Normal Elliptic Bases and Torus-Based Cryptography. In Daniel Panario Gary McGuire, Gary L. Mullen and Igor E. Shparlinski, editors, Finite Fields: Theory and Applications, Contemporary Mathematics, pages 137-153. American Mathematical Society, 2010. Ninth International Conference on Finite Fields and Applications.

NORMAL ELLIPTIC BASES AND TORUS-BASED CRYPTOGRAPHY

CLÉMENT DUNAND AND REYNALD LERCIER

Abstract. We consider representations of algebraic tori $T_n(\mathbb{F}_q)$ over finite fields. We make use of normal elliptic bases to show that, for infinitely many squarefree integers n and infinitely many values of q, we can encode m torus elements, to a small fixed overhead and to m $\varphi(n)$-tuples of $\mathbb{F}_q$ elements, in quasi-linear time in $\log q$.

This improves upon previously known algorithms, which all have a quasi-quadratic complexity. As a result, the cost of the encoding phase is now negligible in Diffie-Hellman cryptographic schemes.

1. Introduction

Multiplicative groups defined by finite fields $\mathbb{F}_{q^n}^\times$ are of first importance in numerous applications, especially in discrete-log based public key cryptography. In this field, Diffie and Hellman's seminal paper [DH76] opened the way to their use in numerous cryptographic standards in the eighties. It turns out that elliptic curves are often prefered today, since there exist subexponential algorithms to solve the discrete logarithm problem in finite fields [Sch93]. But $\mathbb{F}_{q^n}^\times$-subgroups of order $\Phi_n(q)$, where Φ_n denotes the n-th cyclotomic polynomial (the minimal polynomial over $\mathbb{Q}$ of $e^{\frac{2i\pi}{n}}$), has reattracted attention since the publication of Lenstra and Verheul's XTR scheme in 2000 [LV00].

Lenstra and Verheul noticed that in the very particular case $n = 6$, working in the $\mathbb{F}_{q^6}^\times$-subgroup of order $\Phi_6(q) = q^2 - q + 1$ can be done with a $\mathbb{F}_{q^2}^\times$ arithmetic, whereas the best way to break the system remains to solve discrete logarithms problems in $\mathbb{F}_{q^6}^\times$. Certainly, this yields reasonably competitive implementations. But the most surprising is that XTR subgroups are, up to symmetry, generated by the relative trace $\mathrm{Tr}_{\mathbb{F}_{q^6}/\mathbb{F}_{q^2}}$. As a consequence, we can encode them with only two elements of $\mathbb{F}_q$, with time complexity equal to $\log^{1+o(1)} q$ elementary operations.

In this paper, we exhibit for $n > 6$, n fixed, encodings that can be computed very efficiently, that is with $\log^{1+o(1)} q$ bit operations too. To this purpose, we start from the interpretation of XTR-subgroups as algebraic tori, due to Rubin and Silverberg [RS03], and the explicit encoding proposed by van Dijk and Woodruff [DW04].

Algebraic tori over $\mathbb{F}_q$ are algebraic groups defined over $\mathbb{F}_q$ that are isomorphic to some $(G_m)^d$ over $\overline{\mathbb{F}}_q$, where G_m denotes the multiplicative group and d is the dimension of the torus. Algebraic tori involved here are

$$T_n(\mathbb{F}_q) \cong \left\{ x \in \mathbb{F}_{q^n}^\times : N_{\mathbb{F}_{q^n}/F}(x) = 1 \text{ whenever } \mathbb{F}_q \subset F \subsetneq \mathbb{F}_{q^n}, F \text{ a field} \right\} . \qquad (1.1)$$

These are algebraic varieties of dimension $d = \varphi(n)$, where φ is the Euler-totient function. It turns out that in terms of group, $T_n(\mathbb{F}_q)$ is a subgroup of order $\Phi_n(q)$, that is $T_n(\mathbb{F}_q) \cong \{x \in$

Date: January 10, 2015.

$\mathbb{F}_{q^n}^\times : x^{\Phi_n(q)} = 1\}$. An efficient rational parameterization of these tori with $\varphi(n)$-tuples instead of n-tuples would thus allow the same security as in $\mathbb{F}_{q^n}^\times$, but a reduced communication cost. Even though practical constructions exist for particular values of n (for instance, 2, 3 or 6 with LUC [SL93], XTR[LV00] or CEILIDH[RS03]), the rationality or stable rationality of such structures for every n has been a concern for several years now [Vos91].

A nice workaround proposed by van Dijk and Woodruff [DW04] consists in adding to the torus $T_n(\mathbb{F}_q)$ some well chosen finite fields and mapping the whole set into another product of finite fields,

$$\theta : T_n(\mathbb{F}_q) \times \prod_{\substack{d\,|\,n \\ \mu(n/d)=-1}} \mathbb{F}_{q^d}^\times \to \prod_{\substack{d\,|\,n \\ \mu(n/d)=+1}} \mathbb{F}_{q^d}^\times\,, \tag{1.2}$$

where μ is the Moebius function. This bijection enables to compactly represent m elements of $T_n(\mathbb{F}_q)$ with roughly $m\varphi(n)$ elements in $\mathbb{F}_q$ for large enough m. For well chosen q and n, mainly n a product of distinct primes and q of maximal order modulo these primes, evaluating θ requires at least $n^{3+o(1)}\log^{2+o(1)} q$ elementary operations.

In the present work, we observe that the heaviest part of the complexity comes from exponentiations in $\mathbb{F}_{q^n}$ to powers with sparse decomposition in basis q and we succeed in speeding up the algorithm with the help of a new representation of field extensions. Couveignes and Lercier recently constructed a new family of normal bases, called normal elliptic bases [CL09]. They allow to perform low cost arithmetic in $\mathbb{F}_{q^n}$ and in the context of tori this yields encodings with a $\log q$ smaller computational cost. In order to reach this complexity, we need inputs q and n such that $\Phi_e(q)$ and $\Phi_f(q)$ are relatively prime for any distinct divisors e and f of n. This is not a big restriction in applications, since there are infinitely many n and q such that this condition holds.

It is worth to notice that the encoding cost becomes negligible in regard of the major cost in many Diffie-Hellman cryptosystems, $n^{2+o(1)}\log^{2+o(1)} q$ bit operations, due to exponentiations in $\mathbb{F}_{q^n}$. This is particularly interesting since in cryptographic applications q tends to be a large number and n rather small.

We may also remark that these ideas can be easily adapted to the improved variant of θ introduced by Dijk *et al.* in 2005 [DGP+05]. They substitute tori of small dimensions for the finite fields $\mathbb{F}_{q^d}$ in Eq. (1.2), but all the calculations still take place in $\mathbb{F}_{q^n}$ and can be sped up thanks to normal elliptic bases.

Outline. In Section 2, we present some background materials about algebraic tori encodings. Section 3 outlines some nice cyclotomic properties of these algorithms and shows how the use of a normal elliptic basis can yield a $\log q$ speedup. Section 4 discusses some of the cryptographic applications of these mappings.

2. Explicit Algebraic Tori Encodings

Van Dijk and Woodruff first proposed an algorithmic way to encode efficiently a torus $T_n(\mathbb{F}_q)$, modulo some small constraints on q and n [DW04].

