Improving the Berlekamp algorithm for binomials $x^n - a$

Ryuichi Harasawa, Yutaka Sueyoshi, and Aichi Kudo

Graduate School of Engineering, Nagasaki University,
1-14 Bunkyomachi, Nagasaki-shi, Nagasaki, 852-8521, Japan
{harasawa,sueyoshi,kudo}@cis.nagasaki-u.ac.jp

Abstract. In this paper, we describe an improvement of the Berlekamp algorithm, a method for factoring univariate polynomials over finite fields, for binomials $x^n - a$ over finite fields $\mathbb{F}_q$. More precisely, we give a deterministic algorithm for solving the equation $h(x)^q \equiv h(x) \pmod{x^n - a}$ directly without applying the sweeping-out method to the corresponding coefficient matrix. We show that the factorization of binomials using the proposed method is performed in $O\tilde{}(n \log q)$ operations in $\mathbb{F}_q$ if we apply a probabilistic version of the Berlekamp algorithm after the first step in which we propose an improvement. Our method is asymptotically faster than known methods in certain areas of q, n and as fast as them in other areas.

Keywords: finite field, polynomial factorization, Berlekamp algorithm, binomial

1 Introduction

The factorization of univariate polynomials over finite fields is one of the interesting topics in computer algebra, for example, it is used to determine the decomposition of prime numbers in number fields and to construct (non-prime) finite fields and so on.

Applying the formal derivation, we can reduce the factorization of polynomials over finite fields to that of square-free polynomials (i.e., polynomials having no multiple factors) [10, 11]. For the factorization of square-free polynomials over finite fields, the Berlekamp algorithm is well known [4, 10].

In this paper, we propose an improvement of the Berlekamp algorithm for binomials $x^n - a$ over finite fields $\mathbb{F}_q$. More precisely, we give a deterministic algorithm for solving the equation $h(x)^q \equiv h(x) \pmod{x^n - a}$ directly without applying the sweeping-out method to the corresponding coefficient matrix, which is a generarization of the Prange method for $x^n - 1$ [16]. We show that the factorization of binomials using the proposed

method is performed in $O^{\sim}(n \log q)$ operations in $\mathbb{F}_q$ if we apply a probabilistic version of the Berlekamp algorithm after the first step in which we propose an improvement. Our method is asymptotically faster than known methods (for example the Berlekamp method [4], the Gathen and Shoup method [9], and the Kaltofen and Shoup method [13], the Cantor and Zassenhaus method [6], [10, Figure 14.9]) in certain areas of q, n and as fast as them in other areas (Fig. 1). We mention that both the Gathen and Shoup method [9] and the Kaltofen and Shoup method [13] improve the distinct-degree factorization in the Cantor and Zassenhaus method, by using the "iterated Frobenius" method for the former and using fast matrix multiplication for the latter.

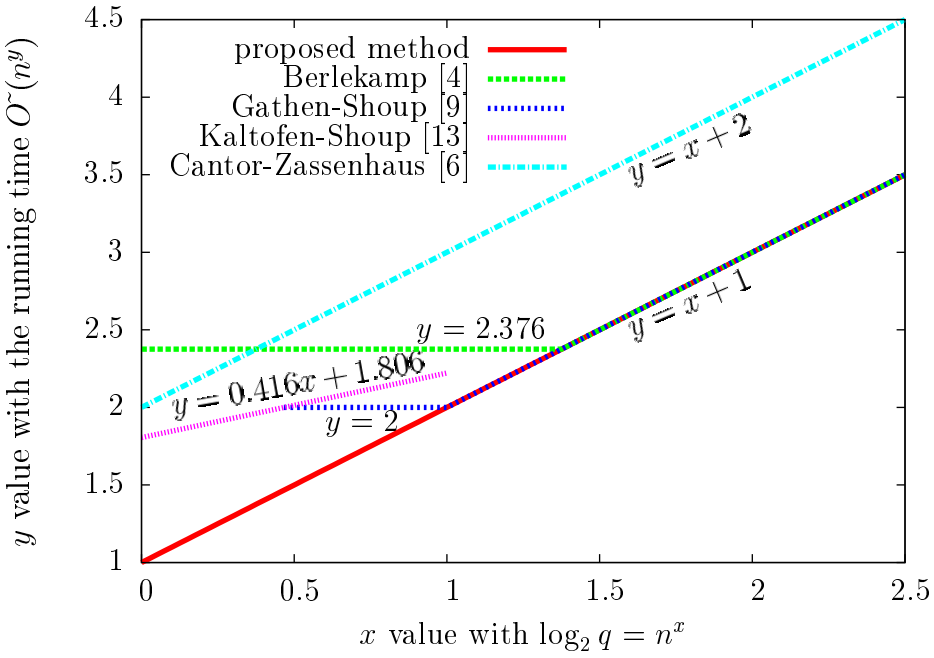

Fig. 1. Running times of some factoring algorithms

Note that there exist some efficient methods for computing the solution of $x^n = a$ over finite fields (e.g., [1, 21]).

