial-time algorithm which breaks the McEliece scheme based on the AG codes whenever the number of errors introduced is $2 < t \leq \lfloor (d^* - g - 1)/2 \rfloor$, that is whenever the decoding algorithm chosen by the users is based on ECP's.

It would be desirable to have an attack for $t = \lfloor (d^* - 1)/2 \rfloor$ errors which will be considered in a long version of this article using the concepts of error correcting arrays or well-behaving sequences. This algorithm runs in $O\left(n^4\right)$ operations in the base field and is based on an explicit computation of a GAP-filtration of the public code.

Acknowledgment

The authors express their deep gratitude to Jean-Pierre Tillich for inspiring discussions.

References

[1] T. Berger and P. Loidreau. How to mask the structure of codes for a cryptographic use. *Des. Codes Cryptogr.*, 35:63–79, 2005.
[2] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).
[3] A. Couvreur, P. Gaborit, V. Gauthier-Umaña, A. Otmani, and J.-P. Tillich. Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes. Presented at the conference WCC 2013, 2013. In revision, for publication in Des. Codes Cryptogr.
[4] A. Couvreur, A. Otmani, and J.-P. Tillich. Polynomial time attack on wild McEliece over quadratic extensions, 2014. Accepted in EUROCRYPT 2014.
[5] I. Duursma and R. Kötter. Error-locating pairs for cyclic codes. *IEEE Trans. Inform. Theory*, 40:1108–1121, 1994.
[6] C. Faure and L. Minder. Cryptanalysis of the McEliece cryptosystem over hyperelliptic codes. In *ACCT 2008*, pages 99–107, 2008.
[7] O. Geil, R. Matsumoto, and D. Ruano. Feng-Rao decoding of primary codes. *Finite Fields Appl.*, 23:35–52, 2013.
[8] H. Janwa and O. Moreno. McEliece public cryptosystem using algebraic-geometric codes. *Des. Codes Cryptogr.*, 8:293–307, 1996.
[9] C. Kirfel and R. Pellikaan. The minimum distance of codes in an array coming from telescopic semigroups. *IEEE Trans. Inform. Theory*, 41(6, part 1):1720–1732, 1995. Special issue on algebraic geometry codes.
[10] R. Kötter. A unified description of an error locating procedure for linear codes. In *Proceedings of Algebraic and Combinatorial Coding Theory*, pages 113–117, Voneshta Voda, 1992.
[11] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan. On the unique representation of very strong algebraic geometry codes. *Des. Codes Cryptogr.*, pages 1–16, 2012.
[12] I. Márquez-Corbella, E. Martínez-Moro, R. Pellikaan, and D. Ruano. Computational aspects of retrieving a representation of an algebraic geometry code. *J. Symb. Comput.*, In press, Corrected Proof.
[13] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Progress Report*, 42–44:114–116, 1978.
[14] L. Minder. *Cryptography based on Error Correcting Codes*. PhD thesis, EPFL, Lausanne, 2007.
[15] L. Minder and A. Shokrollahi. Cryptanalysis of the Sidelnikov cryptosystem. In *EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Comput. Sci.*, pages 347–360. Springer-Verlag Berlin Heidelberg, 2007.
[16] D. Mumford. Varieties defined by quadratic equations. In *Questions on algebraic varieties, C.I.M.E., III Ciclo, Varenna, 1969*, pages 29–100. Edizioni Cremonese, Rome, 1970.
[17] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.
[18] R. Pellikaan. On decoding linear codes by error correcting pairs. Preprint Technical University Eindhoven, 1988.
[19] R. Pellikaan. On decoding by error location and dependent sets of error positions. *Discrete Math.*, 106–107:369–381, 1992.
[20] R. Pellikaan. On the efficient decoding of algebraic-geometric codes. In *Eurocode '92 (Udine, 1992)*, volume 339 of *CISM Courses and Lectures*, pages 231–253. Springer, Vienna, 1993.
[21] C. Peters. Information-set decoding for linear codes over $\mho_q$. In *PQCrypto*, pages 81–94, 2010.
[22] H. Randriambololona. On products and powers of linear codes under componentwise multiplication. ArXiv:1312.0022, 2013.
[23] V. Sidelnikov. A public-key cryptosytem based on Reed-Muller codes. *Discrete Math. Appl.*, 4(3):191–207, 1994.
[24] V. M. Sidelnikov and S. O. Shestakov. On the insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Math. Appl.*, 2:439–444, 1992.
[25] H. Stichtenoth. *Algebraic function fields and codes*, volume 254 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, second edition, 2009.
[26] M. Tsfasman, S. Vlăduţ, and D. Nogin. *Algebraic geometric codes: basic notions*, volume 139 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2007.
[27] C. Wieschebrink. Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In *Post-Quantum Cryptography*, volume 6061 of *Lecture Notes in Comput. Sci.*, pages 61–72. Springer-Verlag Berlin Heidelberg, 2010.