2.1. Principles.

We start from the embedding $T_n(\mathbb{F}_q) \hookrightarrow \mathbb{F}_q^\times$ and we complete both sides with the missing parts in order to create a bijection.

From $q^n - 1 = \prod_{d\,|\,n}\Phi_d(q)$, we have $\mathbb{F}_q^\times \simeq \prod_{d\,|\,n} T_d(\mathbb{F}_q)$. Van Dijk and Woodruff first add the product $\prod_{d\,|\,n, d\neq n} T_d(\mathbb{F}_q)$ to the left hand side of the embedding. Then, they identify factors of the form $\mathbb{F}_{q^d}^\times$ with $d\,|\,n$ in this expression. At this step, we may have to add some newer tori, of smaller dimension. As a result, this will modify the right hand side too. But again, we identify there factors of the form $\mathbb{F}_{q^d}^\times$. After enough such iterations, this yields a bijection θ (*cf.* Eq. (1.2)).

The domain of this bijection is much larger than $T_n(\mathbb{F}_q)$, but in the case where we have m elements of $T_n(\mathbb{F}_q)$ to encode, we can nevertheless recover a quasi optimal encoding rate. We refer to Section 4.1 for details.

Example. Let us see how it works for $n = 15$. We have

$$T_1(\mathbb{F}_q) \times T_3(\mathbb{F}_q) \times T_5(\mathbb{F}_q) \times T_{15}(\mathbb{F}_q) \simeq \mathbb{F}_{q^{15}}^\times\,.$$

So, $(T_1(\mathbb{F}_q) \times T_3(\mathbb{F}_q)) \times (T_1(\mathbb{F}_q) \times T_5(\mathbb{F}_q)) \times T_{15}(\mathbb{F}_q) \simeq \mathbb{F}_{q^{15}}^\times \times T_1(\mathbb{F}_q)$, hence the bijection

$$\mathbb{F}_{q^3}^\times \times \mathbb{F}_{q^5}^\times \times T_{15}(\mathbb{F}_q) \xrightarrow{\sim} \mathbb{F}_{q^{15}}^\times \times \mathbb{F}_q^\times\,,$$

since

$$T_1(\mathbb{F}_q) \simeq \mathbb{F}_q^\times,\; T_3(\mathbb{F}_q) \times T_1(\mathbb{F}_q) \simeq \mathbb{F}_{q^3}^\times \;\text{ and }\; T_5(\mathbb{F}_q) \times T_1(\mathbb{F}_q) \simeq \mathbb{F}_{q^5}^\times.$$

Let us remark that there is no guarantee that the $\Phi_d(q)$'s are coprime, and thus this bijection may not be a group isomorphism.

2.2. **Explicit Encodings.** We now show how we can explicitly construct the bijection θ. We can obtain its inverse in the same way, but for the sake of simplicity, we omit details.

For all $d\,|\,n$, call U_d the smallest positive integer such that

$$\forall e\,|\,d,\ \forall f\,|\,d \text{ with } e \neq f,\ \gcd\left(\Phi_e(q), \Phi_f(q), \frac{q^d-1}{U_d}\right) = 1. \tag{2.1}$$

For $e\,|\,d\,|\,n$, let furthermore $y_{d,e} = \gcd\left(\Phi_e(q), (q^d-1)/U_d\right)$ and $z_{d,e} = \gcd(\Phi_e(q),\, U_d)$. Let finally w_d, $w_{d,e}$ and $u_{d,e}$, $v_{d,e}$ be the coefficients in Bézout's relations

$$\frac{q^d-1}{U_d} w_d + \sum_{e\,|\,d} \frac{q^d-1}{y_{d,e}} w_{d,e} = 1 \;\text{ and }\; \frac{\Phi_e(q)}{y_{d,e}} u_{d,e} + \frac{\Phi_e(q)}{z_{d,e}} v_{d,e} = 1\,. \tag{2.2}$$

With the notations above, we have the following bijections, for all $d\,|\,n$,

$$\mathbb{F}_{q^d}^\times \xrightarrow{\sim} \mathbb{Z}/U_d\mathbb{Z} \times \prod_{e\,|\,d} \mathbb{Z}/y_{d,e}\mathbb{Z} \;\text{ and }\; \mathbb{Z}/U_d\mathbb{Z} \xrightarrow{\sim} \prod_{e\,|\,d} \mathbb{Z}/z_{d,e}\mathbb{Z}\,.$$

These two successive bijections give a full decomposition of each $\mathbb{F}_{q^d}$ into

$$\left(\prod_{e\,|\,d} \mathbb{Z}/y_{d,e}\mathbb{Z}\right) \times \left(\prod_{e\,|\,d} \mathbb{Z}/z_{d,e}\mathbb{Z}\right)\,.$$

The first bijection is a canonical bijection given by the Chinese remainder theorem, whereas the second one is non-canonical and can be performed by a table lookup. Van Dijk and Woodruff have proved that these tables are of reasonable size when some technical conditions are satisfied by n and q, mainly n being a product of distinct primes and q of maximal order modulo these primes.

The idea is now to give a decomposition of both sides of the bijection θ and to identify the small groups on each sides. The same groups appear in a different order, except $T_n(\mathbb{F}_q)$

which is mapped into $\mathbb{Z}/y_{n,n}\mathbb{Z} \times \mathbb{Z}/Z_{n,n}\mathbb{Z}$. For each $d\,|\,n$, $d \neq n$, we identify $\prod_{e\,|\,d} \mathbb{Z}/z_{d,e}\mathbb{Z} \longrightarrow \prod_{e\,|\,d} \mathbb{Z}/z_{\rho_e(d),e}\mathbb{Z}$ where ρ_e is the bijection

$$\rho_e : \{d : e\,|\,d\,|\,n, \mu(n/d) = 1\} \xrightarrow{\sim} \{d : e\,|\,d\,|\,n, \mu(n/d) = -1\}\,.$$

All in all, we obtain Algorithm 1.

Algorithm 1: Computation of θ.

```
Input: $x \in T_n(\mathbb{F}_q)$ and $x_d \in \mathbb{F}_{q^d}^\times$ for all $d\,|\,n$ such that $\mu(n/d) = -1$.
Output: $x_d \in \mathbb{F}_{q^d}^\times$ for all $d\,|\,n$ such that $\mu(n/d) = 1$.

foreach $d\,|\,n$ such that $\mu(n/d) = -1$ do
    Compute $x_d \mapsto x_d^{(q^d-1)/U_d}$, the canonical map $\mathbb{F}_{q^d}^\times \to \mathbb{Z}/U_d\mathbb{Z}$.
    Compute $x_d^{(q^d-1)/U_d} \mapsto (Z_{d,e})_{e\,|\,d}$, the table lookup $\mathbb{Z}/U_d\mathbb{Z} \to \prod_{e\,|\,d} \mathbb{Z}/z_{d,e}\mathbb{Z}$.
    Map $(Z_{d,e})_{e\,|\,d} \mapsto (Z_{\rho_e(d),e})_{e\,|\,d}$ with $Z_{\rho_e(d),e} = (Z_{d,e}^{v_{d,e}} x_d^{(q^d-1)u_{d,e}/y_{d,e}})^{\Phi_e(q)/z_{\rho_e(d),e}}$, that is map $\prod_{e\,|\,d} \mathbb{Z}/z_{d,e}\mathbb{Z} \to \prod_{e\,|\,d} \mathbb{Z}/z_{\rho_e(d),e}\mathbb{Z}$.
end
Compute $Z_{n,n} = x^{\Phi_n(q)/z_{n,n}} \in \mathbb{Z}/z_{\rho(n),n}\mathbb{Z}$.
foreach $d\,|\,n$ such that $\mu(n/d) = 1$ do
    Compute $(Z_{d,e})_{e\,|\,d} \mapsto Z_d$, the table lookup $\prod_{\substack{\rho_e(d')=d,e\,|\,d\\ e\neq d}} \mathbb{Z}/z_{d',e}\mathbb{Z} \to \mathbb{Z}/U_d\mathbb{Z}$.
    Compute $x_d = Z_d^{w^d} \prod_{\substack{\rho_e(d')=d,e\,|\,d\\ e\neq d}} (Z_{d',e}^{v_{d',e}} x_{d'}^{(q^{d'}-1)u_{d',e}/y_{d',e}})^{\Phi_e(q)w_{d,e}/y_{d,e}} \in \mathbb{F}_{q^d}^\times$.
end
Multiply $x_n$ by $x^{\Phi_n(q)w_{n,n}/y_{n,n}}$.
```