The remainder of this paper is organized as follows: In section 2, we introduce some theoretical results on binomials over finite fields. In

Section 3, we describe the Berlekamp algorithm. In Section 4, we propose an improvement of the Berlekamp algorithm for binomials $x^n - a$. In section 5, we estimate the complexity of the proposed method. In Section 6, we give the conclusion and future works.

2 Binomials

In this section, we introduce some (theoretical) facts on binomials over finite fields $\mathbb{F}_q$, while we do not apply directly these facts to our proposed method. However, we use them to make a lot of examples $x^n - a$ for various cases q and n (that is, various patterns of factorizations).

At first, for the irreducibility of binomials, the following result is known:

Theorem 1 [14, Theorem 3.75]**.** *Let $n \geq 2$, $a \in \mathbb{F}_q^*$, and let e be the multiplicative order of a. Then we have*

$$x^n - a \text{ is irreducible in } \mathbb{F}_q[x]$$
$$\iff \begin{cases} (i) \text{ for each prime divisor } p \text{ of } n,\ p|e \text{ and } p \nmid \frac{q-1}{e}, \\ (ii)\ q \equiv 1 \pmod 4 \qquad \text{if } n \equiv 0 \pmod 4. \end{cases}$$

We next describe a result on the number of irreducible factors of binomials. Let $\mu(\cdot)$ denote the Möbius function, that is,

$$\mu(x) = \begin{cases} 1 & (x = 1), \\ 0 & (x \text{ is not square-free}), \\ (-1)^s & (x \text{ is the product of } s \text{ distinct primes}). \end{cases}$$

For a binomial $x^n - a$ over $\mathbb{F}_q$, we denote by δ_i the number of roots of it in $\mathbb{F}_{q^i}$. If we put $d_i = \gcd(q^i - 1, n)$, then we easily see that

$$\delta_i = \begin{cases} d_i & (a^{(q^i-1)/d_i} = 1), \\ 0 & (\text{otherwise}). \end{cases}$$

We get a result on the number of irreducible factors of $x^n - a$ as follows:

Theorem 2 [19]**.** *With the notation as above, we assume* $\operatorname{char} \mathbb{F}_q \nmid n$ *and $a \neq 0$. Let σ_t be the number of irreducible factors of degree t of $x^n - a$. Then we have*

$$\sigma_t = \frac{1}{t} \sum_{i|t} \mu\Big(\frac{t}{i}\Big) \delta_i.$$

Proof. We easily see that

$$\delta_t = \sum_{i|t} i\sigma_i.$$

Applying Möbius's inversion formula, we have

$$t\sigma_t = \sum_{i|t} \mu\Big(\frac{t}{i}\Big)\delta_i,$$

which implies the assertion of the theorem. □

Remark 1. *This theorem implies that, for given q and n with* $\operatorname{char}\mathbb{F}_q \nmid n$, *the values σ_t's depend only on the multiplicative order of a.*

In the rest of this section, we describe a result on the minimal degree (resp. the maximal degree) of irreducible factors of binomials.

Proposition 1. *With the notation as above, we assume* $\operatorname{char}\mathbb{F}_q \nmid n$ *and $a \neq 0$. Let $\kappa = \min\{i \mid \delta_i \neq 0\}$, that is, the extension field $\mathbb{F}_{q^\kappa}$ is the minimal field that contains some root of $x^n - a$. Let ζ_n be a primitive n-th root of unity, and $\lambda = [\mathbb{F}_q(\zeta_n) : \mathbb{F}_q]$ (in other words, the value λ is equal to the order of q in $(\mathbb{Z}/n\mathbb{Z})^*$). Then we have*

1. *the minimal degree of irreducible factors of $x^n - a$ is equal to κ,*
2. *the maximal degree of irreducible factors of $x^n - a$ is equal to* $\operatorname{lcm}(\kappa, \lambda)$.

Proof. It is obvious for the minimal degree.

