Factoring polynomials over an extension field

Paul Camion

INRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France

Tél. (1) 39 63 55 11

Rapports de Recherche

N° 571

FACTORING POLYNOMIALS OVER AN EXTENSION FIELD

Paul CAMION

Octobre 1986

FACTORING POLYNOMIALS OVER AN EXTENSION FIELD

Paul CAMION

ABSTRACT

Let K_0 be a field. Two irreducible polynomials $f_1(Z)$ and $f_2(Z)$ are given in $K_0[Z]$. The aim of the algorithm is to obtain all factors of $f_1(Z)$ in $K_2[Z]$, where K_2 is the field $K_0[Z]/(f_2(Z))$. Complete factorization is shown to be achieved when K_0 has characteristic zero. If K_0 is a finite field, then specific hypothesis are to be considered.

FACTORISATION DE POLYNOMES DANS UNE EXTENSION DU CORPS DES COEFFICIENTS.

RESUME

Soit K_0 un corps commutatif. On se donne deux polynomes irréductibles $f_1(Z)$ et $f_2(Z)$ dans $K_0[Z]$. Le but de l'algorithme est d'obtenir tous les facteurs de $f_1(Z)$ dans $K_2[Z]$, où K_2 est le corps $K_0[Z]/f_2(Z)$. On montre qu'une factorisation complete est obtenue lorsque la caracte'ristique de K_0 est nulle. Lorsque K_0 est un corps fini, la preuve est sujette a' des hypothe'ses restrictives.

INTRODUCTION

In D.E. KNUTH, [11], 1971 printing, an algorithm due to H. ZASSENHAUS for factoring a polynomiale $f(x)$ in $K_0[X]$, K_0 a finite field F_q is described. One essentially computes powers $(X-s)^j - 1 \bmod f(X)$, for well chosen values of j, where s is taken randomly in F_q. The method looks attractive regarding computation time, compared to E.R. BERLEKAMP algorithm [1], at least for large q.

In 1980 the author gave a counterexample to ZASSENHAUS algorithm which appeared in P. CAMION [7] and he there observed that RABIN's algorithm [13], very similar to ZASSENHAUS, avoids the failure by taking s in an extension field. However this makes the computation unusefully expensive when a linear factor is not sought. When knowing BERLEKAMP's algorithm it becomes clear that the failure can be avoided by replacing X by $g(X)$ such that $g^q(X) \equiv g(X) \bmod f(X)$ in $(X-s)^j$, $j=(q-1)/2$, for q *odd*, g and s being taken at random, s in F_q. The author then published a short description of an algorithm based upon that remark, P. CAMION [4], 1980. The same idea was developped in [5], [7] and further in the other papers quoted. Notice that an important contribution to factorization algorithms by F.S.McWILLIAMS has been included in those works as well as a method due to McELIECE [8]. Independently, D.G. CANTOR and H. ZASSENHAUS developed the same idea [10]. In the next printing of D.E. KNUTH [11] the first method of ZASSENHAUS is skipped. The factorization algorithm which is now apparently the so-called "CANTOR-ZASSENHAUS algorithm", has still a weakness. When factoring a polynomial from $F_q[X]$ into $F_{q^k}[X]$ for q even and k *odd*, one has to perform computations in $F_{q^{2k}}[X]$ which is four times more expensive ([7], page 61). That is why we investigated a general algorithm for the problem stated in title. Here we forget the exponentiation techniques and we use power series expansions of rational functions, a method that we introduced in P. CAMION [8] for constructing large irreducible polynomials. Fortunately the algorithm works here for any field K_0. Curiously the algorithm is easily shown to achieve complete factorization only when K_0 has characteristic zero or is a large finite field. Consequently the case of finite fields has to be paid special attention, which is done in section 3. There the algorithm is actually shown to achieve complete factorizations in the following situations. The polynomials f_1 and f_2 of the abstract have respective degrees d_1 and d_2. The field K_0 is a finite field F_q and f_1 and f_2 are both irreducible in $K_0[Z]$ and f_2 is primitive.

Moreover one of the following conditions should be satisfied

i) $d_1=d_2=k$

ii) $d_1=tk, d_2=k, (t, q^k-1)=1$ and k is a prime

iii) $d_1=tk, d_2=k, (t, q^k-1)=1$ and f_1 is primitive.

Outside thoses hypothesis any counterexample or proof would help improvements. For finite fields, the complexity of computation is the same as that one of comparable situations dealt with by exponentiation algorithms.

1. A USEFUL CONSTRUCTION

1.1 A HADAMARD PRODUCT OF SERIES

As told in the abstract, we will have to consider two polynomials, $f_1(Z)$ and $f_2(Z)$ with respective degrees d_1 and d_2, of the form

$$f_1(Z) = \prod_{\alpha \in H_1} (1-\alpha Z), \tag{1}$$

$$f_2(Z) = \prod_{\beta \in H_2} (1-\beta Z).$$

where the elements of $H_1 \cup H_2$ lie in *the splitting field K of* $f_1(Z)f_2(Z)$. In P.CAMION[8], we introduced an algorithm for then constructing the polynomial

$$g(Z) = \prod_{(\alpha,\beta) \in H_1 \times H_2} (1-\alpha\beta Z) \tag{2}$$

of which the most expensive operation is the gcd of two polynomials with degrees $d_1 d_2$, in the case where $(d_1, d_2) = 1$. We first recall the straightforward

Property 1

Let

$$f(Z) = \prod_{\gamma \in \Gamma} (1-\gamma Z)^{r_\gamma}. \tag{3}$$

Then we have that

$$Zf'(Z)/f(Z) = \sum_{\gamma\in\Gamma} -\gamma r_\gamma Z/(1-\gamma Z). \tag{4}$$

Then Property 1 entails

Property 2

The formal power series expansion

$$s(Z) = -Zg'(Z)/g(Z) \tag{5}$$

is the hadamard product of the power series $Zf'_1(Z)/f_1(Z)$ ***and*** $Zf'_2(Z)/f_2(Z)$.

This relies on the fact that namely

$$Zf'_1(Z)/f_1(Z) = -a_1Z - a_2Z^2 - \ldots\ldots - a_nZ^n - \ldots$$

where

$$a_n = \sum_{\alpha\in H_1} \alpha^n, \tag{6}$$

since $-\alpha - \alpha^2Z - \cdots - \alpha^nZ^{n-1} - \cdots$ is the power series expansion of the ***formal logarithmic derivative of*** $1-\alpha Z$

1.2 THE ALGORITHM FOR OBTAINING SOME FACTOR OF $g(Z)$

Since the degree of $g(Z)$ is d_1d_2 then $-Zg'(Z)$ over $g(Z)$ can be computed when the first $2d_1d_2$ coefficients of $s(Z)$ are known. This can be done by the so-called BERLEKAMP-MASSEY algorithm or by the extended EUCLID algorithm.

Expanding the first $2d_1d_2$ terms of $p(Z)/q(Z)$, namely for $p(Z)=-Zf'_i(Z)$, and $q(Z)=f_i(Z), i=1,2$, can be performed in several ways. We first have that $q(0)\neq 0$ and we may assume that $p(0)\neq 0$. Then if we denote by n_p the degree of $p(Z)$ and by n_q the degree of $q(Z)$ by $\tilde{f}(Z)$ the reciprocal polynomial of $f(Z)$, we have that for $m=2d_1d_2+n_q-n_p-1$

$$Z^m\tilde{p}(Z) = \tilde{q}(Z)\tilde{s}_0(Z) + u(Z) \tag{7}$$

where $deg\ u(Z)<n_q$ and $\tilde{s}_0(Z)$ is the reciprocal of the expected $s_0(Z)$ consisting in the first $2d_1d_2$ terms of (5).

Replacing Z by $1/Z$ in (7) and multiplying both sides by Z^{m_0} where $m_0=2d_1d_2+n_q-1$ shows that

$$p(Z)\equiv q(Z)s_0(Z)\ mod\ Z^{2d_1d_2} \tag{8}$$

1.3 THE OBTAINED $g^*(Z)$

By Berlekamp-Massey algorithm, what we obtain is a pair of polynomials $p(Z)$ and $q(Z)$. But $q(Z)$ is not necessarily $g(Z)$. It may be a proper factor of $g(Z)$. Denoting by $g^*(Z)$ the obtained polynomials $q(Z)$, then certainly $g^*(Z)$ has no multiple roots as a consequence of the properties of the formal logarithmic derivative (Property 1).