Example. We focus again on the case $n = 15$, with $U_d = 1$ for all $d\,|\,n$ which gives good insights of what actually happens. We sketch the construction on Fig. 1.

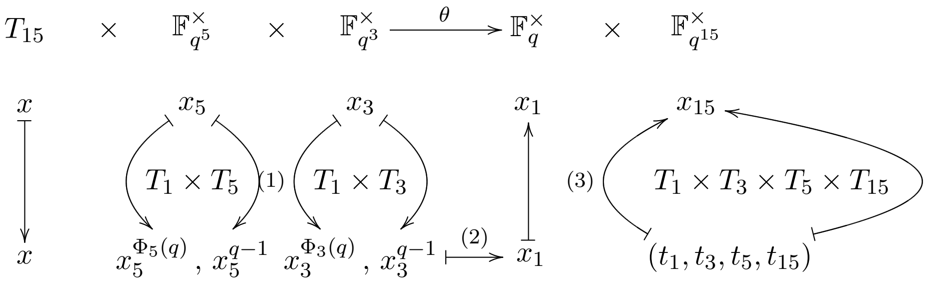
T_{15}
$\times$
$\mathbb{F}_{q^5}^\times$
$\times$
$\mathbb{F}_{q^3}^\times$
θ
$\mathbb{F}_q^\times$
$\times$
$\mathbb{F}_{q^{15}}^\times$
x
x_5
x_3
x_1
x_{15}
$T_1 \times T_5$
(1)
$T_1 \times T_3$
(3)
$T_1 \times T_3 \times T_5 \times T_{15}$
x
$x_5^{\Phi_5(q)}, x_5^{q-1}$
$x_3^{\Phi_3(q)}, x_3^{q-1}$
(2)
x_1
(t_1, t_3, t_5, t_{15})

FIGURE 1. The bijection θ for $n = 15$ and $U_1 = U_3 = U_5 = U_{15} = 1$.

We have here several simplifications. For every $e\,|\,d$, $y_{d,e} = \Phi_e(q)$ and $z_{d,e} = 1$. Then the groups $\mathbb{Z}/y_{d,e}\mathbb{Z}$ involved are nothing but the tori $T_e(\mathbb{F}_q)$. Besides $u_{d,e} = 1$ and $v_{d,e} = 0$. Eq. (2.2) becomes $\sum_{e\,|\,d} \frac{q^d-1}{\Phi_e(q)} w_{d,e} = 1$, and x_{15} is simply given by $x_{15} = t_1^{w_{15,1}} t_3^{w_{15,3}} t_5^{w_{15,5}} t_{15}^{w_{15,15}}$.

An explicit computation shows that the $w_{15,e}$'s have a convenient common denominator, namely 15. So, $x_{15} = (t_1^{r_1} t_3^{r_3} t_5^{r_5} t_{15}^{r_{15}})^{1/15}$, where the r_e's are convenient polynomials in q,

$$\begin{cases} r_1 = 1, \\ r_3 = -q - 2, \\ r_5 = -q^3 - 2\,q^2 - 3\,q - 4, \\ r_{15} = q^7 - 3\,q^5 + 4\,q^4 - 5\,q^3 + 7\,q - 8. \end{cases}$$

The cost is as follows (*cf.* Fig. 1).

Phase (1): Exponentiations to the powers $q-1$, $\Phi_3(q) = q^2+q+1$ and $\Phi_5(q) = q^4+q^3+q^2+q+1$ cost in average, respectively, $\frac{1}{2}\log q$, $\frac{1}{2}(2\log q)$ and $\frac{1}{2}(4\log q)$ multiplications since we perform exponentiations to power of the sizes q, q^2 and q^4.

Phase (2): Negligible.

Phase (3): Recall the expressions of the r_e's. Exponentiation to these powers demands in average $\deg r_e \times (\frac{1}{2}\log q)$. So altogether: $(0+1+3+7) \times (\frac{1}{2}\log q)$.

This elementary calculation shows that, in average, the cost is about $9\log q$ multiplications in $\mathbb{F}_{q^{15}}$, that is $\log^{2+o(1)} q$ elementary operations. Van Dijk and Woodruff propose some insights to improve this cost in practice (multi-exponentiations, redundancies, *etc.*), but the asymptotic complexity remains quasi-quadratic in $\log q$.

2.3. **Computational Complexities.** We can now state more precisely the complexity of Algorithm 1.

We first construct an irreducible polynomial $P(X)$ of degree n over $\mathbb{F}_q$, which can be done in $n^{2+o(1)}\log^{2+o(1)} q$ operations [PR98]. Let $\alpha = X \bmod P(X)$. Then $(1, \alpha, \ldots, \alpha^{n-1})$ is an $\mathbb{F}_q$-basis of $\mathbb{F}_{q^n}$. Additions, subtractions and comparisons require $O(n\log q)$ elementary operations. Multiplications and divisions require $n^{1+o(1)}\log^{1+o(1)} q$ elementary operations.

We also have to handle basis changes between $\mathbb{F}_{q^n}$ and its subfields $\mathbb{F}_{q^d}$. There are $d(n)$ such subfields, where $d(n)$ is the divisor function. This may yield large finite field lattices (see Fig. 2 for an example). To simplify things, and since it does not change the complexity, we consider that $\mathbb{F}_{q^d}$ elements for $d \,|\, n$ are given in the basis $(1, \alpha, \ldots, \alpha^{n-1})$ too. So, we can easily multiply elements given in two distinct subfields. Just, in order to obtain the right dimensions for inputs or outputs of the algorithm, we apply to an $\mathbb{F}_{q^d}$ element given in $\mathbb{F}_{q^n}$ an $\mathbb{F}_q$-linear compression derived from equations of the type $x^{q^d} = x$. This yields matrices $A_{n,d} \in \mathcal{M}_{n,d}(\mathbb{F}_q)$ for the embedding $\mathbb{F}_{q^d} \hookrightarrow \mathbb{F}_{q^n}$. Building and applying such a matrix costs at most n^3 multiplications in $\mathbb{F}_q$. Since there are $d(n) \simeq n^{o(1)}$ of them, this yields a total cost of $n^{3+o(1)}\log^{1+o(1)} q$ bit operations.