For the maximal degree, let e be the multiplicative order of a and ζ_{en} a primitive en-th root of unity. Then there exists an element b in $\mathbb{F}_q$ with multiplicative order e such that ζ_{en} is a root of $x^n - b$. Indeed, since ζ_{en} is a root of $x^{en} - 1$ and $x^{en} - 1 = \prod_{0 \leq i < e}(x^n - a^i)$, there exists j such that ζ_{en} is a root of $x^n - a^j$. Setting $b = a^j$, we see that the multiplicative order of b is equal to e (equivalently $\gcd(j, e) = 1$), because ζ_{en} is a primitive en-th root of unity. We further see, from Theorem 2 (or Remark 1), that the pattern of the factorization of $x^n - a$ coincides with that of $x^n - b$. Namely, for each $i \geq 1$, the number of irreducible factors of degree i of $x^n - a$ is equal to that of $x^n - b$. Therefore it is sufficient to show our assertion for $x^n - b$.

Let u be a root in $\mathbb{F}_{q^\kappa}$ of $x^n - b$. We see that $u\zeta_n^i$ $(0 \leq i \leq n-1)$ are the roots of $x^n - b$, which implies that the splitting field of $x^n - b$ over $\mathbb{F}_q$, say $\mathbb{K}$, becomes $\mathbb{K} = \mathbb{F}_q(u, \zeta_n)$. So the maximal degree of irreducible factors of $x^n - b$ is less than or equal to $[\mathbb{K} : \mathbb{F}_q] = \operatorname{lcm}(\kappa, \lambda)$. On the other hand,

we see that $\{\zeta_{en}^{1+ie} \mid 0 \leq i \leq n-1\}$ gives an alternative representation of the roots of $x^n - b$. Hence we have $\mathbb{K} = \mathbb{F}_q(\zeta_{en})$ by considering the spliting field for this representation of the root of $x^n - b$, which implies the degree of minimal polynomial, say $\psi(x)$, of ζ_{en} over $\mathbb{F}_q$ is equal to $[\mathbb{K} : \mathbb{F}_q] = \mathrm{lcm}(\kappa, \lambda)$. Therefore the maximal degree of irreducible factors of $x^n - b$ is greater than or equal to $\mathrm{lcm}(\kappa, \lambda)$ because $\psi(x)$ is an irreducible factor of $x^n - b$. So we get the desired result. □

3 Berlekamp algorithm

In this section, we assume that $\mathrm{char}\,\mathbb{F}_q > 2$ for simplicity. The Berlekamp algorithm [4, 10] is a well-known algorithm for factoring square-free polynomials over finite fields. In Table 1, we describe its procedure [1]. From Step 3, we see this algorithm is probabilistic, which runs in polynomial time for the input size $O(n \log q)$ with n the degree of polynomial to be factored (Fig. 1). We note that there exists a deterministic procedure for Step 3, but the complexity is not the polynomial time for the input size [3]. So we use a probabilistic version of the Berlekamp method in this paper.

In the next section, we focus on Step 1 in Table 1. More precisely, we consider the equation

$$h(x)^q \equiv h(x) \pmod{f(x)} \tag{1}$$

for a square-free polynomial $f(x)$ over $\mathbb{F}_q$, i.e., the eigenspace V of the eigenvalue 1 for the linear transformation $h(x) \mapsto h(x)^q \pmod{f(x)}$ on the n-dimensional vector space $\mathbb{F}_q[x]/(f(x))$ over $\mathbb{F}_q$.

Let k denote the number of irreducible factors of $f(x)$ and $f(x) = \prod_{1 \leq i \leq k} f_i(x)$ be the factorization of $f(x)$. Then we see that the vector space $\mathbb{F}_q[x]/(f(x))$ is isomorphic to $\bigoplus_{1 \leq i \leq k} \mathbb{F}_q[x]/(f_i(x))$ and that the solution space V of the equation (1) is isomorphic to the subspace of $\bigoplus_{1 \leq i \leq k} \mathbb{F}_q[x]/(f_i(x))$ consisting of $(a_1, \cdots, a_k)$ with each a_i in $\mathbb{F}_q$. Hence, the number of irreducible factors of $f(x)$ is equal to the dimension of V over $\mathbb{F}_q$.

Remark 2. *For Step* 3 *in Table 1, if the polynomial* $v(x)$ *is not irreducible, then the probability that* $\gcd(g(x), v(x))$ *or* $\gcd(g(x)^{\frac{q-1}{2}} - 1, v(x))$ *(or both) is a proper factor of* $v(x)$ *is at least* $\frac{1}{2}$ *from the description above.*

[1] If $q = 2^m$, we perform the same procedure as in $\mathrm{char}\,\mathbb{F}_q > 2$ except for Step 3. For Step 3, we compute $\gcd(\mathrm{Tr}(g(x)), v(x))$ with $g(x)$ a random element in V instead of $\gcd(g(x)^{(q-1)/2} - 1, v(x))$, where $\mathrm{Tr}(g(x)) = g(x) + g(x)^2 + \cdots + g(x)^{2^{m-1}} \bmod v(x)$ [9, Algorithm 3.6 (Step 4)].