1.4 THE ALGORITHM

The subsequent steps in the algorithm of factorization consist in

- computing in $K_0[Z]$ the irreducible factors $g_1(Z),...,g_k(Z)$ of $g^*(Z)$, by any available algorithm For finite fields we refer to P. Camion [4], [5], [6], [7], [8]. See also section 3.3.1 for a gain on complexity.

- computing $gcd(g_i(XY),f_1(X))$ in $K_2[X]$ where $K_2 = K_0[Y]/(f_2(Y))$ for some $i = 1,...,k$.

In section 2, the conditions under which an irreducible factor of $f_1(X)$ in $K_2[X]$ is obtained by the described algorithm are precisely given by theorem 1.

When K_0 is a finite field, all irreducible factors are obtained from one of them, by conjugating the coefficients in K_2. The algorithm was implemented on MACSYMA and we could not find an example where it would fail. In section 3 the case of finite fields is investigated.

2. SOME ALGEBRAIC PROPERTIES

2.1 THE LAGRANGE TRANSFORM

Let K_0 *be a field* and f_1 *a polynomial* in $K_0[Z]$, i=1,2, without multiple root.

We *denote by* B *the algebra* $K_0[X,Y]/(f_1(X),f_2(Y))$. As in P. CAMION [4], we will use an explicit representation of B into a product of fields. The *set of roots of* f_i *is denoted by* H_i, i = 1,2. Let H *be the product* $H_1 \times H_2$. The splitting field of the product $f_1(Z)f_2(Z)$ *is denoted by* K.

For f in B and for $h=(\alpha,\beta)$ in H, we denote by $f(h)$ the field element $f(\alpha,\beta)$.

We usually denote by f the unique polynomial $f(X,Y)$ of degree less than d_1 in X and d_2 in Y in its residue class of $\mathcal{B}$, *where* d_i *is the degree of* f_i, i = 1,2.

We first have the straightforward

Property 1 *Let f be an element in B. Then $f=0$ iff $f(h)=0$ for every h in H.*

We then recall

Proposition 1 : *The Lagrange transform $f: B \to K^H$ defined by $f \to f = (f(h))$, $h \in H$ is a isomorphism of the K_0-algebra B into the K_0 - algebra K^H.*

Since the mapping defined by $f \to f(h)$ is an homomorphism of B into K for any h in H. then proposition 1 proceeds from property 1. □

2.2 A COMMUTATIVE DIAGRAM

We now consider the polynomial

$$g(Z) = \prod_{(\alpha,\beta)\in H_1\times H_2} (1-\alpha\beta Z), \tag{1}$$

and we denote by A the algebra $K_0[Z]/(g(Z))$. Moreover, let us denote by Γ the set

$$\{\gamma | \gamma=\alpha\beta, (\alpha,\beta)\in H_1\times H_2\}. \tag{2}$$

We then have the

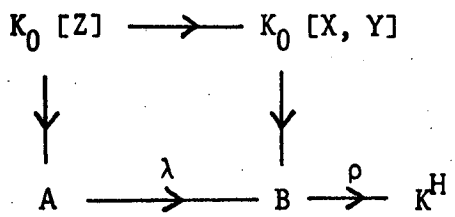
Proposition 2 : *Substituting XY for Z in all polynomials $a(Z)$ in $K_0[Z]$ defines a mapping λ from A into B. We have that λ is a K_0-algebra-homomorphism from A into B. The kernel of λ is the ideal $(g_0(Z))$ of A, where*

$$g_0(Z) = \prod_{\gamma\in\Gamma}(1-\gamma Z). \tag{3}$$

Clearly, $a(Z) \rightarrow a(XY)$ defines a mapping from $K_0[Z]$ into $K_0[X,Y]$ which is an algebra-homomorphism. We see that a residue class of A is mapped into a residue class of B. Indeed if $a(Z)$ and $b(Z)$ are congruent in A, i.e. $a(Z)-b(Z)=g(Z)c(Z)$, then $g(XY)c(XY)$ is in the zero residue class of B, by property 1. □

Now the kernel of λ is a principal ideal $(g_0(Z))$ of A where $g_0(Z)$ is the polynomial with smallest degree such that $g_0(\alpha\beta) = 0$ for very couple $(\alpha,\beta)\in H$.

The diagram



is commutative. By Proposition 2, the diagram is well defined. Now by Proposition 1 it remains to observe that all $a(Z)$ in $K_0[Z]$ and for all (α,β) in H, substituting $\alpha\beta$ to Z in $a(Z)$ gives the same result as substituting XY to Z in $a(Z)$ and then substituting α to X and β to Y.

2.3 THE ORTHOGONAL SET OF PRIMITIVE IDEMPOTENTS OF B

2.3.1 THE ORBITS OF THE GALOIS GROUP G OF K OVER K_0 ACTING ON H.

Clearly G acts as a permutation group on the set H_1 as well as on the set H_2. Hence G acts on H and we have that

$$\forall\sigma\in G\ .\ \forall h = (\alpha\ ,\ \beta)\in H\ .\ \sigma h = (\sigma\alpha\ ,\ \sigma\beta). \qquad (4)$$

The orbits of G on H are denoted $O_1, O_2, \ldots O_s$.

Proposition 3 : *The number of primitive idempotents of B equals the number s of orbits of G on H. These idempotents are*

$$e_i(X,Y) = \sum_{(\zeta,\xi)\in O_i} \prod_{\substack{(\alpha,\beta)\in H \\ (\alpha,\beta)\neq(\zeta,\xi)}} (X-\alpha)(Y-\beta)/(\zeta-\alpha)(\xi-\beta)\ ,\ i = 1,\ldots,s. \qquad (5)$$

An element in K^H is an idempotent iff it is non zero and every component x verifies in $K: x^2 = x$ or equivalently if each of its components is 1 or 0.Since we have that

$$e_i(\zeta,\xi) = \delta_{ij}$$

as (ζ,ξ) runs over $O_j, i,j = 1,\ldots,s$. $e_i(X,Y)$ is an idempotent, by Proposition 1. (Its coefficients are fixed under G, thus $e_i(X,Y)$ is in B). Now assume that e_i is a sum of two orthogonal idempotents ε_1 and ε_2. We would have $S_1 \cap S_2 = 0$ and $S_1 \cup S_2 \subset O_i$, where S_1 and S_2 are the subsets of H on which ε_1 and ε_2 don't cancel, respectively. But for any element a in B, if $a(\zeta,\xi) \neq 0$ for $(\zeta,\xi)\in O_i$ then $a(\zeta,\xi)$ cannot vanish on any element in O_i. □

2.3.2 REMARK 1 : Similar constructions as for e_i show that the inverse of a unit u of K^H lying in ρB is itself in ρB.

2.4 MAXIMAL IDEALS OF A WHICH ARE MAPPED BY λ ONTO MAXIMAL IDEALS OF B.

2.4.1 THE BASIC THEOREM

We first have

Proposition 4 : *An ideal (a) of B is maximal if there exist an orbit O_i such that a cancels on every (α,β) of O_i and nowhere else.*

We first know that B is a principal ideal ring since it is a semi-simple ring. There certainly exists a primitive idempotent e_i which does not belong to (a)

so long as a is not a unit. In particular if (a) is a maximal ideal, for such an idempotent e_i we have that

$$(a)+(e_i) = B, \tag{6}$$

or

$$au+e_iv = 1, \tag{7}$$

for some u, v in B.

