Isomorphisms between Artin-Schreier Towers

Jean-Marc Couveignes
Groupe de Recherche en Mathématiques et Informatique du Mirail
Université de Toulouse II, Le Mirail

Abstract

We give a method for efficiently computing isomorphisms between towers of Artin-Schreier extensions over a finite field. We find that isomorphisms between towers of degree $p^n$ over a fixed field $\mathbb{F}_q$ can be computed, composed and inverted in time essentially linear in $p^n$. The method relies on an approximation process.

1 Introduction

Let $\mathbb{F}_q$ be a finite field with $q = p^d$ elements. Let $L_n$ be an extension of degree $p^n$ of $\mathbb{F}_q$, given as a tower

$$L_n \supset L_{n-1} \supset ... \supset L_1 \supset L_0 = \mathbb{F}_q$$

(1)

of non-trivial Artin-Schreier extensions each defined by

$$L_{k+1} = L_k(x_{k+1}) \text{ with } x_{k+1}^p - x_{k+1} - a_k = 0 \text{ and } a_k \in L_k.$$ 

We call $n$ the length of the tower.

Artin-Schreier towers naturally arise in computational algebraic geometry. In particular, let $G = \text{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_q)$ be the absolute Galois group of $\mathbb{F}_q$. Morphisms between abelian varieties $A$ and $B$ defined over $\mathbb{F}_q$ induce $G$-morphisms between the Tate modules $\mathcal{T}_\ell(A)$ and $\mathcal{T}_\ell(B)$. If $\ell \neq p$, this correspondence is known to be bijective, by a theorem of Tate [8]. If $\ell = p$, $A$ simple and $\mathcal{T}_\ell(A)$ is non-zero, then the correspondence is injective. Assume the $p$-torsion of $A$ and $B$ is defined over $\mathbb{F}_q$. One can easily show that the definition field $L_k$ of the $p^{k+1}$-torsion of $A$ is an extension of $L_0 = \mathbb{F}_q$ with degree dividing $p^k$. Similarly the definition field $M_k$ of the $p^{k+1}$-torsion of $B$ is an extension of $M_0 = L_0 = \mathbb{F}_q$ with degree dividing $p^k$. Assuming the existence of an isogeny between $A$ and $B$ with prime to $p$ degree,
the fields $L_k$ and $M_k$ are isomorphic. These fields can be constructed by taking successive preimages of a $p$-torsion point by separable isogenies of degree $p$. Thus they naturally come as Artin-Schreier towers. In the case of non-supersingular elliptic curves, such isogenies are described in terms of Hasse functions. If we are looking for an isogeny with a given prime to $p$ degree between $A$ and $B$, we can compute it by interpolation at enough $p^k$-torsion points. This reduces to computing an isomorphism between the Artin-Schreier towers we have on each side. This method is of special interest for computing the cardinality of ordinary elliptic curves with the Schoof-Elkies-Atkin algorithm. See [2] where the fastest known algorithm for this purpose is given, assuming the characteristic $p$ is fixed. Surveys on these questions are in [6, 4, 3, 5].

We shall prove the following

**Theorem 1** An isomorphism between two Artin-Schreier towers $L_n$ and $M_n$ of degree $p^n$ over $\mathbb{F}_q = L_0 = M_0$ can be computed in time $O(n^6 p^n)$ multiplications in $\mathbb{F}_q$ for fixed $q$ and $n \to \infty$.

Computational aspects of Artin-Schreier towers have already been studied by D. G. Cantor in [1]. For any integer $u$ in $[0, p^n]$ with $p$-adic expansion $u = u_1 + u_2 p + \ldots + u_n p^{n-1}$ he sets $\chi_u = x_1^{u_1} x_2^{u_2} \ldots x_n^{u_n}$. The monomials $(\chi_u)_{0 \leq u < p^n}$ form a basis $\mathcal{X}$ of the $L_0$-vector space $L_k$. If $a_0 = 1$ and $a_k = \chi_{p^k-1} + \sum_{u=0}^{p^k-2} c_u \chi_u$ with all the $c_u \in \mathbb{F}_q$, we say that the tower in formula 1 is a Cantor tower. One of the results in [1] is that for any prime $p$ there exists a constant $K_p$ such that two elements in a Cantor tower of length $n$ over $\mathbb{F}_p$ can be multiplied at the expense of $K_p n^2 p^n$ operations in $\mathbb{F}_p$. The same holds for Cantor towers over a non-necessarily prime field $\mathbb{F}_q$. We shall need this result and the corresponding algorithm. In order to compute an isomorphism between two Artin-Schreier towers, we shall first compute isomorphisms between each of the two towers and a given Cantor tower. The expected isomorphism will then be obtained as a composition of these two isomorphisms. It is the purpose of lemma 1 to state how efficiently isomorphisms between Artin-Schreier towers can be dealt with.