Table 1. Berlekamp algorithm

Input: A square-free polynomial $f(x)$ over $\mathbb{F}_q$. **Output:** The factorization of $f(x)$.
Step 1: Compute the polynomials $h(x)$ over $\mathbb{F}_q$ of degree less than $\deg f(x)$ such that $h(x)^q \equiv h(x) \pmod{f(x)}$. The set V of $h(x)$'s forms an $\mathbb{F}_q$-vector space. Let $\{h_1(x), \ldots, h_k(x)\}$ be a basis of V.
Step 2: $F \leftarrow \{f(x)\}$. **if** $k = 1$, go to Step 4.

Step 3: **while** $\#F < k$
Choose a random element $g(x)$ in V
($g(x)$ is of the form $\sum_{1 \le i \le k} \lambda_i h_i(x)$ with $\lambda_i \in \mathbb{F}_q$).
For each $v(x) \in F$, do the following procedure:
Compute $d(x) = \gcd(g(x), v(x))$.
if $0 < \deg d(x) < \deg v(x)$
$F \leftarrow (F \setminus \{v(x)\}) \cup \{d(x), v(x)/d(x)\}$.
if $\#F = k$, go to Step 4.
$v(x) \leftarrow v(x)/d(x)$.
end if
Compute $d(x) = \gcd(g(x)^{(q-1)/2} - 1, v(x))$.
if $0 < \deg d(x) < \deg v(x)$
$F \leftarrow (F \setminus \{v(x)\}) \cup \{d(x), v(x)/d(x)\}$.
if $\#F = k$, go to Step 4.
end if
end while

Step 4: Return F (the product of the elements in F equals $f(x)$).

In the remainder of this section, for the solution $h(x)$ to the equation $h(x)^q \equiv h(x) \pmod{f(x)}$, we mention the method [14, Theorems 4.3 and 4.5] using the Prange method [16] for $f(x) = x^n - 1$.

Let $f(x)$ be a polynomial over $\mathbb{F}_q$ with $f(0) \neq 0$ and s the least positive integer satisfying $f(x) \mid x^s - 1$, which is called the order of $f(x)$. Applying the Prange method to $x^s - 1$, we then get the solution above [14, Theorems 4.3 and 4.5]. Namely, for each $\alpha \geq 0$, let l be the least positive integer such that

$$x^{\alpha q^l} \equiv x^\alpha \pmod{f(x)},$$

which is equivalent to

$$\alpha q^l \equiv \alpha \pmod{s}.$$

Defining

$$h_\alpha(x) = x^\alpha + x^{\alpha q} + \cdots + x^{\alpha q^{l-1}},$$

we see that $h_\alpha(x)^q \equiv h_\alpha(x) \pmod{f(x)}$. Moreover, the set $\{h_\alpha(x) \bmod x^s - 1 \mid \alpha \in \mathbb{Z}/s\mathbb{Z}\}$ forms a basis of the solution space of $h_\alpha(x)^q \equiv h_\alpha(x) \pmod{x^s - 1}$, which we see in the next section (see Remark 3. Also [10, p. 419 (Exercise 14.47)] or [16]). If $f(x) \neq x^s - 1$, then the set $\{h_\alpha(x) \bmod f(x) \mid \alpha \in \mathbb{Z}/s\mathbb{Z}\}$ forms a generator system of the solution space of $h_\alpha(x)^q \equiv h_\alpha(x) \pmod{f(x)}$.

Our main assertion in this paper is that we analyze the method above more strictly and simplify it in the case of binomials $f(x) = x^n - a$. We emphasize that our method does not need the computation of the value s, the order of $f(x)$. In the case $f(x) = x^n - a$, the order of $f(x)$ is $s = en$ with e the multiplicative order of a [14, Lemma 3.17]. So, if we need the value s, then we need to find the multiplicative order of a, for which we might compute the factorization of $q - 1$. However the task becomes extremely heavy as q becomes large.

4 Our algorithm

In this section, we propose an improved method for obtaining a basis of V in Step 1 of Table 1 for $x^n - a$. Namely, we solve

$$h(x)^q \equiv h(x) \pmod{x^n - a} \tag{2}$$

by a new method. Since $x^n - a$ is assumed to be square-free, $\operatorname{char} \mathbb{F}_q$ does not divide n.

For $f(x) = x^n - a$, instead of dealing with the coefficient matrix corresponding to the equation above, we consider the orbits in $\mathbb{Z}/n\mathbb{Z}$ according to the action of $\langle q \rangle$ by multiplication.