This shows that a cannot cancel outside 0_i. Moreover if there was an (α,β) in 0_i for which $a(\alpha,\beta)\neq 0$, then a would be a unit in B since ρa is a unit in $\rho B \subset K^H$, by the above remark. Conversely if a only cancels on 0_i, then it is easily seen that $(a) = \sum_{j\neq i}(e_j) = (\sum_{j\neq i} e_j)$ which is a maximal ideal of B. □

For a better description of representation ρ of B in K^H we recall

Proposition 5 : *Algebra B is the direct sum of its minimal ideals e_iB. The dimension of e_iB over K_0 is the length of the orbit 0_i. $i = 1, \ldots, s$.*

These properties where already used in P.CAMION [9]. We have that $dim_{K_0}B = |H|$, since a basis for B is $\{X^iY^j\}_{i<d_1, j<d_2}$. But also

$$dim_{K_0}B = \sum_{1\leq i\leq s} dim_{K_0}e_iB. \tag{8}$$

We are left with proving that $dim_{K_0}e_iB \geq |0_i|$. But the ideal e_iB, considered as a ring, is a field K' and we just have to show that the degree of the extension of K' over K_0 is at least $|0_i|$. We recall the line of the argument in P. CAMION [9]. First $a \to a(h)$ for a fixed h in 0_i, as a runs over e_iB, defines a field isomorphism of e_iB onto K'. On the other hand every couple h, h' in 0_i defines a mapping by $a(h) \to a(h')$ which is a field automorphism over K_0. It is then seen that $|0_i|$ distinct field automorphism over K_0 are exhibited in that way, which shows by ARTIN's Lemma that $dim_{K_0}K' \geq |0_i|$. □

We are now in a position to investigate the maximal ideals of A which are mapped by λ onto maximal ideals of B.

By (1), (2) and (3), we see that γ is a multiple root of $g(Z)$ whenever there are two distinct (α,β) and (α',β') in $H_1 \times H_2$ with $\gamma = \alpha\beta = \alpha'\beta'$.

For the algorithm in view, the following statement is the most informative.

Theorem 1 : *Let $f_1(Z)$ and $f_2(Z)$ be polynomials without multiple roots in $K_0[Z]$ and $g(Z)$ be defined as in 2 1. Also*

$$A = K_0[Z]/(g(Z)),$$

$$B = K_0[X,Y]/(f_1(X), f_2(Y))$$

and $\lambda: A \to B$ is defined by $a(Z) \to a(XY)$. We have that a maximal ideal $(d(Z))$ of A maps onto a maximal ideal $(\lambda d(Z))$ of B iff for γ a root of $d(Z)$, whenever (α,β) and (α',β') in $H_1 \times H_2$ are such that $\gamma = \alpha\beta = \alpha'\beta'$, then there exist an automorphism σ of the splitting field K of $f_1(Z)f_2(Z)$ over K_0 such that $\alpha' = \sigma\alpha$ and $\beta' = \sigma\beta$.

Let γ and γ' be any two roots of the irreducible polynomial $d(Z)$. Then $\gamma = \alpha\beta$ and $\gamma' = \alpha'\beta'$ for $(\alpha,\beta),(\alpha',\beta')$ in H. But there exist an automorphism σ of the Galois group G' of the splitting field K' of $g(Z)$ over K_0 that maps γ onto γ'. Let 0_i be the orbit of H under G to which (α,β) belongs. Then $(\sigma\alpha,\sigma\beta)$ is in 0_i and since $\alpha'\beta' = \sigma\alpha.\sigma\beta = \gamma'$, we have that (α',β') is in 0_i, by hypothesis. Proposition 4 completes the proof. □

Corollary 1 : *If γ is a simple root of $g(Z)$ then its minimal polynomial $d(Z)$ in $K_0[Z]$ generates an ideal of A which is mapped by λ onto a maximal ideal of A.*

Every root of $d(Z)$ here is a simple root of $g(Z)$. In the preceeding argument when we observe that $\alpha'\beta' = \sigma\alpha.\sigma\beta = \gamma'$ we may conclude that $\alpha' = \sigma\alpha$ and $\beta' = \sigma\beta$ straight away and get the same conclusion. □

Corollary 2 *Assume that $f_1(Z)$ and $f_2(Z)$ are both irreducible in $K_0[Z]$. Denote $K_0[Z]/(f_i(Z))$ by K_i, $i=1,2$. If $g(Z)$ is irreducible in $K_0[Z]$ then $f_1(Z)$ is irreducible in $K_2[Z]$ and $f_2(Z)$ is irreducible in $K_1[Z]$.*

For finite fields we will next prove a converse to corollary 2.

This is actually entailed by Proposition 2. We have that A is a field and λ maps isomorphically A onto B. But B is isomorphic to both $K_2[X]/(f_1(X))$ and $K_1[Y]/(f_2(Y))$. □

Examples :

1) - We take $K_0 = F_2$, $f_1(Z) = Z^3+Z+1$ and $f_2(Z) = Z^3+Z^2+1$. Then $g(Z) = (Z+1)^3(Z^3+Z+1)(Z^3+Z^2+1)$. If α is a root of Z^3+Z+1, then the roots of $f_1(Z)$ are α,α^2,α^4 and those of $f_2(Z)$ are $\alpha^6,\alpha^5,\alpha^3$. The Galois group of the splitting field $K = F_8$ of $(Z^7+1)/(Z+1)$ over F_2 has three orbits on H. One of the orbits is $\{(\alpha,\alpha^6),(\alpha^2,\alpha^5),(\alpha^4,\alpha^3)\}$. It corresponds to the maximal ideal $(Z+1)$ of A which maps onto the maximal ideal $(XY+1)$ of B. We then must have that $gcd(XY+1,X^3+X+1)$ is an irreducible factor of X^3+X+1 in $K_2[X]$. But here $K_2 = F_8$ and the canonical map of B onto $K_2[X]/(f_1(X))$ sends Y onto a root, say α^6, of $f_2(Y)$. We thus obtain $X+Y^{-1} = X+\alpha$ as a factor of $f_1(X)$.

2) - We take $K_0 = \mathbf{Q}$, $f_1(Z) = \Phi_5(Z)$ and $f_2(Z) = \Phi_{10}(Z)$. If ζ denotes a primitive tenth root of unity, then the roots of $\Phi_{10}(Z)$ are $\zeta,\zeta^3,\zeta^9,\zeta^7$ and those of $\Phi_5(Z)$, are $\zeta^2,\zeta^6,\zeta^8,\zeta^4$. The Galois group of $\mathbf{Q}(\zeta)$ over $\mathbf{Q}$ is the group $\{1,3,9,7\}$ of units, of Z/mZ , $m = 10$. One sees that H has four orbits under G and $g(Z) = (Z+1)^4\varphi_{10}^3(Z)$. We then have that $gcd(XY+1,\Phi_5(X)) = X+Y^{-1}$. We then get in that way a linear factor of $\Phi_5(X)$ in $K_2[X]$. However, it is easily seen that $\Phi_{10}(XY)$ is a primitive idempotent of B. Hence the maximal ideal $(\Phi_{10}(Z))$ of A maps onto the minimal ideal $(\Phi_{10}(XY))$ of B.

3) - We take $K_0 = \mathbf{Q}$, $f_1(Z) = Z^4 + 1$ and $F_2(Z) = Z^2 + 1$. The algorithm fails to factor $Z^4 + 1$ as $(Z^2 + i)(Z^2 - i)$. For, the polynomial $g(Z)$ here is $(Z^4 + 1)^2$.

Corollary 3 *If K_0 has characteristic zero, then $f_2(Z)$ can be changed to $f_2(Z-c)$ for some well chosen c in K_0 in order that $g(Z)$ has only simple roots*

and the algorithm always achieves a complete factorization.

Consider the polynomial

$$p(X) = \prod_{\substack{(\alpha,\beta,\eta,\vartheta)\in H\times H \\ (\alpha,\beta)\neq(\eta,\vartheta)}} (\alpha(X+\beta) - \eta(X+\vartheta)),$$

and let c be in K_0, $p(c)\neq 0$. Such a c exists since K_0 is infinite. Now going back to 1.4, one sees that the polynomials $gcd(g_i(XY), f_1(X))$ in $K_2[Z]$ $i = 1,\ldots,k$ are all irreducible factors of $f_1(X)$ in $K_2[X]$, by Theorem 1, Corollary 1.

As an example of application, take $f_1 = Z^4 + 1$ and $f_2 = Z^2 + 1$. For $c = -1$ we have that f_2 becomes $Z^2 + 2Z + 2$. The algorithm then actually produces $Z^4 + 1 = (Z^2 - y - 1)(Z^2 + y + 1)$ where y being a root of the new f_2, we have that $y = i - 1$.