Van Dijk and Woodruff outline that for "reasonable" integers n and q, mainly n a product of distinct primes and q of maximal order modulo these primes, table lookup costs are negligible and the main costs are Step 4 and Step 9 of the algorithm. They involve exponents which are derived from cyclotomic polynomials. Computing Φ_n can be done in time essentially equal to its size (start from complex floating point approximations of primitive n-th roots of unity and reconstruct Φ_n from these roots). We know that this is a polynomial of degree $\varphi(n)$ with coefficients upperbounded by $n^{d(n)/2}$ [Erd46, Bat49], that is a size of at most $n^{1+o(1)}$ bits. Evaluating all the Φ_d's at q yields exponents with $d\log q$ bits and can be done with $n^{2+o(1)}\log^{1+o(1)} q$ elementary operations. Using finally the approximate growth rate $\sum_{d\,|\,n} d \simeq n^{1+o(1)}$, the total cost of Step 4 and Step 9 is equal to $n^{3+o(1)}\log^{2+o(1)} q$.

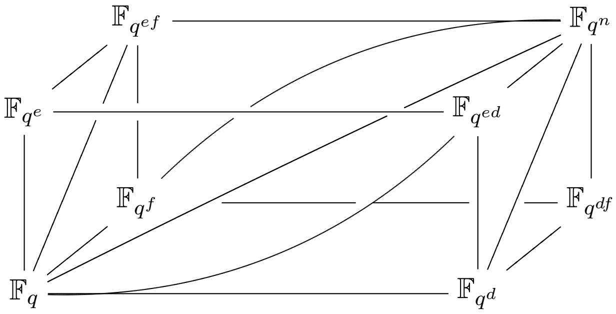

Figure 2. Finite field lattices for $n = def$, a product of three distinct primes.

3. Elliptic Periods and Algebraic Tori

We now focus on the case $U_d = 1$ for every $d\,|\,n$. That is no big restriction, at least for cryptographic purposes. Indeed Lemma 1 in Section 3.1 shows that we can find infinitely many values of q for infinitely many values of n working.

We observe in Section 3.3 that most of the exponentiations occuring in Algorithm 1 involve exponents with a sparse decomposition in basis q. This yields interests for handling $\mathbb{F}_{q^n}$ with a normal basis $(\alpha, \alpha^q, \ldots, \alpha^{q^{n-1}})$ instead of a power basis $(1, \alpha, \ldots, \alpha^{n-1})$, since with such a choice q-th powers become inexpensive. Since we need to multiply elements of $\mathbb{F}_{q^n}$ in quasi-linear time too, normal elliptic basis are a natural choice that we introduce in Section 3.2.

3.1. Restrictions on n and q.

For squarefree integers n, we can prove the following result.

Lemma 1. *For infinitely many squarefree integers n, there are infinitely many values of q such that $U_d = 1$ for all $d\,|\,n$.*

Proof. From Eq. (2.1), we deduce

$$U_d = 1 \Leftrightarrow \forall e\,|\,d,\ \forall f\,|\,d\ e \neq f,\ \gcd(\Phi_e(q), \Phi_f(q)) = 1\,. \tag{3.1}$$

The right hand side condition is always satisfied when $\mathrm{Res}(\Phi_e, \Phi_f) = 1$ and it is widely known that this is equivalent to the condition $f \neq e\,p^i$ with p prime and $i \geqslant 1$ (see [Dun09] for a proof). This is a corollary of the following formula due to Apostol [Apo70], for $f > e > 1$,

$$\mathrm{Res}(\Phi_f, \Phi_e) = \prod_{\substack{d\,|\,e \\ p \text{ prime},\ \frac{f}{(f,d)} = p^i}} p^{\mu(e/d)\frac{\varphi(f)}{\varphi(p^i)}}\,. \tag{3.2}$$

There remains to check that when $f = e\,p^i$, there exist integers q such that Eq. (3.1) is satisfied. Since n is supposed to be squarefree, the only cases are $f = ep$, p prime.

Case $e = 1$: The divisor f is then equal to the prime p and $\mathrm{Res}(\Phi_1, \Phi_f) = f$. In order to have $\gcd(\Phi_e(q), \Phi_f(q)) = 1$, q must not be a common root of Φ_e and Φ_f modulo f. In other words, we must have $q \not\equiv 1 \bmod f$.

Case $e > 1$**:** The divisor f is then equal to pe where p is a prime. Since e is squarefree, we know from Eq. (3.2) that $\mathrm{Res}(\Phi_e, \Phi_{pe}) = p^{\varphi(e)}$. So, q must not be a common root of Φ_e and Φ_{pe} modulo p. Modulo p, Φ_e have a decomposition into irreducible polynomials of same degree, and this degree is equal to $p \bmod e$ (*cf.* [LN83]). In other words, Φ_e and Φ_{pe} can only have a common root when $p \equiv 1 \bmod e$. In this case, q must not be one of the $\varphi(e)$ roots of Φ_e modulo p.

The restrictions above leave infinitely many possibilities for q, at least for infinitely many values of n. For instance let $n = p(p+2)$ be the product of two twin primes and q such that $q \not\equiv 1 \bmod p$ and $q \not\equiv 1 \bmod (p+2)$. Besides since $p+2 \not\equiv 1 \bmod p$, all the conditions above are satisfied. Thus we have a infinite family of numbers q suitable for each n, and an infinite number of possible values for n itself. □

3.2. **Normal Elliptic Basis.** We mimic here Couveignes and Lercier's construction.

Let $E/\mathbb{F}_q$ be an elliptic curve given by some Weierstrass model

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3\,.$$

If A is a point in $E(\mathbb{F}_q)$, we denote by $\tau_A : E \to E$ the translation by A. We set $x_A = x \circ \tau_{-A}$ and $y_A = y \circ \tau_{-A}$. If A, B and C are three pairwise distinct points in $E(\mathbb{F}_q)$, we define

$$\Gamma(A, B, C) = \frac{y(C-A) - y(A-B)}{x(C-A) - x(A-B)}\,.$$

We define a function $u_{A,B} \in \mathbb{F}_q(E)$ by $u_{A,B}(C) = \Gamma(A, B, C)$. It has degree two with two simple poles, at a and b.

We can prove the following identities (with Taylor expansions at poles),

$$\left\{\begin{array}{rcl} \Gamma(A,B,C) & = & \Gamma(B,C,A) = -\Gamma(B,A,C) - a_1, \\ & = & -\Gamma(-A,-B,-C) - a_1\,, \\ u_{A,B} + u_{B,C} + u_{C,A} & = & \Gamma(A,B,C) - a_1\,, \\ u_{A,B}u_{A,C} & = & x_A + \Gamma(A,B,C)u_{A,C} + \Gamma(A,C,B)u_{A,B} \\ & & +a_2 + x_A(B) + x_A(C)\,, \\ u_{A,B}^2 & = & x_A + x_B - a_1u_{A,B} + x_A(B) + a_2\,. \end{array}\right. \tag{3.3}$$

Assume $E(\mathbb{F}_q)$ contains a cyclic subgroup $\mathcal{T}$ of order n and let $I : E \to E'$ be the degree n cyclic isogeny with kernel $\mathcal{T}$, then the quotient $E'(\mathbb{F}_q)/I(E(\mathbb{F}_q))$ is isomorphic to $\mathcal{T}$.