For α in $\mathbb{Z}/n\mathbb{Z} = \{0, 1, \ldots, n-1\}$, let l be the least positive integer such that $q^l \alpha \equiv \alpha \pmod{n}$ and let $\alpha_i = q^i \alpha \bmod n$ for $0 \leq i < l$. We denote the orbit of α by $\bar{\alpha} = \{\alpha_0, \alpha_1, \ldots, \alpha_{l-1}\}$. In particular, we have $\bar{0} = \{0\}$.

For each orbit $\bar{\alpha}$, we consider the subspace

$$T_{\bar{\alpha}} = \{\beta_0 x^{\alpha_0} + \beta_1 x^{\alpha_1} + \cdots + \beta_{l-1} x^{\alpha_{l-1}} \bmod x^n - a \mid \beta_i \in \mathbb{F}_q\}$$

of $\mathbb{F}_q[x]/(x^n - a)$.

Since $q\alpha_i \equiv \alpha_{i+1} \pmod{n}$ $(0 \leq i < l-1)$ and $q\alpha_{l-1} \equiv \alpha_0 \pmod{n}$, we put

$$q\alpha_i = c_i n + \alpha_{i+1} \ (0 \leq i < l-1) \text{ and } q\alpha_{l-1} = c_{l-1} n + \alpha_0$$

with integers c_i $(0 \le i < l)$. Then we see

$$(x^{\alpha_i})^q \equiv a^{c_i} x^{\alpha_{i+1}} \pmod{x^n - a} \ (0 \le i < l-1) \text{ and}$$
$$(x^{\alpha_{l-1}})^q \equiv a^{c_{l-1}} x^{\alpha_0} \pmod{x^n - a}.$$

Hence, for $h_{\bar{\alpha}}(x) = \beta_0 x^{\alpha_0} + \beta_1 x^{\alpha_1} + \cdots + \beta_{l-1} x^{\alpha_{l-1}}$ in $T_{\bar{\alpha}}$, we have

$$h_{\bar{\alpha}}(x)^q \equiv a^{c_{l-1}} \beta_{l-1} x^{\alpha_0} + a^{c_0} \beta_0 x^{\alpha_1} + \cdots + a^{c_{l-2}} \beta_{l-2} x^{\alpha_{l-1}} \pmod{x^n - a}.$$

Therefore, for the linear transformation π_q of $\mathbb{F}_q[x]/(x^n - a)$ defined by $\pi_q(h(x)) = h(x)^q \bmod x^n - a$, the subspace $T_{\bar{\alpha}}$ is π_q-invariant and $\mathbb{F}_q[x]/(x^n - a) = \bigoplus_{\bar{\alpha}} T_{\bar{\alpha}}$. We put $V_{\bar{\alpha}} = V \cap T_{\bar{\alpha}}$, then $V = \bigoplus_{\bar{\alpha}} V_{\bar{\alpha}}$ and $V_{\bar{0}} = T_{\bar{0}} \simeq \mathbb{F}_q$.

If there exist w orbits $\bar{\alpha}$'s in $\mathbb{Z}/n\mathbb{Z}$, the equation (2) is divided into w equations

$$h_{\bar{\alpha}}(x)^q \equiv h_{\bar{\alpha}}(x) \pmod{x^n - a} \tag{3}$$

where $h(x) = \sum_{\bar{\alpha}} h_{\bar{\alpha}}(x)$ with $h_{\bar{\alpha}}(x)$ as above. For the orbit $\bar{0}$, the constant polynomial 1 forms a basis of one-dimensional vector space $V_{\bar{0}}$.

We consider the orbit $\bar{\alpha} \neq \bar{0}$. Then the equation (3) is written as

$$\begin{cases} \beta_0 = a^{c_{l-1}} \beta_{l-1} \\ \beta_1 = a^{c_0} \beta_0 \\ \quad \vdots \\ \beta_{l-1} = a^{c_{l-2}} \beta_{l-2}, \end{cases}$$

which leads to the relation

$$\beta_0 = a^{c_0 + c_1 + \cdots + c_{l-1}} \beta_0.$$

Therefore, we obtain the solution(s) of (3) as follows:

$$\begin{cases} \beta(x^{\alpha_0} + a^{c_0} x^{\alpha_1} + a^{c_0+c_1} x^{\alpha_2} + \cdots + a^{c_0+c_1+\cdots+c_{l-2}} x^{\alpha_{l-1}}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\text{if } a^{c_0+c_1+\cdots+c_{l-1}} = 1), \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\text{otherwise}), \end{cases}$$

where β runs over all elements of $\mathbb{F}_q$. The solution space $V_{\bar{\alpha}}$ of the equation (3) is $\{0\}$ if $a^{c_0+c_1+\cdots+c_{l-1}} \neq 1$ and, otherwise, forms one-dimensional subspace of $T_{\bar{\alpha}}$ generated by $x^{\alpha_0} + a^{c_0} x^{\alpha_1} + a^{c_0+c_1} x^{\alpha_2} + \cdots + a^{c_0+c_1+\cdots+c_{l-2}} x^{\alpha_{l-1}}$.