Now let ϑ_1 be a root of $f_1(Z)$ and ϑ_2 a root of $f_2(Z)$. The algorithm also tells if $K_0(\vartheta_1)$ is contained in $K_0(\vartheta_2)$. This occurs iff $f_1(X)$ has a linear factor in $K_2[X]$ for $K_2 = K_0(\vartheta_2)$. Such an example is dealt with as an application of the algorithm at the end of this paper. We there consider $f_1(Z) = Z^4 + 5Z^2 + 4Z + 1$ which has the root $\Theta_1 = \Theta_2 + \Theta_2^2$, where Θ_2 is a root of $f_2(Z) = Z^4 + Z + 1$. All computations up to obtaining $g(Z)$ are performed *mod* p for p = 9999999967 to avoid huge integers.

Clearly, Corollary 3 also applies to finite fields whenever $|K_0|$ is larger than the product $d_1 d_2$. Thus there is a real problem for justifying the algorithm only if the size of K_0 is small. That problem is investigated in section 3.

2.4.2 THE CASE OF FINITE FIELDS

Let K_0 be the finite field F_q. Denote by F_{q_1} and F_{q_2}, $q_1 = q^{d_1}$ and $q_2 = q^{d_2}$, two subfields of a finite field. Denoting (d_1,d_2) by d, we know that $F_{q_1} \cap F_{q_2} = F_{q^d}$. This is a straight consequence of the fact that $(q^{d_1} - 1, q^{d_2} - 1) = q^d - 1$.

We have the

Corollary 4 *Let $f_1(Z)$ and $f_2(Z)$ be defined as in Corollary 2. Then $g(Z)$ is irreducible in $K_0[Z]$ iff $(d_1,d_2) = 1$*

If $g(Z)$ is irreducible in $K_0[Z]$, then by Corollary 2, $f_1(Z)$ is irreducible in $K_2[Z]$. We then know that $K_1 \cap K_2 = K_0 = F_q$. (see also section 2.5). Hence $(d_1,d_2) = 1$. Conversely, consider the permutation τ on $H_1 \times H_2$ defined by $\tau(\alpha,\beta) = (\tau\alpha,\tau\beta) = (\alpha^q,\beta^q)$. Since $(d_1,d_2) = 1$ then τ is made of a single cycle of length d_1d_2. We must now show that for $\alpha\beta = (\tau^i\alpha)(\tau^i\beta), i>0$, then we have that $i = d_1d_2$. Necessarily the element $\alpha^{-1}\tau^i\alpha$ is in $K_1 \cap K_2 = F_q$. Denote permutation τ^i by μ. Then $\mu\alpha = k\alpha, k \in F_q$. Since μ is a Galois automorphism of K_1 over K_0, then there exist a smallest j such that $\alpha = \mu^j\alpha = k^j\alpha$. We have that j is the order of k in F_q and also divides the order d_1 of the Galois group of K_1 over K_0. But similarly, we see that j also divides the order d_2 of the Galois group of K_2 over K_0. Thus $j = 1$. This entails that $k = 1$ and i is a multiple of d_1d_2. □

Remark If an irreducible polynomial $v(Z)$ in $F_q[Z]$ is such that $v(\vartheta) = v(k\vartheta) = 0$ for $1 \neq k \in F_q$, then it is easily shown that $v(Z) = u(Z^j)$ where j is the order of k in F_q. The remark in section 3.2.2. deals with such an example.

2.5 NUMBER OF ORBITS AND NUMBER OF FACTORS

With the notation introduced in section 1.1, 2.1, 2 3.1 and in the statement of Corollary2 of this section, we prove the following

Theorem 2 *Assume that both $f_1(Z)$ and $f_2(Z)$ are irreducible over $K_0[Z]$. If moreover $f_1(Z)$ splits over $K_1[Z]$, then the number s of factor of $f_1(Z)$ over $K_2[Z]$ is given by (11) and s is the number of factors of $f_2(Z)$ over $K_1[Z]$ as well. Further, the dimension of $K_0(\alpha,\beta)$ over K_0 does not depend on a particular choice of (α,β) in $H_1 \times H_2$.* We have the relation

$$|G_{(\alpha,\beta)}|\,|(\alpha,\beta)^G| = |G|, \qquad (9)$$

where (α,β) is an element in H and $G_{(\alpha,\beta)}$ denotes the stabilizer of (α,β) in G. Moreover $(\alpha,\beta)^G$ denotes the orbit of H under G to which (α,β) belongs. We recall our previous notations $K_1 = K_0[X]/(f_1(X))$ and $K_2 = K_0[Y]/(f_2(Y))$. We first observe that

$$G_{(\alpha,\beta)} = G_\alpha \cap G_\beta. \tag{10}$$

where α is a root of $f_1(X)$ and β a root of $f_2(Y)$. We first observe that for any two members (α,β) and (α',β') in $H_1 \times H_2$ we have that $G_{(\alpha,\beta)}$ is a conjugate of $G_{(\alpha',\beta')}$ in G. For, there exists a σ in G such that $\sigma\beta = \beta'$ and therefore

$$\sigma^{-1} G_{(\alpha,\beta)^\sigma} = G_{(\sigma\alpha,\beta')} = \sigma^{-1} G_\alpha \sigma \cap G_{\beta'} = G_\alpha \cap G_{\beta'} = G_{\alpha'} \cap G_{\beta'},$$

since G_α is an invariant subgroup of G, by hypothesis. This shows that all orbits $(\alpha,\beta)^G$ have the same length. This also shows that the dimension of $K_0(\alpha,\beta)$ over K_0 does not depend on a particular choice of (α,β) in H. Indeed that dimension is the index of of $G_{(\alpha,\beta)}$ in G, since $G_{(\alpha,\beta)}$ is the Galois group of K over $K_0(\alpha,\beta)$. By Proposition 5, the common length of the orbits is the common dimension of a minimal ideal of B. It also is the product of $dim_{K_0} K_1 = d_1$(resp $dim_{K_0} K_2 = d_2$) by the degree of each irreducible factor of $f_2(Y)$ in $K_1[Y]$ (resp$f_1(X) in K_2[X]$). ***Consequently the number s of factors in both cases is***

$$s = d_1 d_2 |G_\alpha \cap G_\beta| / |G| \tag{11}$$

and the degree of a factor of $f_1(X)$ is d_1/s (resp of $f_2(Y)$ is d_2/s). □

In particular, we see that if $gcd(d_1 d_2) = 1$, then necessarily $s = 1$, which entails that $g(Z)$ is irreducible in $K_0[Z]$. By (11) we see that for finite fields, we always have that $s = gcd(d_1 d_2)$. This is not true in general since $\Phi_9(Z)$ has the same degree as $\Phi_7(Z)$ and the irreducible polynomial $g(Z)$ here is

$$Z^{36} - Z^{33} + Z^{27} - Z^{24} + Z^{18} - Z^{12} + Z^9 - Z^3 + 1,$$

that is $\Phi_{63}(Z)$.

3. THE ALGORITHM FOR K_0 A FINITE FIELD F_q.

3.1 THE CASES TO BE CONSIDERED AT THE LIGHT OF THEOREM 1

A general problem here to be submitted to the algorithm is the following. Given two irreducible polynomials $f_1(Z)$ and $f_2(Z)$ in $K_0[Z]$ with respective degrees d_1 and d_2,$(d_1,d_2) = k$, find all factors of $f_1(Z)$ in $F_{q^k}[Z]$, namely k irreductible factors as mentionned in section 2.5. Each factor has degree d_1/k and the coefficients of two terms with the same degree in any two factors are conjugated under the Galois group of F_{q^k} over F_q. Hence we just ask the algorithm to produce one of the irreducible factors of $f_1(Z)$ sought.

The coefficients in F_{q^k} are obtained as elements in $K_2 = F_q[Y]/(f_2(Y))$.

All such general factorization problems dealt with by the algorithm introduced in section 1 2 were solved by means of a program made of MACSYMA Command lines. The used factor $d(Z)$ of $g(Z)$ of which the required property *i.e.* $(\lambda d(Z))$ should be a maximal ideal of B was obtained by factorizing $g(Z)$ in $K_0[Z]$ by using the MACSYMA "FACTOR" C.L. with the appropriate flag when q is a prime. The fact that $(\lambda d(Z))$ is maximal is here a straight consequence of the fact that the degree of the found factor of $f_1(Z)$ is actually d_1/k, and this never failed to be observed. Thus practically, the algorithm works all right since we would be warned if no irreducible factor $d(Z)$ of $g(Z)$ obtained by the algorithm would meet the requirement stated in Theorem 1.

However there are some natural cases where we can assert in advance that the algorithm will work efficiently. The first case is the following.