If $\alpha, \beta \in L_n$ we define the \textit{\'etale} $d(\alpha, \beta)$ to be the logarithm (with base $p$) of the degree of the extension $\mathbb{F}_q(\alpha - \beta)/\mathbb{F}_q$. The triangle inequality is easily checked. Note that $d$ is not a distance since $d(\alpha, \beta) = 0$ if and only if $\alpha - \beta$ is in $\mathbb{F}_q$. On the other hand, $d$ is invariant under translation.

For any two positive integers $i$ and $j$ we define the following polynomials in $\mathbb{F}_p[X]$

$$\Phi_i(X) = X^{p^i}$$
$$\varphi_i(X) = X^{p^i} - X$$
$$T_{i,j} = X + X^{p^j} + X^{p^{2j}} + \ldots + X^{p^{(i-1)j}}. $$

The polynomial $\varphi_k$ is usually called an isogeny [7]. To simplify we set $T_i = T_{i,1}$.

We have the trivial relations

2
\[ \varphi_i \circ \varphi_j = \varphi_j \circ \varphi_i \] and \( \varphi_j \circ T_{i,j} = T_{i,j} \circ \varphi_j = \varphi_{ij} \) and \( T_{j,k} \circ T_{i,j,k} = T_{i,j,k} \).

If \( K \subset L \) is an extension of finite fields with cardinalities \( p^i \) and \( p^j \) respectively, we have the following exact sequence of \( K \)-vector spaces.

\[
0 \to K \to L \xrightarrow{\psi} L \xrightarrow{T_{i,j}} K \to 0.
\]

Assume we are looking for an isomorphism \( \iota : M_n \to L_n \) between two Artin-Schreier towers \( L_n \) and \( M_n \) with \( M_n \) defined by

\[ M_n \supset M_{n-1} \supset \ldots \supset M_1 \supset M_0 = \mathbb{F}_q \]

and

\[ M_{k+1} = M_k(y_{k+1}) \text{ and } y_{k+1}^p - y_{k+1} - b_k = 0 \text{ with } b_k \in M_k. \]

We define \( \zeta_u = y_1^{u_1} y_2^{u_2} \ldots y_n^{u_n} \) similarly to \( \chi_u \). We may assume that an isomorphism has already been constructed between \( L_{n-1} \) and \( M_{n-1} \). In order to extend it, we have to solve in \( L_n \) an Artin-Schreier equation.

Consider such an equation

\[ \varphi_1(Y) = Y^p - Y = \beta. \] (2)

with \( \beta \in L_n \) and \( \text{Tr}_{L_n/\mathbb{F}_p}(\beta) = 0. \)

This is a linear equation over \( \mathbb{F}_p \). The corresponding linear system of dimension \( dp^n \) over \( \mathbb{F}_p \) can be solved with Gauss’s algorithm at the expense of \( O(d^3 p^{3n}) \) operations in \( \mathbb{F}_p \). We notice, however, that equation 2 implies

\[ \varphi_1(Y) = Y^{p^i} - Y = \beta + \beta^p + \ldots + \beta^{p^{i-1}} = T_i(\beta) \] (3)

which is linear over the intermediate field \( \mathbb{F}_{p^i} \). The corresponding linear system of dimension \( dp^n/i \) over \( \mathbb{F}_{p^i} \) can be solved with Gauss’s algorithm at the expense of \( O(d^3 p^{3n}/i^3) \) operations in \( \mathbb{F}_{p^i} \). This is better when multiplication is fast in \( L_n \) (e.g. when \( L_n \) is a Cantor tower).