We describe the proposed algorithm in Table 2.

Table 2. Solutions of $h(x)^q \equiv h(x) \pmod{x^n - a}$

Input: A binomial $x^n - a$ over $\mathbb{F}_q$ with $\operatorname{char} \mathbb{F}_q \nmid n$.
Output: A basis B of the solution space V of $h(x)^q \equiv h(x) \pmod{x^n - a}$.
Step 1: $B \leftarrow \{1\}$, $G \leftarrow \{1, 2, \ldots, n-1\}$.
Step 2: **if** $G = \emptyset$, return B.
Step 3: $i_0 \leftarrow \min\{i \mid i \in G\}$, $G \leftarrow G \setminus \{i_0\}$, $j \leftarrow i_0$, $f \leftarrow x^j$, $b \leftarrow 1$
Step 4: Compute the integers $t,\ r$ such that $jq = tn + r$ with $0 \le r < n$. $b \leftarrow b \cdot a^t$.
Step 5: **while** $r \neq i_0$ $G \leftarrow G \setminus \{r\}$, $f \leftarrow f + b \cdot x^r$, $j \leftarrow r$. Compute the integers $t,\ r$ such that $jq = tn + r$ with $0 \le r < n$. $b \leftarrow b \cdot a^t$. **end while**
Step 6: **if** $b = 1$, $B \leftarrow B \cup \{f\}$. **goto** Step 2.

Remark 3. *For $i \ge 1$, we see $x^{\alpha q^i} \equiv a^{c_0+c_1+\cdots+c_{i-1}} x^{\alpha_i} \pmod{x^n - a}$ by the definition of α_i and c_i. This implies that $a^{c_0+c_1+\cdots+c_{l-1}} = 1$ holds if and only if $x^{\alpha q^l} \equiv x^\alpha \pmod{x^n - a}$, and then $\sum_{0 \le i \le l-1} x^{\alpha q^i} \bmod x^n - a$ is a solution of the equation (3). Especially, in the case of $a = 1$, the equation $a^{c_0+c_1+\cdots+c_{l-1}} = 1$ always holds for all $\alpha \in \mathbb{Z}/n\mathbb{Z}$, which implies that the set $\{\sum_{0 \le i \le l-1} x^{\alpha q^i} \bmod x^n - 1 \mid \alpha \in \mathbb{Z}/n\mathbb{Z}\}$ forms an $\mathbb{F}_q$-basis of the solution space V and that the dimension of V (i.e., the number of the irreducible factors of $x^n - 1$) is equal to the number of the orbits in $\mathbb{Z}/n\mathbb{Z}$ with respect to $\langle q \rangle$.*

As a theoretical consideration about the proposed method, we get the following result on the congruence equation $x^{\alpha q^l} \equiv x^\alpha \pmod{x^n - a}$ in Remark 3.

Proposition 2. *With the notation as above, we put $d = \gcd(\alpha, n)$. Then we have*

1. $\#\bar{d} = \#\bar{\alpha}(= l)$.
2. $x^{dq^l} \equiv x^d \pmod{x^n - a} \iff x^{\alpha q^l} \equiv x^\alpha \pmod{x^n - a}$.

Proof. From the definition of d, we denote $\alpha = sd$ with $\gcd(s, \frac{n}{d}) = 1$.

For the first assertion, it is sufficient to show that $dq^i \equiv d \pmod{n}$ if and only if $\alpha q^i \equiv \alpha \pmod{n}$. By multiplying s (resp. $s^{-1} \bmod \frac{n}{d}$) to the both sides, we obtain the only if part (resp. the if part).

In the same way, we get the second assertion. □

Proposition 2 implies that, for two orbits $\bar{\alpha}$ and $\bar{\alpha'}$ with $\gcd(\alpha, n) = \gcd(\alpha', n)$, $\sum_{0 \le i \le l-1} x^{\alpha q^i} \bmod x^n - a$ is a solution of the equation (3) if and only if so is $\sum_{0 \le i \le l-1} x^{\alpha' q^i} \bmod x^n - a$.