3.2 THE POLYNOMIALS f_1 AND f_2 ARE BOTH IRREDUCIBLE IN $K_0[Z]$. MOREOVER f_2 IS PRIMITIVE AND $d_1 = d_2 = k$.

3.2.1 THE HERE INTRODUCED ALGORITHM COMPARED TO RABIN'S

Here the problem could be solved by RABIN's first introduced probabilistic algorithm of factorization which would produce all linear factors of f_1[13]. However that algorithm only works for q odd. The operations are performed in $F_{q^k}[Z]$. All products of polynomials are reduced mod $f_1(Z)$ in that algorithm and since a product in F_{q^k} needs $O(k^2)$ products over F_q, every

product of polynomials needs $O(k^4)$ products over F_q for any usual algorithm of polynomial products. But $O(k \log q)$ products are needed for an exponentiation by d , $d = (q^k - 1)/2$. This leads to $O(k^5 \log q)$ products over F_q to get a factor. All other linear factors are the conjugated of the first one obtained.

In the here presented algorithm, polynomial $g^*(Z)$ of section 1.3 is first obtained by performing the g.c.d of two polynomials of degree $2k^2$ By a classical algorithm this needs $O(k^4)$ products in F_q. Now for this algorithm to be comparable with the one of RABIN, $g^*(Z)$ has to be factored over $F_q[Z]$ in $O(k^5 \log q)$ operations. This is made possible by first observing that the degree of each irreducible factor of $g^*(Z)$ in $F_q[Z]$ divides k. As a consequence we have that the algebra $A_0 = F_q[Z]/(g^*(Z))$ is isomorphic to a product of fields, each of them isomorphic to F_{q^k} or to one of its subfield. With that representation of A_0 we may define the F_q-subalgebra $\mathcal{B}$ of A_0 represented by a product of subfields isomorphic to F_q as we did in P.CAMION, [4], [5], [6], [7].

In P. CAMION [5] we recall the definition of Mc ELIECE operator T and we derive some new properties for T. That operator here maps A_0 onto $\mathcal{B}$ and it actually takes the trace from F_{q^k} onto F_q of each component of v in A_0 represented as hereabove described. Then

$$Tv = \sum_{0 \le i < k} v(X^{q^i}). \tag{1}$$

Since the degree of $g^*(Z)$ is k^2, obtaining the $X^{q^i}, i = 0,\ldots,k-1$ and from there the polynomial Tv needs $O(k^5 \log q)$ operations over F_q. Such polynomials v may be selected as required by Theorem 1 of [5]. Now Tv being obtained, two cases are to be considered.

i) q is odd

Then $(Tv)^d$ is computed in $A_0, d = (q-1)/2$. The result is the square-root of an idempotent. Added to one, the polynomial obtained generally shares a non-trivial factor w with $g^*(Z)$. That factor w or $g^*(Z)/w$ will replace $g^*(Z)$ in a next step if necessary (See [4], [5] or [9]).

ii) q **is even**

Here $q = 2^t$. We can again use Mc ELIECE operator which takes the trace of every component of the representation of Tv from F_q onto F_2. We obtain a polynomial w with the some properties as in i).

If obtaining one factor is considered as a ***main step*** in this algorithm as well as in RABIN's one, then clearly the number of main steps required to get an irreducible factor of $g^*(Z)$ in $F_q[Z]$ is essentially the same as for getting a linear factor in RABIN's algoritnm.

Finally, performing $gcd(d(XY), f_1(X))$ in $K_2[X]$ needs $O(k^4)$ operations in a classical algorithm.

Conclusion Our algorithm is comparable in execution time to RABIN's one in this particular case, help to the use of Mc ELIECE operator. However the algorithm here presented is more versatile and works the same for q even or odd except in one easy step.

3.2.2. FACTORIZATION IS ALWAY ACHIEVED

From Theorem 1 of section 2.4, we get an easy sufficient condition for the existence of a factor $d(Z)$ of $g^*(Z)$ such that $(\lambda d(Z))$ be a maximal ideal of B. All we here need is that $d(Z)$ be irreducible with degree k and that it be a simple factor of $g(Z)$. We will see that this is always true if $f_2(Z)$ is primitive. If not the hereby remarks gives a counterexample.

Remark Here is an example for K_0 a finite field $F_q, q = 3$, where the polynomial $g(Z)$ has no simple root at all. Let ϑ be primitive 80 th root of unity and $\alpha = \vartheta^5$. The polynomial $f_1(Z)$ of degree 4 has α and its conjugates as roots and $f_2(Z)$ is the reciprocal of $f_1(Z)$. Then the roots of $g(Z)$ are $\vartheta^i, i \in I$ for $I = \{0,10,30,40,50,70\}$. The 16 roots of $g(Z)$ consist in two quadruple roots 1 and -1 and four double roots. Moreover the highest degree of a factor of $g(Z)$ is two.

The Proof.

Let ϑ be a primitive element in F_{q^k}. We recall the

Property. *The integer c is such that ϑ^c is a root of an irreducible polynomial of degree k in $F_q[Z]$ iff $q^w c \equiv c \bmod (q^k-1)$ as long as $w \equiv 0 \bmod k$.*

On the other hand, we know that ϑ^b is primitive iff $(b,m)=1$ for $m = q^k-1$. Now let ϑ^a be a root of $f_2(Z)$, $(b,m)=1$, and ϑ^c a root of $f_1(Z)$. Then each factor of $g(Z)$ has a root of the form ϑ^a for $a \equiv b + q^r c \bmod m$, for some r.

Lemma 1 of section 4 proves that the minimal polynomial $d(Z)$ of such a ϑ^a firstly has degree k (j must equal 0) and secondly has multiplicity one (i must equal r) for at least one integer r in $[0,k[$.

3.3. THE POLYNOMICALS f_1 AND f_2 ARE BOTH IRREDUCIBLE IN $K_0[Z]$. MOREOVER f_2 IS PRIMITIVE AND d_2 DIVIDES d_1.

In the following we denote d_1 by tk and d_2 by k.

3.3.1 COMPARISON OF THE ALGORITHM WITH AN EXPONENTIATION ALGORITHM AS P.CAMION'S [4].

In the quoted algorithm, the algebra $A^0 = K_2[Z]/(f_1(Z))$ is to be considered, where $K_2 = K_0[Z]/(f_2(Z)) - F_{q^k}$. Then A^0 is isomorphic to a product of fields, precisely of k fields, each one isomorphic to $F_{q^{tk}}$. What we called BERLEKAMP subalgebra $\mathcal{B}$ of A^0 in [4] is the subalgebra of A^0 isomorphic to a product of subfields F_{q^k} in the hereabove representation. Constructing a basis of $\mathcal{B}$ takes $O(t^3k^3)$ operations over F_q. Such a basis is made of k polynomials in $F_q[Z]$. Then a polynomial $v(Z)$ is chosen randomly in $\mathcal{B}$ and raised to the power d where $d = (q^k-1)/2$ for q odd and $d = (q^k-1)/3$ for q even and k even (The case where q is even and k odd was dealt with in [7], page 61). This needs $O(t^2k^5 \log q)$ operations over F_q, since $v(Z)$ is actually in $F_{q^k}[Z]$, as a linear combination over F_{q^k} of the polynomials in the basis of $\mathcal{B}$.