Equation (3), of course, does not imply equation (2) but if we know a solution \( \gamma \) to equation (3) and set \( Y = Z + \gamma \) in equation (2) we get

\[ \varphi_1(Z) = Z^p - Z = \beta - \gamma^p + \gamma. \]

Let \( \delta = \beta - \gamma^p + \gamma \). We have \( \varphi_1(\delta) = \varphi_1(\beta) - \varphi_1(\varphi_1(\gamma)) = \varphi_1(\beta) - \varphi_1(\varphi_1(\gamma)) = \varphi_1(\beta) - \varphi_1(T_i(\beta)) = 0 \) so \( \delta \in \mathbb{F}_{p^i} \). We also check easily that \( T_i(\delta) = T_i(\beta) - \varphi_1(T_i(\gamma)) = T_i(\beta) - \varphi_1(\gamma) = 0. \) We conclude that the écart between \( \gamma \) and
any solution of 2 is at most \( \log_p(i/\gcd(d, i)) \). We say that \( \delta \) is an approximate solution to equation 2 with accuracy \( \log_p(i/\gcd(i, d)) \).

Since our strategy is to deal with the smallest possible matrices, we shall take \( i = dp^{n-1} \). This way, for \( \beta \in L_n \) and \( \text{Tr}_{L_n/\mathbb{F}_p}(\beta) = 0 \), a solution to \( Y^p - Y = \beta \) can be found in three steps

1. compute \( B = T_{dp^{n-1}}(\beta) \).

2. find a solution \( \gamma \) to \( Y^{dp^{n-1}} - Y = B \) which amounts to solving a linear system of dimension \( p \) over \( L_{n-1} \).

3. solve \( Z^p - Z = \delta \) where \( \delta = \beta - \gamma^p + \gamma \) is in \( L_{n-1} \) and \( \text{Tr}_{L_{n-1}/\mathbb{F}_p}(\delta) = 0 \).

And the same method is applied recursively to the equation in step 3. After \( k \) steps, we obtain an approximate solution to equation 2 with accuracy \( n - k \).

After \( n \) steps, we reduce to an Artin-Schreier equation over the base field \( \mathbb{F}_q \).

In the rest of this paper, we provide details and a complexity analysis for the algorithm sketched above.

## 2 Artin-Schreier towers

We recall a few elementary facts about Artin-Schreier extensions. Let \( K \) be a field of characteristic \( p \), not necessarily finite, and \( L = K[X]/(X^p - X - \alpha) \) an Artin-Schreier extension. Set \( x = X \mod X^p - X - \alpha \). Its conjugates are the \( x + c \) with \( c \in \mathbb{F}_p \). The trace is given by

\[
\text{Tr}_{L/K}\left( \sum_{0 \leq i \leq p-1} u_i x^i \right) = -u_{p-1} \text{ when } u_i \in K
\]

and the dual basis of \( (1, x, x^2, \ldots, x^{p-1}) \) is \((-x^{p-1}+1, -x^{p-2}, -x^{p-3}, \ldots, -x, -1)\).

In such an Artin-Schreier extension, \( p \)-powers are easy to compute. Indeed

\[
x^{ip^h} = (x + T_h(\alpha))^i.
\]  

(4)

In particular if \( K = \mathbb{F}_q \) with \( q = p^d \) elements then

\[
x^{iq} = (x + \text{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(\alpha))^i.
\]

and \( \text{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(\alpha) \) is in \( \mathbb{F}_p \). Thus the \( p \times p \) matrix of the Frobenius automorphism \( x \mapsto x^q \) has coefficients in \( \mathbb{F}_p \).

We shall first prove a few complexity estimates concerning basic computations with isomorphisms between Artin-Schreier towers over finite fields.

We consider an isomorphism \( \iota \) between two towers \( L_n \) and \( M_n \)

\[
\iota : M_n \to L_n
\]
The computer representation of $i$ will consist of the images of the $y^i_k$ by $i$ for $0 \leq i \leq p - 1$ and $1 \leq k \leq n$.

We shall see that this representation is very efficient. For $0 \leq k \leq n$, we denote by $C^L_{X}(k)$ the complexity of multiplication in $L_k$. This complexity is given as a number of multiplications in the base field $F_q$, disregarding additions. We denote by $C^M_{X}(k)$ the complexity of multiplication in $M_k$. Let $C_i(n)$ be the cost of evaluating $i$ at some $\mu$ in $M_n$. Let $C^*_i(n)$ be the complexity of computing $i^{-1}(\nu)$ for $\nu$ in $L_n$.

We shall first prove the following

**Lemma 1** Given an isomorphism $i : M_n \to L_n$ between two Artin-Schreier towers, we have, with the notation given above

\[
C_i(n) \leq pn C^L_{X}(n), \tag{5}
\]

\[
C^*_i(n) \leq 2np^3 C^L_{X}(n) \tag{6}
\]

\[
C^M_{X}(n) \leq 4np^3 C^L_{X}(n). \tag{7}
\]

We first prove inequality 5. For $\mu \in M_n$, let us write $\mu = \sum_{0 \leq i \leq p - 1} \mu_i y^i_n$ with $\mu_i \in M_{n-1}$. Then $i(\mu) = \sum_{i} i(\mu_i) i(y^i_n)$ and since we have stored the $i(y^i_n)$, we reduce to computing $p$ multiplications in $L_n$ and the images $i(\mu_i)$. Therefore

\[
C_i(n) \leq p(C_i(n - 1) + C^L_{X}(n))
\]

and the result follows iterating the above inequality and using the easy inequality

\[
C^L_{X}(n) \geq pC^L_{X}(n - 1).
\]

In order to compute the inverse image of $\nu \in L_n$, we first express $\nu$ as a linear combination

\[
\nu = \sum_{0 \leq i \leq p - 1} \nu_i y^i_n \tag{8}
\]

with $\nu_i \in L_{n-1}$ for all $i$. This is achieved at the expense of $2p^3$ multiplications in $L_n$ using Gauss's algorithm. From equation 8 we deduce

\[
i^{-1}(\nu) = \sum_{0 \leq i \leq p - 1} i^{-1}(\nu_i) y^i_n.
\]

We thus reduce to computing the $p$ preimages of the $\nu_i \in L_{n-1}$. Therefore

\[
C^*_i(n) \leq 2p^3 C^L_{X}(n) + pC^*_i(n - 1)
\]

and inequality 6 follows.
Inequality 7 follows easily from inequalities 5 and 6.
This shows that if we can multiply efficiently in \( L_n \), the knowledge of \( \iota \) allows fast multiplication in \( M_n \) as well.

The crucial step in our isomorphism computations will be the evaluation of polynomials \( T_{i,j} \) at numbers \( \mu \) that are not necessarily in \( \mathbb{F}_{p^j} \). Lemma 2 states how efficiently one can compute \( \Phi_{d_{df}}(\mu) = \mu^{p^k} \) and \( T_{d_{df}}(\mu) \) for \( \mu \in L_k \) and \( 0 \leq l \leq k \).

We denote by \( C_{\Phi}^L(l,k) \) the complexity of computing \( \Phi_{d_{df}}(\mu) \) for \( \mu \in L_k \). We denote by \( C_T^L(l,k) \) the complexity of computing \( T_{d_{df}}(\mu) \) for \( \mu \in L_k \).

In order to compute \( T_{d_{df}}(\mu) \) we notice that

\[
T_{d_{df}} = T_d \circ T_{p,d} \circ ... \circ T_{p,d_{df}^{-2}} \circ T_{p,d_{df}^{-1}}.
\]

Using this formula we obtain

\[
C_{T}^L(l,k) \leq p(C_{\Phi}^L(l-1,k) + C_{\Phi}^L(l-2,k) + \cdots + C_{\Phi}^L(1,k) + C_{\Phi}^L(0,k)) + pd C_{\Phi}^L(k).
\]

If we now want to compute \( \Phi_{d_{df}}(\mu) \) we use formula 4.
Writing \( \mu = \sum_{0 \leq i \leq p-1} \mu_i x_k^i \) we have

\[
\Phi_{d_{df}}(\mu) = \sum_{0 \leq i \leq p-1} \Phi_{d_{df}}(\mu_i) \Phi_{d_{df}}(x_k^i) = \sum_{0 \leq i \leq p-1} \Phi_{d_{df}}(\mu_i)(x_k + T_{d_{df}}(a_{k-1}))^i
\]

since \( x_k^p - x_k = a_{k-1} \).

We first assume that we already computed and stored the \( T_{d_{df}}(a_{\kappa}) \) and their first \( p \) powers for all \( l \) and \( \kappa \) such that \( 0 \leq l \leq \kappa < k \) which is the same as computing the expansions of polynomials \( (x + T_{d_{df}}(a_{\kappa}))^i \) for \( 0 \leq i \leq p-1 \).

We call \( C_{\Phi}^L(l,k) \) the complexity of computing \( \Phi_{d_{df}}(\mu) \) for \( \mu \in L_k \) under this assumption. We define \( \tilde{C}_{\Phi}^L(l,k) \) to be the complexity of computing \( T_{d_{df}}(\mu) \) for \( \mu \in L_k \) in the same situation.

From equation 11 we deduce

\[
\tilde{C}_{\Phi}^L(l,k) \leq p \tilde{C}_{\Phi}^L(l,k-1) + p^2 C_{\Phi}^L(k-1).
\]

Since \( C_{\Phi}^L(l,k) = 0 \) as soon as \( l \geq k \), we obtain

\[
\tilde{C}_{\Phi}^L(l,k) \leq p(k-l) C_{\Phi}^L(k).
\]

and from equation 10 and the definition of \( T_{d_{df}} \)

\[
\tilde{C}_{T}^L(l,k) \leq (p^2 kl + pd) C_{\Phi}^L(k) \leq 2p^2 kld C_{\Phi}^L(k).
\]
We now bound the cost $C^L_{\text{init}}(k)$ of precomputing all the $T_{dp'}(a_{\kappa})$ and their first $p$ powers for all $l$ and $\kappa$ such that $0 \leq l \leq \kappa < k$.

We first bound $C^L_{\text{init}}(k + 1) - C^L_{\text{init}}(k)$. Indeed if we already know the $T_{dp'}(a_{\kappa})$ and their first $p$ powers for all $0 \leq l \leq \kappa < k$, then computing the $T_{dp'}(a_{\kappa})$ for all $0 \leq l \leq k$ will require less than $2(k + 1)p^2k^2dC^L_{\chi}(k)$ multiplications (using formula 12) and computing the powers will take time $p(k + 1)C^L_{\chi}(k)$. Therefore

$$C^L_{\text{init}}(k + 1) \leq C^L_{\text{init}}(k) + (k + 1)(p + 2p^2k^2d)C^L_{\chi}(k).$$

We obtain

$$C^L_{\text{init}}(k) \leq 6p^2k^3dC^L_{\chi}(k).$$

**Lemma 2** For $0 \leq l \leq k$ and for any $\mu$ in $L_k$, one can compute $\Phi_{dp'}(\mu)$ (resp. $T_{dp'}(\mu)$) in time $\tilde{C}^L_{\Phi}(l, k)$ (resp. $\tilde{C}^L_{T}(l, k)$) with

$$\tilde{C}^L_{\Phi}(l, k) \leq p(k - l)C^L_{\chi}(k) \quad (13)$$

$$\tilde{C}^L_{T}(l, k) \leq 2p^2kldC^L_{\chi}(k) \quad (14)$$

using data that only depend on $L_k$ and can be computed once and for all in time $C^L_{\text{init}}(k)$ with

$$C^L_{\text{init}}(k) \leq 6p^2k^3dC^L_{\chi}(k). \quad (15)$$

We call $C^L_{\text{AS}}(n)$ the complexity of solving equation 2 in $L_n$ for $\beta \in L_n$ and $\text{Tr}_{L_n/F_n}(\beta) = T_{dp^n}(\beta) = 0$. We shall adopt the three steps strategy described in the introduction.

We first compute and store the $T_{dp'}(a_{\kappa})$ for all $0 \leq l \leq \kappa < n$. This takes time $C^L_{\text{init}}(n)$. We call $\tilde{C}^L_{\text{AS}}(n)$ the complexity of solving equation 2 once all this precomputation has been done.

In these conditions, step 1 (the computation of $B = T_{dp^n-1}(\beta)$) will take time $\tilde{C}^L_{T}(n - 1, n)$.

The second step reduces to computing the $p \times p$ matrix representing the $L_{n - 1}$-linear map $\phi_{dp^{n - 1}} : L_n \to L_n$ in the basis $(1, x, x^2, \ldots, x^{p - 1})$. Using Gauss’s algorithm, we then find a solution $\gamma$ to the equation $\phi_{dp^{n - 1}}(\gamma) = B$.

All this is achieved at the expense of $p\tilde{C}^L_{\Phi}(n - 1, n) + 2p^3\tilde{C}^L_{\chi}(n - 1)$ multiplications.

The third step is done in time $p\tilde{C}^L_{\chi}(n) + \tilde{C}^L_{\text{AS}}(n - 1)$.

We thus have

$$\tilde{C}^L_{\text{AS}}(n) \leq \tilde{C}^L_{\text{AS}}(n - 1) + \tilde{C}^L_{T}(n - 1, n) + p\tilde{C}^L_{\Phi}(n - 1, n) + 2p^3\tilde{C}^L_{\chi}(n - 1) + p\tilde{C}^L_{\chi}(n)$$
and using lemma 2

\[ \tilde{C}^L_{AS}(n) \leq \tilde{C}^L_{AS}(n - 1) + 6p^2 n^2 d C^L_{X}(n) \]

thus

\[ \tilde{C}^L_{AS}(n) \leq 12n^2 p^2 d C^L_{X}(n) + C_{AS} \]  \hspace{1cm} (16)

where \( C_{AS} = C^L_{AS}(0) \) is the complexity of solving an Artin-Schreier equation in the base field \( \mathbb{F}_q \).