Take A in $E'(\mathbb{F}_q)$ such that $A \bmod I(E(\mathbb{F}_q))$ generates this quotient. The fiber $\mathcal{P} = I^{-1}(A) = \sum_{T\in\mathcal{T}}[B+T]$ is an irreducible divisor. The n geometric points above A are defined on a degree n extension of $\mathbb{F}_q$ (and permuted by Galois action), that is $\mathbb{F}_{q^n}$ is the residue extension of $\mathbb{F}_q(E)$ at $\mathcal{P}$.

For $k \in \mathbb{Z}/n\mathbb{Z}$, we set $u_k = \mathfrak{a}u_{kT,(k+1)T} + \mathfrak{b}$. ($\mathfrak{a}$ and $\mathfrak{b}$, constants chosen such that $\sum u_k = 1$). Then the system $\Theta = (u_k(B))_{k\in\mathbb{Z}/n\mathbb{Z}}$ is an $\mathbb{F}_q$ normal basis of $\mathbb{F}_{q^n}$.

Furthermore, there exists an algorithm with quasi-linear complexity to multiply two elements given in an elliptic normal basis, mostly based on Eq. (3.3). It consists in evaluations and interpolations at d points $R + kT$, where $R \in E(\mathbb{F}_q) - E[n]$.

All of these yields Theorem 1.

Theorem 1 ([CL09])**.** *To every couple (q, n) with q a prime power and $n \geqslant 2$ an integer such that $n_q \leqslant \sqrt{q}$, one can associate a normal basis $\Theta(q, n)$ of the degree n extension of $\mathbb{F}_q$ such that the following holds.*

- *There exists an algorithm that multiplies two elements given in $\Theta(q, n)$ at the expense of $n^{1+o(1)} \log^{1+o(1)} q$ elementary operations.*

Here n_q is such that

- $v_\ell(n_q) = v_\ell(n)$ if ℓ is prime to $q - 1$, $v_\ell(n_q) = 0$ if $v_\ell(n) = 0$,
- $v_\ell(n_q) = \max(2v_\ell(q - 1) + 1, 2v_\ell(n))$ if ℓ divides both $q - 1$ and n.

This result can be easily extended to a result without any restriction on q and n (see [CL09]).

3.3. **Van Dijk and Woodruff's Encoding Revisited.** Since $U_d = 1$ for all $d \,|\, n$, van Dijk and Woodruf's encoding can be slightly simplified. It is not only a bijection, but also a group isomorphism.

For every $e \,|\, d$, $y_{d,e} = \Phi_e(q)$ and $z_{d,e} = 1$. Then the groups $\mathbb{Z}/y_{d,e}\mathbb{Z}$ involved are nothing but the tori $T_e(\mathbb{F}_q)$. Besides $u_{d,e} = 1$ and $v_{d,e} = 0$. So most of Algorithm 1 is reduced to two main phases: the decomposition $\mathbb{F}_{q^d}^\times \to \prod_{e \,|\, d} T_e(\mathbb{F}_q)$ for d any divisor of n such that $\mu(n/d) = -1$ on the left hand side and the reconstruction $\prod_{e \,|\, d} T_e(\mathbb{F}_q) \to \mathbb{F}_{q^d}^\times$ for d any divisor of n such that $\mu(n/d) = 1$ on the right hand side.

Now we need to know what we gain with a normal elliptic basis. Essentially, it makes each exponentiation to a power of q be a simple permutation of the basis. We thus gain a $\log q$ factor for each exponentiation of this type. It is not difficult to see that the exponents occuring in the decomposition phase have a sparse decomposition in basis q since they are products of evaluations of cyclotomic polynomials at q. But the reconstruction phase is more tricky because it involves exponentiations by Bézout's coefficients $w_{d,e}$ which do not have such a nice decomposition in basis q. Instead, we prefer to compute Bézout's polynomials $W_{d,e}$ such that

$$\sum_{e \,|\, d} \frac{X^d - 1}{\Phi_e(X)} W_{d,e}(X) = 1\,.$$

Of course, $w_{d,e} = W_{d,e}(q) \bmod \Phi_e(q)$.

Unlike cyclotomic polynomials, these polynomials do not have integer coefficients, but for squarefree integers n, and thus squarefree divisors d, all their coefficients have a common denominator, equal to d. More precisely, we have

$$W_{d,e}(X) = \prod_{f \,|\, d, f \neq e} \Phi_f(X)^{-1} \bmod \Phi_e(X)\,. \tag{3.4}$$

We may notice on the first hand that $\Phi_f(X)^{-1} \bmod \Phi_e(X)$ has got integer coefficients if and only if $f \neq e\,p^i$ with p prime and $i \geqslant 1$, since $\mathrm{Res}(\Phi_e, \Phi_f) = 1$ in that case (see proof of Lemma 1). On the other hand, when $f = e\,p^i$, the coefficients of $\Phi_f(X)^{-1} \bmod \Phi_e(X)$ have a common denominator, equal to f. From Eq. (3.4), and from the squarefree property satisfied by d, we deduce thus that the coefficients of $W_{d,e}(X)$ have a common denominator exactly equal to d.

We observed that the numerators $R_{d,e}$ of the $W_{d,e}$'s have small coefficients too (see Section 3.3.1 for a detailed analysis in the case $n = pr$). Consequently, we restrict q to prime powers such that n is invertible modulo $q^n - 1$ and slightly modify θ to output x_d^n instead of x_d for each $d \,|\, n$ such that $\mu(n/d) = 1$. We denote $\widetilde{\theta}$ this variant (*cf.* Algorithm 2).

Algorithm 2: Computation of $\widetilde{\theta}$.

```
Input: $x \in T_n(\mathbb{F}_q)$ and $x_d \in \mathbb{F}_{q^d}^\times$ for all $d \,|\, n$ such that $\mu(n/d) = -1$.
Output: $x_d \in \mathbb{F}_{q^d}^\times$ for all $d \,|\, n$ such that $\mu(n/d) = 1$.

foreach $d \,|\, n$ such that $\mu(n/d) = -1$ do
    Compute $x_d \mapsto (Z_{\rho_e(d),e})_{e \,|\, d}$ with $Z_{\rho_e(d),e} = x_d^{(q^d-1)/\Phi_e(q)}$.
end
Set $Z_{n,n} = x$.
foreach $d \,|\, n$ such that $\mu(n/d) = 1$ do
    Compute $x_d = \prod_{\substack{\rho_e(d')=d, e \,|\, d \\ e \neq d}} Z_{d',e}^{nW_{d,e}(q)} \in \mathbb{F}_{q^d}^\times$.
end
```

Fortunately, we do not need any more compression matrices $A_{n,d}$ with normal basis (*cf.* Section 2.3). In truth, a $\mathbb{F}_{q^d}$ element has got a periodic set of components in any normal basis of $\mathbb{F}_{q^n}$. Consequently, compressing simply consists in truncating to the d first components and expanding consists in concatenating n/d copies of a d-tuple of $\mathbb{F}_q$ elements. Costs are negligible.

Before considering in detail the case $n = pr$ a product of two primes in Section 3.3.1, and discuss the general case in Section 3.3.2, we focus on an explicit example, namely $n = 15$ in order to compare with Section 2.2.

Example. Recall Fig. 1 for the notations, the costs are the following.

Phase (1) : Exponentiations to the powers $\Phi_3(q) = q^2 + q + 1$ and $\Phi_5(q) = q^4 + q^3 + q^2 + q + 1$ cost respectively 2 and 4 multiplications since exponentiation to a power of q is free (mere permutation of the basis). Exponentiation to the power $q - 1$ costs an inversion, which is performed in linear time.

Phase (2) : Negligible.

Phase (3) : Recall the expressions of the r_e's. For instance $r_{15} = q^7 - 3\,q^5 + 4\,q^4 - 5\,q^3 + 7\,q - 8$. Exponentiation to this power demands 6×3 multiplications for the coefficients (6 coefficients of size at most 2^3) and 6 multiplications to add the 7 monomials. The same calculation for each r_e gives the global cost of Phase (3): $3 + ((0) + (1 \times 1 + 1) + (2 \times 2 + 2) + (6 \times 3 + 6))$ multiplications and 3 inversions.

If we remind the total found for computations without normal elliptic bases, it is a clear practical improvement. The most important is that asymptotically, the $\log q$ factor vanishes.

$$\begin{tikzcd}
T_n & \times & \mathbb{F}_{q^p}^{\times} & \times & \mathbb{F}_{q^r}^{\times} \arrow[r] & \mathbb{F}_q^{\times} & \times & \mathbb{F}_{q^{pr}}^{\times} \\
x \arrow[dd, mapsto] & & x_p \arrow[dd, mapsto, bend right] \arrow[dd, mapsto, bend left] & & x_r \arrow[dd, mapsto, bend right] \arrow[dd, mapsto, bend left] & x_1^n & & x_{pr}^n \\
 & & T_1 \times T_p & (1) & T_1 \times T_r & & (3) & T_1 \times T_p \times T_r \times T_{pr} \\
x & & x_p^{\Phi_p(q)}\,,\ x_p^{q-1} & & x_r^{\Phi_r(q)}\,,\ x_r^{q-1} \arrow[r, mapsto, "(2)"] & x_1 \arrow[uu, mapsto] & & (t_1, t_p, t_r, t_{pr}) \arrow[uu, mapsto, bend left] \arrow[uu, mapsto, bend right]
\end{tikzcd}$$

FIGURE 3. The bijection $\widetilde{\theta}$ for $n = pr$ and $U_1 = U_p = U_r = U_{pr} = 1$.

3.3.1. *Case $n = pr$ with p, r distinct primes.* In the case $n = pr$ with p, r distinct primes, the situation is very similar to our $n = 15$ example (*cf.* Fig. 3).

Especially, the cost of Phase (1) comes from exponentiations to the powers $\Phi_p(q)$ and $\Phi_r(q)$, that is p and r multiplications since exponentiation to a power of q is free. This costs $n^{2+o(1)} \log^{1+o(1)} q$ bit operations. Exponentiation to the power $q-1$ costs an inversion, which is asymptotically performed in quasi-linear time.

We now give details on the cost of Phase (3). We perform the embedding in two steps. First, we combine t_1 and t_{pr} on one hand and t_p and t_r on the other hand. Then, we combine the two results again to form the element x_{pr}. We summarize this process on Fig. 4.

$$\begin{tikzcd}
(T_1(\mathbb{F}_q) \times T_{pr}(\mathbb{F}_q)) & \times & (T_p(\mathbb{F}_q) \times T_r(\mathbb{F}_q)) \arrow[r] & \mathbb{F}_{q^{pr}}^{\times} \\
(t_1, t_{pr}) \arrow[dd, mapsto, bend right] & , & (t_p, t_r) \arrow[dd, mapsto, bend left] & x_{pr} = y_1^{v_1} y_2^{v_2} \\
G_1 & \times & G_2 & \\
y_1 = t_1^{u_1} t_{pr}^{u_{pr}} & & y_2 = t_p^{u_p} t_r^{u_r} \arrow[uur, mapsto, bend right] &
\end{tikzcd}$$

FIGURE 4. Reconstruction step in the case $n = pr$.

So the first step consists in two mappings,

$$\begin{array}{rcl} T_1(\mathbb{F}_q) \times T_{pr}(\mathbb{F}_q) & \xrightarrow{\sim} & G_1 \subset \mathbb{F}_{q^{pr}}^{\times}\,, \\ (t_1, t_{pr}) & \mapsto & y_1 = t_1^{u_1} t_{pr}^{u_{pr}}\,, \end{array} \quad \text{where } \Phi_{pr}(q) u_1 + \Phi_1(q) u_{pr} = 1$$

and

$$\begin{array}{rcl} T_p(\mathbb{F}_q) \times T_r(\mathbb{F}_q) & \xrightarrow{\sim} & G_2 \subset \mathbb{F}_{q^{pr}}^{\times} \\ (t_p, t_r) & \mapsto & y_2 = t_p^{u_p} t_r^{u_r} \end{array} \quad \text{where } \Phi_r(q) u_p + \Phi_p(q) u_r = 1\,.$$

The final recombination is

$$\begin{array}{rcl} G_1 \times G_2 & \to & \mathbb{F}_{q^{pr}}^{\times} \\ (y_1, y_2) & \mapsto & y_1^{v_1} y_2^{v_2} \end{array} \quad \text{where } \frac{q^{pr}-1}{\Phi_1(q)\Phi_{pr}(q)} v_1 + \frac{q^{pr}-1}{\Phi_p(q)\Phi_r(q)} v_2 = 1\,.$$

The powers involved in the mappings of the first step, u_1, u_p, u_r and u_{pr} are the evaluations in q of respectively $\Phi_{pr}^{-1} \bmod \Phi_1$, ${\Phi_r}^{-1} \bmod \Phi_p$, ${\Phi_p}^{-1} \bmod \Phi_r$, ${\Phi_1}^{-1} \bmod \Phi_{pr}$. Actually, the

n-th cyclotomic polynomial has small coefficients, $n^{1+o(1)}$ bits (*cf.* Section 2.3), and its computation can be done with $n^{2+o(1)}$ elementary operations.

We would need similar magnitude results for modular inverses of cyclotomic polynomials. To that end, Dunand recently found such bounds.

Theorem 2 ([Dun09])**.** *For all p and r distinct prime numbers,*

(i) $\Phi_p^{-1} \bmod \Phi_1 = 1/p$ *and* $\Phi_1^{-1} \bmod \Phi_p = (-1/p)(X^{p-2} + 2X^{p-3} + \ldots + p - 1)$.

(ii) $\Phi_{pr}^{-1} \bmod \Phi_1 = 1$ *and* $\Phi_1^{-1} \bmod \Phi_{pr} = \sum_{i=0}^{\varphi(pr)-1} v_i X^i$ *with* $v_i \in \{-1, 0, 1\}$.

(iii) $\Phi_{pr}^{-1} \bmod \Phi_p = \frac{1}{r}\sum_{i=0}^{d} X^i$ *with* $d \equiv r - 1 \bmod p$ *and* $\Phi_p^{-1} \bmod \Phi_{pr} = \frac{1}{r}\sum_{i=0}^{\varphi(pr)-1} v_i X^i$ *with* $v_i < r$.