In the present algorithm, A^0 is replaced by A_0 as in 3.2 while polynomial $g(Z)$ here has degree tk^2. However A_0 is again a product of fields isomorphic to $F_{q^{tk}}$ and Mc ELIECE operator yields a trace from $F_{q^{tk}}$ onto F_{q^k}. Nevertheless,

if we must work in A_0 for finding polynomials in $\mathcal{B}$ we will need more computer time and larger memory space since $deg\ g(Z) = kdeg\ f_1(Z)$. Fortunately working with Mc ELIECE operator in A^0 yields polynomials which actually lie in the BERLEKAMP subspace of A_0. For, let v be any polynomial in A^0 and

$$u = \mathcal{T}v = \sum_{0 \le i < t} v(Z^{q^{ki}}) \in A^0.$$

All we have to show is that u being considered in A_0, the LAGRANGE transform of λu has its components in F_{q^k}. We have that for $(\alpha,\beta) \in H$.

$$u(\alpha\beta) = \sum_{0 \le i < t} v((\alpha\beta)^{q^{ki}}) = \beta \sum_{0 \le i < t} v(\alpha^{q^{ki}}) \in F_q k,$$

since β is in F_{q^k} and v is in the BERLEKAMP space of A^0. We show that the cost of constructing such a u is $O(t^3k^3)$, provided v is taken from $A^0 \cap F_q[Z]$. For, we first compute $Z^{q^k}\ mod\ f_1(Z)$. The other polynomials needed in A^0 are

$$Z^{iq^k} \equiv \sum_{0 \le j < tk} a_{ij} Z^j\ mod\ f_1(Z), i = 0,\ldots,tk-1 \qquad (2)$$

This needs tk operations over A^0, i.e. $0(t^3k^3)$ operations over F_q as for BERLEKAMP's algorithm. Next, we compute $\mathcal{T}v = \sum_{0 \le j < t} v(Z^{q^{kj}})$ by shifting t times polynomials by the substitution $X \leftarrow X^{q^k}$ and performing for each shift a linear combination of the rows of matrix (a_{ij}). This needs $O(t^3k^2)$ operations over F_q. After computing $\mathcal{T}v$ which is now considered in A_0, step i) or ii) of 3.2.1 is applied with appropriate value of d. This needs $O(t^2k^5 \log q)$ operations. Since we only need one irreducible factor of $g(Z)$ then that step is repeated at most $\log_2 k$ times. Moreover the smallest factor obtained, say $d(Z)$, replaces the previous one (the first is $g^*(Z)$) at the end of each such step. The polynomials X^{iq^k} are reduced $mod\ d(Z)$.

Conclusion When a polynomial $f_1(Z)$ is to be factored over $F_{q^k}[Z]$ but actually lies in $F_q[Z]$, the here presented algorithm is comparable to the exponentia-

tion algorithm of P CAMION [4]. The algorithm is programmed for any couple of degrees d_1 and d_2 and need not be specialized for $d_1 = tk$ and $d_2 = k$. An application to factoring any polymomial in $F_{q^k}[Z]$ for $q = 2$ and k even may also be derived. The algorithm doesn t care of q being even or odd except in one easy step. The coefficients of all polynomials concerned lie in te ground field F_q.

However this algorithm needs some further theoretical investigations. The following discussion in section 3.3.2 and the remark after Lemma 2 prove that if f_1 is primitive as f_2 is and if $(t, q^k-1) = 1$ then the algorithm always achieves a complete factorization of $f_1(X)$ in $K_2[X], K_2 = K_0[Y]/(f_2(Y))$. Moreover, we show in particular that when k is a prime, f_1 need not be primitive.

3.3.2 DOES THE ALGORITHM ALWAYS ACHIEVE A COMPLETE FACTORIZATION ?

As told in 3 1 the algorithm never failed so far. However some hypothesis on the integers d_1 and d_2 are needed for the proofs given here. We here show that under the hypothesis stated for this section 3.3, if k is a prime, and if no prime factor of t divides q^k-1, then the algorithm always produces an irreducible factor of f_1 in $F_{q^k}[Z]$ where F_{q^k} is given by $F_q[Z]/(f_2(Z))$. Remember however that the algorithm always factors whenever q is larger than $d_1 d_2$ (section 2.4.1 last paragraph).

The arithmetic condition is verified in particular whenever for every prime factor p of t one has :

$$q \not\equiv 1 \ mod \ p \text{ and } p \not\equiv 1 \ mod \ k. \qquad (2)$$

This is true in particular if $q \leq p \leq k$ for every prime factor p of t or if $q=2$ whenever t is odd and $p \leq k$ Now assume k is a product of primes, $k = k_1 k_2 \cdots k_s$, $k_1 \geq k_2 \geq \cdots \geq k_s$. Moreover we have that $q \not\equiv 1 mod p$, $q \not\equiv 1 mod k_i$ and $p \not\equiv 1 mod k_i, i = 1, \ldots, s$ for every prime factor p of t. Then at least we can prove that s applications of the algorithm will produce an irreducible factor of f_1 in $F_{q^k}[Z]$. For, we first take as f_2 a primitive polynomial of degree k_1

and we obtain a factor of degree tk/k_1 of f_1 over $F_{q^{k_1}}[Z]$. Indeed condition (2) is verified since $k_j \leq k_1, j = 2,\ldots,s$. Next, all conditions required are met for another application where F_q is replaced by $F_{q^{k_1}}$ and f_2 is now a primitive polynomial in $F_{q^{k_1}}[Z]$. Finally we will obtain an irreducible factor of f_1 in $F_{q^k}[Z]$ as demanded.

The reason why the algorithm achives the factorization is only slightly different from that one explained in section 3.2.2. Here ϑ will denote a primitive element in $F_{q^{kt}}$. Then with the notations $n = q^{kt}-1, m = q^k-1$ and $dm = n$, we have that ϑ^d is a primitive element of F_{q^k}. And a primitive polynomial f_2 in $F_q[Z]$ of prime degree k is the minimal polynomial of ϑ^d for some such ϑ. Now in the Property stated in 3.2.2, we replace k by tk and c by s to characterize the irreducible polynomial f_1 in $F_q[Z]$, minimal polynomial of ϑ^s. Then each factor of $g(Z)$ has a root of the form ϑ^a for $a \equiv s + q^r d \bmod n$, for some r, since each root of $g(Z)$ is the product of a root of $f_1(Z)$ and one of $f_2(Z)$. Finally, Lemma 2 of section 4 proves that the minimal polynomial $d(Z)$ of such a ϑ^a firstly has degree kt (j must equal 0) and secondly has multiplicity one (i must equal r) for at least one integer r in $[0,k[$.

4. SOME ARITHMETIC PROPERTIES

Before stating our first Lemma, we recall the easy

Property 1 *One has* $q^i \equiv q^j \bmod (q^k-1)$ *iff* $i \equiv j \bmod k$

Lemma 1 *Denote* q^k-1 *by* m *and let* q, k *and* b *be integers such that* $(b,m) = 1$ *for* $m = q^k-1$. *Moreover let* c *be an integer such that* $q^w c \equiv c \bmod m$ *only if* $w \equiv 0 \bmod k$. *Then there exist an integer* r *such that the equation for* i *and* j :

$$b + q^r c \equiv q^j (b + q^i c) \bmod m, \qquad (1)$$

has no other solution that $j = 0$ *and* $i = r$.

Denoting by a the product of c by the inverse of b *mod* m, then equation (1) can be replaced by

$$1 + q^r a \equiv q^j (1 + q^i a) \ mod \ m \qquad (2)$$

We will show that taking for r the integer in $[0,k[$ such that the remainder of the division of $q^r a$ by m be the smallest possible, then equation (2) has for only solution $j = 0$ and $i = r$.

We first introduce some usefull notations. We denote the set $[0,q[$ by K_0 and we identify $c = c_0 c_1 \cdots c_{k-1} \in K_0^k$ with the integer $c_0 + c_1 q + \cdots + c_{k-1} q^{k-1}$ in $[0,q^k[$. This is convenient because multiflying the integer c in $[0,q^k[$ by q^r corresponds to transiating the set of positions of c by $-r$ *mod* k. Thus $q^r c$, as an element of K_0^k, is $c_{-r} c_{-r+1} \cdots c_0 c_1 \cdots c_{k-r-1}$. There is no loss in generality by taking for a the smallest integer among all residues *mod* m of $q^r a$, $r = 0,\ldots,k-1$. We will refer subsequently to that property by just saying tnat a *is minimum*. Thus we make $r = 0$ in (2). We will now consider two cases.

i) $a \equiv q-1 \ mod \ q$

Together with that property, then a considered in K_0^k should have the longest among ali possible runs of zero's for its last components. Let s be the length of that run of zero's. If for example $s = 0$ then none of the components of a is zero. Necessarily, $a + 1$ has exactly one run of s' zero's, $s'>s$, since $a_0 = q-1$. Thus there is only one j such that (2) holds for a given i. Moreover $1 + q^i a$ must have a run of s' zero's and thatfor a_{-i} must equal $q-1$ and that a_{-i} must take place immediatly after a run of s zero's. Since the two runs of s' zero's must coincide after the permutation yielded by q^j, then $a_0 + 1$ must coincide with $a_{-i} + 1$. This entails that $j = 0$. But then by (2), $a \equiv q^i a \ mod \ m$, and multiplying both members by b, we get $c = q^i c \ mod \ m$. Hence by hypothesis $i = 0$.

ii) $a \not\equiv q-1 \ mod \ q$

We still assume that a is minimum. We restate equation (2) as

$$q^u(i + a) \equiv q^v + a \ mod \ m. \tag{3}$$

with $u = -j-i$ and $v = -i$ We first observe that $a_0 = a_v < q-1$. We thus prove that there is no other solution to (3) than $u = v = 0$. Else we first consider the case where $(u,k) = 1$. Thus to every integer $v<k$ there corresponds a smallest w such that

$$wu = dk + v. \tag{4}$$

This means that iterating a translation $mod\ k$ by $-u$ a number of times equal to w yields a translation by $-v$.