We now want to compute an isomorphism between two Artin-Schreier towers of length \( n \) over \( \mathbb{F}_q \)

\[ L_n \supset L_{n-1} \supset \ldots \supset L_1 \supset L_0 = \mathbb{F}_q \]

and

\[ M_n \supset M_{n-1} \supset \ldots \supset M_1 \supset M_0 = \mathbb{F}_q \]

We look for an isomorphism \( \iota : M_n \to L_n \) given by \( \iota(y^i) \) for \( 0 \leq i < p \) and \( 0 \leq k \leq n \).

We let the length \( k \) increase from 0 to \( n \). We call \( C^L_{M}(k) \) the complexity of computing an isomorphism from \( M_k \) to \( L_k \). We call \( \tilde{C}^L_{M}(k) \) the complexity of computing an isomorphism from \( M_k \) to \( L_k \) assuming the \( T_{dp}(a_\kappa) \) have been computed for all \( 0 \leq l \leq \kappa < k \). We want to bound \( \tilde{C}^L_{M}(n) - \tilde{C}^L_{M}(n - 1) \). Thus assume we have computed the isomorphism up to length \( n - 1 \). In order to go further we have to solve the Artin-Schreier extension

\[ Y^p - Y = \iota(b_{n-1}) \]  \hspace{1cm} (17)

over \( L_n \). We first apply \( \iota \) to \( b_{n-1} \) in time \( C_{\iota}(n - 1) \). Solving equation 17 takes time \( \tilde{C}^L_{AS}(n) \). We take \( \iota(y_n) \) to be one of the solution we found. We then compute the powers \( \iota(y_n)^i \) for \( 0 \leq i \leq p - 1 \) which takes time \( p C^L_{X}(n) \).

We thus have

\[ \tilde{C}^L_{M}(n) \leq \tilde{C}^L_{M}(n - 1) + C_{\iota}(n - 1) + \tilde{C}^L_{AS}(n) + p C^L_{X}(n) \]

and using lemma 1 and inequality 16,}\n
\[ \tilde{C}^L_{M}(n) \leq \tilde{C}^L_{M}(n - 1) + 14n^2 p^2 d C^L_{X}(n) + C_{AS} \]

Summing up we have

\[ \tilde{C}^L_{M}(n) \leq 28n^2 p^2 d C^L_{X}(n) + n C_{AS} \]

and using 15

\[ C^L_{M}(n) \leq 34n^3 p^2 d C^L_{X}(n) + n C_{AS} \]  \hspace{1cm} (18)
Assume now we have a third Artin-Schreier tower $N_n$ over $\mathbb{F}_q$. We shall relate the complexity $C^L_X(n)$ of multiplication in $L_n$ and the complexity $C^M_M(n)$ of computing an isomorphism from $N_n$ to $M_n$. This makes sense in case $L_n$ has been designed to allow fast multiplication (e.g. $L_n$ is a Cantor Tower).

We first compute an isomorphism $\iota_1$ from $M_n$ to $L_n$ at the expense of $C^L_M(n)$ multiplications in $\mathbb{F}_q$.

We then compute an isomorphism $\iota_2$ from $N_n$ to $M_n$ at the expense of $C^M_M(n)$ multiplications in $\mathbb{F}_q$.

Using inequality 18 and inequality 7 we find

**Lemma 3** Let $L_n$, $M_n$, $N_n$ be three Artin-Schreier towers of length $n$ over $\mathbb{F}_q$ the field with $q = p^d$ elements and let $C^L_X(n)$ be the complexity of multiplication in $L_n$. Let $C_{AS}$ be the complexity of solving an Artin-Schreier equation in $\mathbb{F}_q$. An isomorphism between $M_n$ and $N_n$ can be found at the expense of $C^M_M(n)$ multiplications in $\mathbb{F}_q$ with

$$C^M_M(n) \leq 170p^5 n^4 d C^L_X(n) + 2n C_{AS}.$$

If we take $L_n$ to be a Cantor tower we have $C^L_X(n) \leq K_q n^2 p^n$ where $K_q$ only depends on $q$. Using the Berlekamp factorisation algorithm we have $C_{AS} = O(p^3 d)$ and theorem 1 follows.

**References**