(iv) $\Phi_p^{-1} \bmod \Phi_r = \sum_{i=0}^{\varphi(r)-1} v_i X^i$ *with* $v_i \in \{0, -1, +1\}$.

The decomposition of u_1, u_p, u_r and u_{pr} in basis q is very sparse, with only -1, 0, or 1 coefficients. The complexity of this step is thus $O(n)$ multiplications and few inversions in $\mathbb{F}_{q^n}$, that is $n^{2+o(1)} \log^{1+o(1)} q$ elementary operations.

The powers in the second step, v_1 and v_2, are the evaluations in q of respectively $\Phi_p^{-1}\Phi_r^{-1} \bmod \Phi_1\Phi_{pr}$ and $\Phi_1^{-1}\Phi_{pr}^{-1} \bmod \Phi_p\Phi_r$. Their computations require the knowledge of Φ_p^{-1} modulo Φ_1 and Φ_{pr}, Φ_r^{-1} modulo Φ_1 and Φ_{pr}, Φ_1^{-1} modulo Φ_p and Φ_r and finally Φ_{pr}^{-1} modulo Φ_p and Φ_r. To compute inverses modulo a product of two cyclotomic polynomials, we make use of the Chinese remainder theorem. If $\Phi = A \bmod \Phi_{pr}$ and $\Phi = B \bmod \Phi_1$, then

$$\Phi = \left(\frac{\Phi_1}{\Phi_1 \bmod \Phi_{pr}} A + \frac{\Phi_{pr}}{\Phi_{pr} \bmod \Phi_1} B \right) \bmod \Phi_1\Phi_{pr} \,.$$

And we have of course a similar formula for the second case. This yields the following coefficient bounds (in absolute value),

$$\begin{aligned} \Phi_p^{-1} \bmod \Phi_1\Phi_{pr} = {} & \underbrace{\Phi_1}_{\text{at most } 1} \underbrace{(\Phi_1^{-1} \bmod \Phi_{pr})}_{\text{at most } 1} \underbrace{(\Phi_p^{-1} \bmod \Phi_{pr})}_{=1/p} \\ & + \underbrace{\Phi_{pr}}_{\text{at most } 1} \underbrace{(\Phi_{pr}^{-1} \bmod \Phi_1)}_{=1} \underbrace{(\Phi_p^{-1} \bmod \Phi_1)}_{\text{at most } r} \operatorname{mod} \Phi_1\Phi_{pr} \end{aligned} \tag{3.5}$$

We have such a bound for $\Phi_r^{-1} \bmod \Phi_1\Phi_{pr}$ too (exchange p and r in Eq. (3.5)).

Finally v_1 is the product of Φ_p^{-1} and Φ_r^{-1} modulo $\Phi_1\Phi_{pr}$. The factor $1/pr$ appearing leads us to return x_{pr}^n instead of x_{pr}. So the powers involved in the last step will be nv_1 and nv_2. A very quick analysis show that the coefficients of their decomposition in basis q are upperbounded in absolute value by n^5 and this impacts the complexity by an additional but negligible $n^{o(1)}$ penalty. The total complexity of the reconstruction phase is thus equal to $n^{2+o(1)} \log^{1+o(1)} q$ elementary operations.

As a conclusion, our variant of the bijection θ asymptotically costs, for $n = pr$ the product of two primes, $n^{2+o(1)} \log^{1+o(1)} q$ elementary operations.

3.3.2. *Case of integers n with more than two prime factors.* The decomposition phase is the easiest to quantify for general n. We have to perform exponentiations to powers equal to cyclotomic polynomials evaluated at q. Since we have at most $d(n) = n^{o(1)}$ such polynomials, since they are of degree at most n and since their coefficients have got $n^{1+o(1)}$ bits, this yields a clear $n^{3+o(1)} \log^{1+o(1)} q$ bit complexity.

The reconstruction phase involves modular inverses of cyclotomic polynomials and with our current knowledge, is seems very difficult to have in full generality bounds similar to Dunand's ones in the case $n = pr$. It seems, but we have no proof of this, that for integers n with a fixed number of prime factors, the coefficients of these cyclotomic inverses are upperbounded in absolute value by a fixed power of n. And so, the reconstruction complexity would not exceed the complexity of the decomposition phase.

For more general integers n, it is very hard to state something, except of course that the complexity is no longer quasi-quadratic, but quasi-linear, in $\log q$.

4. Cryptographic Applications

In [DW04], van Dijk and Woodruff give several applications, including a Diffie-Hellman-like multiple key exchange. We show here how this scheme can be adapted to our case.

4.1. **Key agreement.** We denote in the following $\theta : T_n(\mathbb{F}_q) \times \Pi^- \to \Pi^+$, the bijection θ initially defined by Eq. (1.2).

Let us assume that Alice and Bob need to agree not on a single key but on a sequence $(K_i)_{1\leqslant i\leqslant m}$ of keys, with a Diffie-Hellman based system. Indeed, after having agreed on a generator g of $T_n(\mathbb{F}_q)$, each of the keys will be $K_i = g^{x_i y_i}$ where x_i and y_i will be randomly chosen respectively by Alice and Bob.

Alice computes the points $A_i = g^{x_i}$ on the torus and after having chosen a random $S_0 \in \Pi^-$, she computes in turn $\theta(A_i, S_{i-1}) = (a_i, S_i)$ for i from 1 to m. She sends the $(a_i)_{1\leqslant i\leqslant m}$ and the last output S_m to Bob. So he can recover all the A_i's by applying $\theta^{-1}(a_i, S_i) = (A_i, S_{i-1})$ for i decreasing from m to 1. Finally the key is $K_i = A_i^{y_i}$.

In this way, S_m and $a_1, \ldots, a_m$ encode $A_1, \ldots, A_m$. This encoding is optimal except the small overhead S_m, that is negligible for a large enough m.

Similarly, if Bob chooses $T_0 \in \Pi^-$ and computes successively $(b_i, T_i) = \theta(B_i, T_{i-1})$, he can send $(b_i)_i$ and T_m to Alice, who can recover $(B_i)_i$ by $(B_i, T_{i-1}) = \theta^{-1}(b_i, T_i)$, for i from m to 1. Then $K_i = B_i^{x_i}$ gives the keys.

4.2. **Adaptation.** We need to modify this system since our bijection $\widetilde{\theta}$ is not exactly the same.

We focus here on the case $n = pr$ but it works in the same way for more general integers n. We want to use the bijection given in Fig. 3. Yet what we can efficiently calculate in the third step is $(t_1, t_p, t_r, t_{pr}) \mapsto x_{pr}^n$. So we are going to use the slightly different mapping $\widetilde{\theta}$ and a reverse mapping $\widetilde{\theta}'$,

$$\widetilde{\theta}: \begin{array}{ccc} T_n(\mathbb{F}_q)\times\mathbb{F}_{q^p}^\times\times\mathbb{F}_{q^r}^\times & \to & \mathbb{F}_q^\times\times\mathbb{F}_{q^n}^\times\,,\\ (x,x_p,x_r) & \mapsto & (x_1^n, x_n^n)\,,\end{array} \text{ and } \widetilde{\theta}': \begin{array}{ccc} \mathbb{F}_q^\times\times\mathbb{F}_{q^n}^\times & \to & T_n(\mathbb{F}_q)\times\mathbb{F}_{q^p}^\times\times\mathbb{F}_{q^r}^\times\,,\\ (x_1,x_n) & \mapsto & (x^n, x_p^n, x_r^n)\,.\end{array}$$

Since $\widetilde{\theta}' \circ \widetilde{\theta}(x, x_p, x_r)$ is no longer equal to (x, x_p, x_r) but to $(x^{n^2}, x_p^{n^2}, x_r^{n^2})$, we cannot make a direct use of the previous Diffie Hellman scheme. We have to raise the output of our mappings to the $1/n$-th power instead. This can be easily done by a straightforward exponentiation, but at cost $n^{2+o(1)} \log^{2+o(1)} q$.

It turns out that this cost can be decreased, but at the expense of an additional constraint on q.

Lemma 2. *Let n be an odd integer, let q be a prime power such that n divides $q+1$ and denote $k=(n-1)/2$, then*

$$1/n \bmod (q^n-1) = \mu_0 + \mu_1\, q + \mu_0\, q^2 + \cdots + \mu_1\, q^{n-2} + \mu_0\, q^{n-1}\,, \tag{4.1}$$

where

$$\mu_0 = \frac{k(q-1)+q}{n} \text{ and } \mu_1 = \frac{k(q-1)-1}{n}\,.$$

Proof. We have

$$n\,(\mu_0 + \mu_1\, q + \mu_0\, q^2 + \cdots + \mu_1\, q^{n-2} + \mu_0\, q^{n-1}) - 1 - k\,(q^n-1) = \\ \frac{kq^{n+2} + n\mu_0\, q^{1+n} + nq^n\,(\mu_1 - k) - (k+1)\, q^2 - n\mu_1\, q - n\mu_0 + k + 1}{q^2-1}\,.$$

The numerator of the right hand side is thus equal to

$$q^n(kq^2 + n\mu_0 q + n(\mu_1 - k)) - (k+1)\, q^2 - n\mu_1\, q - n\mu_0 + k + 1$$

and then we need to check that the coefficient of q^n and the remaining part of this expression are both equal to zero with μ_0 and μ_1 as given above. □

Raising elements of $\mathbb{F}_{q^n}$ to the $1/n$-th power where $1/n$ is given by Eq. (4.1) can be done with $n^{1+o(1)} \log^{2+o(1)} q$ elementary operations with a normal basis. The global asymptotical cost of the encodings in the key agreement is thus in this case m times $n^{2+o(1)} \log^{1+o(1)} q + n^{1+o(1)} \log^{2+o(1)} q$ bit operations. This is smaller than m times $n^{2+o(1)} \log^{2+o(1)} q$, the cost of m Diffie-Hellman exponentiations.

Remark. Computing n-th roots in $\mathbb{F}_{q^n}$ excludes even integers n in the construction, at least for odd prime powers q. But an easy workaround consists in working in the quadratic residue subgroup of $T_1(\mathbb{F}_q)$ and $T_2(\mathbb{F}_q)$. This is equivalent to substitute $(q-1)/2$ and $(q+1)/2$ for $\Phi_1(q)$ and $\Phi_2(q)$ everywhere in the construction of $\widetilde{\theta}$. So, we are left at the end to compute $n/2$-th roots in $\mathbb{F}_{q^n}$ and all of these do not change the overall complexity of the scheme.

References

[Apo70] T. M. Apostol, *Resultants of cyclotomic polynomials*, Proceedings of the American Mathematical Society **24** (1970), 457–462.

[Bat49] P. T. Bateman, *Note on the coefficients of the cyclotomic polynomial*, Bulletin of the American Mathematical Society **55** (1949), no. 12, 1180–1181.

[CL09] J.-M. Couveignes and R. Lercier, *Elliptic periods for finite fields*, Finite Fields and their Applications **15** (2009), no. 1, 1–22.

[DGP+05] M. van Dijk, R. Granger, D. Page, K. Rubin, A. Silverberg, M. Stam, and D. P. Woodruff, *Practical Cryptography in High Dimensional Tori*, Advances in Cryptology - EUROCRYPT 2005 (Ronald Cramer, ed.), Lecture Notes in Computer Science, vol. 3494, Springer, 2005, pp. 234–250.

[DH76] W. Diffie and M. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory **22** (1976), no. 6, 644–654.

[Dun09] C. Dunand, *On Modular Inverses of Cyclotomic Polynomials and the Magnitude of their Coefficients*, Preprint, 2009, Available at http://arxiv.org/abs/0907.5543.

[DW04] M. van Dijk and D. Woodruff, *Asymptotically Optimal Communication for Torus-Based Cryptography*, Advances in Cryptology – CRYPTO ' 2004 (Matthew K. Franklin, ed.), Lecture Notes in Computer Science, vol. 3152, Springer, 2004, pp. 157–178.

[Erd46] P. Erdös, *On the coefficients of the cyclotomic polynomial*, Bulletin of the American Mathematical Society **52** (1946), no. 2, 179–184.

[LN83] R. Lidl and H. Niederreiter, *Finite Fields*, Encyclopedia of Mathematics and its Applications, vol. 20, Addison–Wesley, 1983.

[LV00] A. K. Lenstra and E. R. Verheul, *The XTR public key system*, Advances in Cryptology – CRYPTO ' 2000 (Mihir Bellare, ed.), Lecture Notes in Computer Science, vol. 1880, Springer, 2000, pp. 1–19.

[PR98] D. Panario and B. Richmond, *Analysis of Ben-Or's polynomial irreducibility test*, Random Structures and Algorithms **13** (1998), no. 439–456, 439–456.

[RS03] K. Rubin and A. Silverberg, *Torus-Based Cryptography*, Advances in Cryptology – CRYPTO ' 2003 (Dan Boneh, ed.), Lecture Notes in Computer Science, vol. 2729, Springer, 2003, pp. 349–365.

[Sch93] O. Schirokauer, *Discrete Logarithms and Local Units*, Philisophical Transactions of the Royal Society of London (A) **345** (1993), 409–423.

[SL93] P. J. Smith and M. J. Lennon, *LUC: A New Public Key System*, Computer Security, Proceedings of the IFIP TC11, Ninth International Conference on Information Security, IFIP/Sec '93, Toronto, Canada, 12-14 May 1993 (E. Graham Dougall, ed.), IFIP Transactions, vol. A-37, North-Holland, 1993, pp. 103–117.

[Vos91] V. E. Voskresinskiĭ, *Algebraic Groups and Their Birational Invariants*, Translations of Mathematical Monographs, vol. 179, American Mathematical Socieтy, 1991.

Institut de recherche mathématique de Rennes, Université de Rennes 1, Campus de Beaulieu, F-35042 Rennes Cedex, France.

E-mail address: `clement.dunand@univ-rennes1.fr`

DGA/CÉLAR, La Roche Marguerite, F-35174 Bruz Cedex, France.

Institut de recherche mathématique de Rennes, Université de Rennes 1, Campus de Beaulieu, F-35042 Rennes Cedex, France.

E-mail address: `reynald.lercier@m4x.org`