Let x be the value of a_0. We have that $w = 1$ iff $v = u$ by (4), and (3) entails that $q^u a \equiv a \ mod \ m$ which by the hypothesis on c is only possible if $u = 0, v = 0$. Else $a_0 + 1 = a_u = x + 1$ which is denoted by y. For all l less than w then $a_{lu} = y$. Then necessarily $a_{(w-1)u} = a_v + 1$ Thus $a_{wu} = a_v = x$. Further as long as l is less than k we have that $a_{lu} = a_{(l-1)u} = x$ since only the v^{th} component of $q^v + a$ differs from its homologous in a. Finally we have that $a_0 = a_{(k-1)u} = x$.

This last remark shows that we must have $v>0, v<k$ if $w>1$. But this is impossible since $a_0 = x<y$ and a is minimum.

Next we consider the case where $(u,k) = t<k$. We then write $u = tu'$ and $k = tk'$. Hence $(u',k') = 1$ and the case is reduced to the previous one when considering the involved integers in the radix $q' = q^t$ in place of q. A slight difference however appears in the proof. The integer $a + q^v$ is not necessarily obtained from a by encreasing its component a_v by one.

First the concerned component in the radix q' is a_g where $g = [v/t]$ and it is seen that $a_g = a_{g-u} - q^{v'}$ where $v' = v - gt$. We write this as $y - q^{v'} = z$ which is worth x in the only case where $v' = 0$. But the argument further shows that $x = a_0 = a_{(k'-1)u} = z$. Hence $v' = 0$. The rest of the argument

holds. □

Remark When a is not minimum in (2) for $r = 0$ we may actually have other solutions that $i = j = 0$. Take $q = k = 3$ and $a = 16$. We then have that $1 + a \equiv 3(1 + 3a) \bmod 26$.

In the proof of Lemme 2 we will make use of the easy

Property 2 *If k is a prime integer and $v < k$ then, if v is not zero, $q^v - 1$ is a unit mod $(q^k - 1)$.*

In view of applying Lemma 2 to section 3.3.2 we also recall

Property 3 *Denote by d the integer $(q^{tk} - 1)/(q^k - 1)$. Then $(d, q^k - 1) = (t, q^k - 1)$.*

We now have

Lemma 2 *Let q and t be given integers and let k be a prime integer. Denote $q^{kt} - 1$ by n, $q^k - 1$ by m and $d = n/m$. Now let s be any integer in $[0, kt[$ such that $(q, s) = 1$ and $q^j s \equiv s \bmod n$ only for $j \equiv 0 \bmod kt$. Moreover assume that $(t, m) = 1$. Then there exist an r in $[0, k[$ such that the equation*

$$s + q^r d \equiv q^j (s + q^i d) \bmod n \tag{5}$$

for the unknowns i, j with i in $[0, k[$ and j in $[0, kt[$ has for only solution $j = 0$ and $i = r$.

Preliminary remark Since $q^r d$ and $q^i d$ are elements in $\mathbb{Z}/n\mathbb{Z}$ in equation (5), then r and i may be defined indifferently in $\mathbb{Z}/k\mathbb{Z}$ or in $\mathbb{Z}/kt\mathbb{Z}$.

Now equation (5) may be rewritten as

$$s(q^j - 1)/d = q^u (q^v - 1) + wm \tag{6}$$

Where $u \equiv i + j \bmod k$ and $v \equiv r - i - j \bmod k$. With the notations

$n_1 = (s,d), d = n_1 n_2$ and $s = n_1 b$. then L.H.S. of (6) writes

$$b((q^j-1)/n_2). \tag{7}$$

Then by Property 2 we have either that $(b,m) = 1$ or that $v = 0$ If $v = 0$ then $j \equiv 0 \ mod \ kt$, by hypothesis. We now rewrite (5) in the form

$$b + q^r n_2 = q^j(b + q^i n_2) + w n_2 m \tag{8}$$

But by Property 3 and by hypothesis. we have that $(n_2,m) = 1$ which entails that n_2 verifies the hypothesis on c of Lemma 1. Then Lemma 1 applies and there exists an r in $[0,k[$ such that necessarily $j \equiv 0 \ mod \ k$ and $i \equiv r \ mod \ k$. But this is not enough since we need that $j \equiv 0 \ mod \ kt$. Coming back to equation (6) under the form

$$s(q^j-1) = q^u(q^v-1)d + wn. \tag{9}$$

since now m divides q^j-1 and $(m,d) = 1$, then m divides q^v-1 , $s(q^j-1) \equiv 0 \ mod \ n$ and by hypothesis, $j \equiv 0 \ mod \ kt$.