5 Complexity

We estimate the complexity of the factorization of binomials using the proposed procedure described in the previous section (Table 2). Namely, we describe the complexity not only of the proposed method but also of the procedures of square-free factorization and of finding irreducible factors using the basis which is obtained by the proposed method. The notation $O\tilde{}(x)$ means $O(x(\log x)^t)$ with some positive constant t.

We assume that the multiplication of l-bit integer and m-bit integer (resp. the division/remainder of l-bit integer by m-bit integer) needs $O(lm)$ bit operations (resp. $O(m(l-m))$ bit operations).

We first see that the square-free factorization of binomials $x^n - a$ over $\mathbb{F}_q$ of characteristic p is reduced to the computation of i and r such that $n = p^i r$ with $\gcd(p, r) = 1$ and the computation of the p^i-th root of a, say a'. Namely, the square-free factorization of $x^n - a$ becomes $x^n - a = (x^r - a')^{p^i}$. The computation of i and r takes at most

$$\begin{aligned} & O(\log p \log \frac{n}{p}) + O(\log p \log \frac{n}{p^2}) + \cdots + O(\log p \log \frac{n}{p^v}) \\ &= O(v \log p \log n) \qquad (\text{by } \log \frac{n}{p^i} \le \log n) \\ &= O(\log^2 n) \qquad (\text{by } v \log p = \log p^v \le \log n) \end{aligned}$$

bit operations, where $v := \lfloor \log_p n \rfloor$. For the computation of a', if we let $q = p^m$ then we get the p^i-th root of a by $a' = a^{p^\eta}$ with $\eta := -i \bmod m$, from which we see the computation of a' takes at most

$$O(\log p^\eta) = O(\log q) \qquad (\text{by } 0 \le \eta < m \text{ and } q = p^m)$$

operations in $\mathbb{F}_q$.

With the notations q, n and a as before, we assume that the binomial $x^n - a$ to be factored is square-free. In order to perform the proposed method (Table 2) for Step 1 in Table 1, for each j with $1 \leq j \leq n-1$, we must execute the following computations:

- the computation of the values t, r such that $jq = tn + r$ with $0 \leq r < n$;
- for the value t above, the computaion of $b \cdot a^t$ with b being a prescribed element in $\mathbb{F}_q$.

The former computation takes $O(\log j \cdot \log q) + O(\log t \cdot \log n)$ bit operations and the latter one takes $O(\log t)$ operations in $\mathbb{F}_q$. By the fact $t \leq \frac{jq}{n}$ and Stirling's formula $\log(n!) = O(n \cdot \log n)$, we estimate an upper bound of the complexity of the former task as

$$\begin{aligned}
&\sum_{1 \leq j \leq n-1} \{O(\log j \cdot \log q) + O(\log \frac{jq}{n} \cdot \log n)\} \\
&= O(n \log n \cdot \log q) + O(n \log n \cdot \log q) \qquad (\text{by } \log \frac{jq}{n} \leq \log q) \\
&= O(n \log n \cdot \log q) \\
&= \tilde{O}(n \log q) \text{ bit operations,}
\end{aligned}$$

and an upper bound of the complexity of the latter task as

$$\begin{aligned}
&\sum_{1 \leq j \leq n-1} O(\log \frac{jq}{n}) \\
&= O(n \log q) \text{ operations in } \mathbb{F}_q \qquad (\text{by } \log \frac{jq}{n} \leq \log q).
\end{aligned}$$

Therefore, the proposed method runs in $\tilde{O}(n \log q)$ operations in $\mathbb{F}_q$.

We note that, for alternative methods, it takes $O(n^3)$ (resp. $O(n^{2.376})$) operations in $\mathbb{F}_q$ using the original Gaussian elimination (resp. the Gaussian elimination using a fast method for matrix multiplication [8]) and $\tilde{O}(n^2)$ operations in $\mathbb{F}_q$ using the Kaltofen-Lobo method [12] based on the Wiedemann method [22], which is known as the fastest method for solving linear equations.

We additionally perform the final step (Step 3 in Table 1) in $\tilde{O}(n \log q)$ operations in $\mathbb{F}_q$ on average [10, Theorems 14.11 and 14.32], assuming that one applies a fast arithmetic in $\mathbb{F}_q[x]$. Namely, for two polynomials in $\mathbb{F}_q[x]$ of degree at most n, the multiplication, the division with remainder and the greatest common divisor are performed in $\tilde{O}(n)$ operations in $\mathbb{F}_q$ using the fast arithmetic in $\mathbb{F}_q[x]$ [2, 7, 17, 18]. We therefore see that

the factorization of binomials is performed in $O\tilde{}(n \log q)$ operations in $\mathbb{F}_q$, which is asymptotically faster than known methods (e.g., [4, 6, 9, 13] or [10, Figure 14.9]) in certain areas of q, n and as fast as them in other areas (Fig. 1 in Section 1).

6 Conclusion and future works

In this paper, we described an improvement of the Berlekamp algorithm for binomials $x^n - a$ over finite fields $\mathbb{F}_q$. More precisely, we proposed a method for solving the equation $h(x)^q \equiv h(x) \pmod{x^n - a}$ directly. We evaluate the complexity as $O\tilde{}(n \log q)$ operations in $\mathbb{F}_q$. Our method is asymptotically faster than known methods in certain areas of q, n and as fast as them in other areas. Our future works include the experimental consideration, the detailed analysis of other methods (e.g., the Cantor and Zassenhaus method [6] and its improvements [5, 9, 13, 15, 20]) in the case of binomials, and the combination of our method with other ones as above.

Acknowledgments

We are grateful to the referees for giving a lot of useful comments and suggestions which make this paper more valuable. This work was partially supported by the Japan Society for the Promotion of Science (JSPS) under the Grant-in-Aid for challenging Exploratory Research No. 24650009.

References

1. L. Adleman, K. Menders and G. Miller, *On taking roots in finite fields*, Proc. 18th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 175 – 178, 1977.
2. A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Reading, MA, Addison-Wesley, 1974.
3. E. R. Berlekamp, *Factoring polynomials over finite fields*, Bell System Technical Journal, **46**, pp. 1853 – 1859, 1967.
4. E. R. Berlekamp, *Factoring polynomials over large finite fields*, Math. Comp., **24**, pp. 713 – 735, 1970.
5. P. Camion, *Improving an algorithm for factoring polynomials over a finite field and constructing large irreducible polynomials*, IEEE Transactions on Information Theory, **29** (3), pp. 378 – 385, 1983.
6. D. G. Cantor and H. Zassenhaus, *A new algorithm for factoring polynomials over finite fields*, Math. Comp., **36**, pp. 587 – 592, 1981.
7. D. G. Cantor and E. Kaltofen, *On fast multiplication of polynomials over arbitrary algebras*, Acta Inform., **28**, pp. 693 – 701, 1991.

8. D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, J. Symb. Comput., **9**, pp. 251 – 280, 1990.
9. J. von zur Gathen and V. Shoup, *Computing Frobenius maps and factoring polynomials*, Comput. Complexity, **2**, pp. 187 – 224, 1992.
10. J. von zur Gathen and J. Gerhard, *Modern Computer Algebra (Second Edition)*, Cambridge, 2003.
11. K. Geddes, S. Czapor and G. Labahn, *Algorithms for Computer Algebra*, Kluwer Academic Publishers, 1992.
12. E. Kaltofen and A. Lobo, *Factoring high-degree polynomials by black box Berlekamp algorithm*, Proceedings of ISSAC'94, pp. 90 – 98, ACM Press, 1994.
13. E. Kaltofen and V. Shoup, *Subquadratic-time factoring of polynomials over finite fields*, Math. Comp., **67**, pp. 1179 – 1197, 1998
14. R. Lidl and H. Niederreiter, *Finite Fields*, Addison-Wesley, 1983.
15. R. J. McEliece, *Factorization of polynomials over finite fields*, Math. Comp., **23**, pp. 861 – 867, 1969.
16. E. Prange, *An algorithm for factoring* $X^n - 1$ *over a finite field*, Technical Report AFCRC-TN-59-775, Air Force Cambridge Research Center, Bedford MA, 1956.
17. A. Schönhage and V. Strassen, *Schnelle Multiplikation großer Zahlen*, Computing, **7**, pp. 281 – 292, 1971.
18. A. Schönhage, *Schnelle Multiplikation von Polynomen über Körpern der Charakteristik* 2, Acta Inform., **7**, pp. 395 – 398, 1977.
19. Š. Schwarz, *On the reducibility of binomial congruences and on the bound of the least integer belonging to given exponent mod p*, Časopis pro pěstování matematiky, **74**, pp. 1 – 16, 1949. Available at http://dml.cz/dmlcz/109143.
20. V. Shoup, *On the deterministic complexity of factoring polynomials over finite fields*, Information Processing Letters, **33**, pp. 261 – 267, 1990.
21. T. W. Sze, *On solving univariate polynomial equations over finite fields and some related problem*, preprint, available at http://people.apache.org/ szetszwo/umd/papers/poly.pdf.
22. D. H. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Transactions on Information Theory, **32**, pp. 54 – 62, 1986.