Remark If $(s,n) = 1$ we don't need k to be a prime. For, here $b = s$ and $(s,m) = 1$.

```
(d1)              [>user_dir_dir>Algo>Camion>writefile.output]

(c2) batch(es);

(c3) p1:z↑4+5*z↑2+4*z+1;
                                   4      2
(d3)                              z  + 5 z  + 4 z + 1

(c4) p2:z↑4+z+1;
                                     4
(d4)                                z  + z + 1

(c5) showtime:true$
Time= 3 msec.

(c6) q1:rat(diff(p1,z))$
Time= 103 msec.

(c7) q2:rat(diff(p2,z))$
Time= 15 msec.

(c8) b1:lopow(q1,z)$
Time= 17 msec.

(c9) b2:lopow(q2,z)$
Time= 11 msec.

(c10) q1:rat(q1/z↑b1)$
Time= 26 msec.
```

```
(c11) q2:rat(q2/z^b2)$
Time= 13 msec.

(c12) h1:hipow(p1,z)$
Time= 6 msec.

(c13) h2:hipow(p2,z)$
Time= 6 msec.

(c14) n:h1+h2;
Time= 4 msec.
(d14)                                  16

(c15) m:2*n-1$
Time= 8 msec.

(c16) n1:hipow(q1,z)$
Time= 14 msec.

(c17) n2:hipow(q2,z)$
Time= 11 msec.

(c18) m1:h1+2*n-n1-1$
Time= 10 msec.

(c19) m2:h2+2*n-n2-1$
Time= 10 msec.

(c20) rp1:expand(z^h1*subst(1/z,z,p1))$
Time= 75 msec.

(c21) rp2:expand(z^h2*subst(1/z,z,p2))$
Time= 30 msec.

(c22) q1:expand(z^n1*subst(1/z,z,q1))$
Time= 54 msec.
```

```
(c23) q2:expand(ztn2*subst(1/z,z,q2))$
Time= 35 msec.

(c24) l1:expand(ztm1*q1)$
Time= 32 msec.

(c25) l2:expand(ztm2*q2)$
Time= 21 msec.

(c26) prem:9999999967$
Time= 3 msec.

(c27) modulus:prem$
Time= 101 msec.

(c28) s1:first(divide(l1,rp1,z))$
Time= 285 msec.

(c29) s2:first(divide(l2,rp2,z))$
Time= 247 msec.

(c30) t1:hipow(s1,z)$
Time= 15 msec.

(c31) t2:hipow(s2,z)$
Time= 14 msec.

(c32) s1:expand(ztt1*subst(1/z,z,s1))$
Time= 884 msec.
```

```
(c33) s2:expand(ztt2*subst(1/z,z,s2))$
Time= 818 msec.

(c34) s1:expand(ztb1*s1)$
Time= 474 msec.

(c35) s2:expand(ztb2*s2)$
Time= 445 msec.

(c36) s:rat(sum(coeff(s1,z,i)*coeff(s2,z,i)*zti,i,0,m));
Time= 2334 msec.
                      31              30              29              28
(d36)/R/ 4356606612 z   - 1990129252 z   + 4408090363 z   + 1960092229 z

              27              26              25              24
 - 3511797398 z   - 2817512357 z   + 4343382825 z   - 4532248556 z

              23              22              21              20
 + 2628461888 z   + 2331847903 z   + 2479428783 z   - 1315430700 z

             19              18              17             16             15
 + 762725124 z   + 601379796 z   + 189318156 z   + 17541216 z   - 8496154 z

            14             13             12             11            10       9
 - 1764816 z   + 2043824 z   + 1598744 z   + 603900 z   + 116736 z   + 36 z

        8        7        6        5       4       3      2
 - 1796 z  + 4134 z  + 4344 z  + 1500 z  + 384 z  + 54 z  + 4 z  + 6 z + 4

(c37) e1:zt(2*n)$
Time= 7 msec.

(c38) e2:s$
Time= 4 msec.

(c39) q:e1$
```

```
Time= 4 msec.

(c40) for i:1 thru m unless hipow(q,z)<n do(
                                        q:remainder(e1,e2,z),e1:e2,e2:q
                                        );
Time= 2270 msec.
(d40)                                done

(c41) q;
Time= 1 msec.
                      15              12              11              9
(d41)/R/ - 409336809 z   + 1491033896 z   + 877880040 z   + 2077989971 z

             8              7              6              5              4
 - 2630234690 z  + 827187193 z  - 1783660576 z  + 3421007812 z  - 1929973916 z

             3
 + 4239681430 z  + 3494164482 z + 2602334194

(c42) bq:lopow(q,z)$
Time= 52 msec.

(c43) q:rat(q/ztbq)$
Time= 15 msec.

(c44) hq:hipow(q,z)$
Time= 52 msec.

(c45) bs:lopow(s,z)$
Time= 136 msec.
```

```
(c46) s:rat(s/ztbs)$
Time= 20 msec.

(c47) hs:hipow(s,z)$
Time= 140 msec.

(c48) q:expand(ztbq*subst(1/z,z,q))$
Time= 277 msec.

(c49) s:expand(ztbs*subst(1/z,z,s));
Time= 2983 msec.
          31       30       29        28         27          26          25          24
(d49) 4 z   + 6 z   + 4 z   + 54 z   + 384 z   + 1500 z   + 4344 z   + 4134 z

          23        22           21           20           19           18
 - 1796 z   + 36 z   + 116736 z   + 603900 z   + 1598744 z   + 2043824 z

           17           16           15           14           13
 - 1764816 z   - 8496154 z   + 17541216 z   + 189318156 z   + 601379796 z

             12           11           10          9
 + 762725124 z   - 1315430700 z   + 2479428783 z   + 2331847903 z

             8           7           6          5
 + 2628461888 z  - 4532248556 z  + 4343382825 z  - 2817512357 z

             4           3           2
 - 3511797398 z  + 1960092229 z  + 4408090363 z  - 1990129252 z + 4356606612

(c50) ns:hs+bs+n-hq-bq$
Time= 10 msec.

(c51) lq:expand(ztns*q)$
Time= 144 msec.
```

```
(c52) pol:first(divide(lq,s,z));
Time= 435 msec.
                    16              15              14              12
(d52) - 4349416435 z   - 2602334194 z   - 1747082241 z   + 3940079626 z

          11              10              9              8
 + 4385994770 z   - 3903501291 z   - 2602334194 z  - 1353398395 z

             7              6             4              3
 - 1929973916 z  + 2792200993 z  - 73156670 z  + 2192997385 z  - 4349416435

(c53) pol:num(pol)$
Time= 12 msec.

(c54) pol:rat(subst(1/z,z,pol)*zthipow(pol,z));
Time= 275 msec.
                    16              13              12              10
(d54)/R/ - 4349416435 z   + 2192997385 z   - 73156670 z   + 2792200993 z

             9              8              7              6
 - 1929973916 z  - 1353398395 z  - 2602334194 z  - 3903501291 z

             5              4              2
 + 4385994770 z  + 3940079626 z  - 1747082241 z  - 2602334194 z - 4349416435

(c55) pol:rat(pol/coeff(pol,z,hipow(pol,z)));
Time= 81 msec.
          16      13      12      10      9      8      7      6
(d55)/R/ z   - 12 z   + 46 z   + 158 z  + 120 z  + 851 z  - 4 z  - 6 z

                                                5      4      2
                                             - 24 z  - 17 z  + 5 z  - 4 z + 1
```

```
(c56) modulus:false$
Time= 4 msec.

(c57) factor(pol);
Time= 2753 msec.
        4      3      2                 12      11      10      9       8       7
(d57) (z  + 3 z  + 5 z  - 5 z + 1) (z   - 3 z   + 4 z   - 4 z  + 22 z  - 23 z

                                      6       5       4       3      2
                                + 93 z  + 70 z  + 39 z  + 17 z  + 5 z  + z + 1)

(c58) fac:first(%)$
Time= 28 msec.

(c59) fac:subst(x*y,z,fac);
Time= 47 msec.
                      4  4      3  3      2  2
(d59)                x  y  + 3 x  y  + 5 x  y  - 5 x y + 1

(c60) r1:subst(x,z,p1)$
Time= 18 msec.

(c61) r2:subst(y,z,p2)$
Time= 10 msec.

(c62) algebraic:true;
Time= 4 msec.
(d62)                                 true

(c63) tellrat(r2);
Time= 14 msec.
                                       4
(d63)                                 [y  + y + 1]
```

```
(c64) r:fac$
Time= 4 msec.

(c65) for i:1 thru m2 unless hipow(fac,x)=0 do(
fac:remainder(r,r1,x),r:r1,r1:fac)$
Time= 965 msec.

(c66) fac:rat(r,x);
Time= 33 msec.
                          3                    2
(d66)/R/ ((1106103749 y  + 3657942137 y  - 4404169234 y + 5322210562) x

                          3                    2
         + 746227097 y  + 188062421 y  + 547939073 y + 4764045886)/3198655227

(c67) g:coeff(fac,x,hipow(fac,x))$
Time= 69 msec.

(c68) g:rat(1/g)$
Time= 78 msec.

(c69) fac:g*fac$
Time= 70 msec.

(c70) fac:rat(fac,x);
Time= 19 msec.
                                      2
(d70)/R/                        x - y  - y

Time= 22601 msec.
(d71)                           BATCH DONE

(c72) closefile(tetaplusteta2);
```

REFERENCE

[1] E.R. Berlekamp
"Factoring polynomials over finite fields"
Bell System Tech. J. 46 (1967) 1853-1859

[2] E.R. Berlekamp
"Factoring polynomials over large finite fields"
Math. Comp. 24 (1970) 713-735

[3] E.R. Berlekamp
"Algebraic coding theory"
Mac Graw-Hill (1968)

[4] P. Camion
"Un algorithme de construction des idempotents primitifs d'ide'aux d'alge'bres sur F_q."
C.R. Acad. Sc. Paris t.291 (20 octobre 1980)

[5] P. Camion
"Factorisation des polynomes de $Fq[X]$"
Revue du CETHEDEC, NS. 812 (4eme trimestre 1981)

[6] P. Camion
"A deterministic algorithm for factorizing polynomials of $F_q[X]$,"
Annals of Discrete Math 17 (1983) 149-157
North-Holland Publishing Company.

[7] P. Camion
"Un algorithme de construction des idempotents primitifs d'ide'aux d'alge'bres sur F_q."
Annals of discrete Math 12/1982 55-63

[8] P. Camion
"Improving an algorithm for Factoring Polynomials over a Finite Field and Constructing Large Irreducible Polynomials"
IEE mai 1983

[9] P. Camion
"Abelian Codes"
MRC Technical Summary
Report 1059 (1971)

[10] D.G. Cantor and H Zassenhaus
"A New Algorithm for Factoring Polynomials over Finite Fields"
Math. Comp. Vol 36. #154 (1981) 587-592

[11] D.E. Knuth
"The art of computer programming"
Vol. 2. Seminumerical algorithms (Addison-Wesley, Reading, MA)

[12] S. Lang
"Algebra"
Addison Wesley.

[13] M.O. Rabin
"Probabilistic algorithms in fnite fields"
SIAM J. On comp. 9 (1980) 273-280

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique