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

Ryuichi Harasawa[1], Yutaka Sueyoshi[1], Aichi Kudo[1], and Liang Cui[2]

[1] Faculty of Engineering, Nagasaki University
[2] Graduate School of Science and Technology, 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 for binomials $x^n - a$ over prime fields $\mathbb{F}_p$. We implement the proposed method for various cases and compare the results with the original Berlekamp method. The proposed method can be extended easily to the case where the base field is not a prime field.

**Keywords:** factorization, Berlekamp algorithm, binomial

## 1 Introduction

The factorization of polynomials over finite fields is one of the important topics in computational number theory, for example, it is used in the construction of (non-prime) finite fields and the prime ideal decomposition in number 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) [5, 6]. For the factorization of square-free polynomials over finite fields, the Berlekamp algorithm is well known [2, 6].

In this paper, we propose an improvement of the Berlekamp algorithm for binomials $x^n - a$ over prime fields $\mathbb{F}_p$. More precisely, we give the solution of the equation $h(x)^p \equiv h(x) \pmod{x^n - a}$ directly without applying the sweeping-out method to the corresponding coefficient matrix. We further implement the proposed method for various cases and compare the results with the original Berlekamp method. The proposed method can be extended easily to the case where the base field is not a prime field.

Note that there exist some efficient methods for computing the solution of $x^n = a$ over finite fields (e.g., [1, 7]).

The remainder of this paper is organized as follows: In Section 2, we describe the Berlekamp algorithm. In Section 3, we propose an improvement of the Berlekamp algorithm for binomials $x^n - a$. In Section 4, we implement the original Berlekamp algorithm and the proposed algorithm for binomials, and compare the results. In Section 5, we give the conclusion and future works.

## 2 Berlekamp algorithm

The Berlekamp algorithm [2, 6] is a well-known algorithm for factoring square-free polynomials over finite fields, which we describe in Table 1.

**Table 1.** Berlekamp's algorithm

| | |
|---|---|
| **Input:** | A square-free polynomial $f(x)$ over $\mathbb{F}_p$. |
| **Output:** | The factorization of $f(x)$. |
| **Step 1:** | Compute the polynomials $h(x)$ over $\mathbb{F}_p$ of degree less than $\deg f(x)$ such that $h(x)^p \equiv h(x) \pmod{f(x)}$. The set $V$ of $h(x)$'s above forms an $\mathbb{F}_p$-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:** | **for** $i$ from 1 to $k$ For each $v(x) \in F$ and each $\alpha \in \mathbb{F}_q$, compute $d(x) := \gcd(v(x), h_i(x) - \alpha)$. **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 for** |
| **Step 4:** | Return $F$ (the product of the elements in $F$ equals $f(x)$). |

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

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

for a square-free polynomial $f(x)$ over a prime field $\mathbb{F}_p$. For the linear transformation $h(x) \mapsto h(x)^p \pmod{f(x)}$ on the $n$-dimensional vector

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

We remark that the most time-consuming step of the Berlekamp algorithm is Step 3 in Table 1, which takes exponential time of $\log p$. So both the original method and the proposed method work well only for small fields.

## 3 Proposed algorithm

In this section, we describe an improved method for solving the equation

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

of Step 1 in Table 1. Without loss of generality, we may assume $p \not| n$ because we have $x^{p^i \ell} - a = (x^\ell - a)^{p^i}$. Therefore, the binomial $x^n - a$ is square-free.

Instead of dealing with the coefficient matrix corresponding to the equation above, we consider the orbits of $\mathbb{Z}/n\mathbb{Z}$ with respect to $\langle p \rangle$, the subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$ generated by $p$. Let $\bar{\alpha}$ denote the orbit containing $\alpha \in \mathbb{Z}/n\mathbb{Z}$, that is, $\bar{\alpha} = \{\alpha p^j \bmod n \mid j = 0, 1, 2, \ldots\}$, especially $\bar{0} = \{0\}$. Then $\mathbb{Z}/n\mathbb{Z}$ is the disjoint union of $\bar{\alpha}$'s.

For each orbit $\bar{\alpha}$, let $\bar{\alpha} = \{\alpha_0, \ldots, \alpha_{\ell-1}\}$, where $\alpha_j := \alpha p^j \bmod n$ $(0 \le j \le \ell - 1)$ and $\alpha p^\ell \bmod n = \alpha (= \alpha_0)$. We consider a polynomial $h_{\bar{\alpha}}(x) = \beta_0 x^{\alpha_0} + \beta_1 x^{\alpha_1} + \cdots + \beta_{\ell-1} x^{\alpha_{\ell-1}}$ over $\mathbb{F}_p$. From the definition of $\bar{\alpha}$, we have

$$h_{\bar{\alpha}}(x)^p \equiv a^{\gamma_{\ell-1}} \beta_{\ell-1} x^{\alpha_0} + a^{\gamma_0} \beta_0 x^{\alpha_1} + \cdots + a^{\gamma_{\ell-2}} \beta_{\ell-2} x^{\alpha_{\ell-1}} \pmod{x^n - a}, \tag{3}$$

where $\gamma_i$ is the quotient of $p\alpha_i$ by $n$ (i.e., $p\alpha_i = \gamma_i n + \alpha_{i+1 \bmod \ell}$). In other words, considering $T_{\bar{\alpha}} = \{\beta_0 x^{\alpha_0} + \beta_1 x^{\alpha_1} + \cdots + \beta_{\ell-1} x^{\alpha_{\ell-1}} \mid \beta_i \in \mathbb{F}_p\}$ and the linear transformation $\pi_p : \mathbb{F}_p[x]/(x^n - a) \to \mathbb{F}_p[x]/(x^n - a)$ defined by $h(x) \mapsto h(x)^p \pmod{f(x)}$, we see that $T_{\bar{\alpha}}$ is a $\pi_p$-invariant subspace, that is, $\pi_p(T_{\bar{\alpha}}) \subseteq T_{\bar{\alpha}}$ (in fact, we see $\pi_p(T_{\bar{\alpha}}) = T_{\bar{\alpha}}$), and we have $\mathbb{F}_p[x]/(x^n - a) = \bigoplus_{\bar{\alpha}} T_{\bar{\alpha}}$.

So, letting $k$ be the number of orbits of $\mathbb{Z}/n\mathbb{Z}$ with respect to $\langle p \rangle$, we see that the equation (2) can be devided into $k$ equations in the form

$$h_{\bar{\alpha}}(x)^p \equiv h_{\bar{\alpha}}(x) \pmod{x^n - a} \tag{4}$$

with $h(x) = \sum_{\bar{\alpha}} h_{\bar{\alpha}}(x)$. Namely, we have $V = \bigoplus_{\bar{\alpha}} V_{\bar{\alpha}}$, where $V_{\bar{\alpha}} := V \cap T_{\bar{\alpha}}$.

For the orbit $\bar{\alpha} = \bar{0}$, the solution space

$$V_{\bar{0}} = \{\beta x^0 \mid (\beta x^0)^p \equiv \beta x^0 \pmod{x^n - a},\ \beta \in \mathbb{F}_p\}$$

of the equation (4) becomes $\mathbb{F}_p$, which implies that the dimension of the solution space of the equation (2) over $\mathbb{F}_p$ is at least one.

We consider the case where $\bar{\alpha} \neq \bar{0}$. Comparing the coefficients in both sides of the equation (4), we have

$$\begin{cases} \beta_0 = a^{\gamma_{\ell-1}} \beta_{\ell-1} \\ \beta_1 = a^{\gamma_0} \beta_0 \\ \quad \vdots \\ \beta_{\ell-1} = a^{\gamma_{\ell-2}} \beta_{\ell-2}, \end{cases}$$

which leads to the relation

$$\beta_0 = a^{\gamma_0 + \gamma_1 + \cdots + \gamma_{\ell-1}} \beta_0.$$

Therefore, we obtain the solution(s) of (4) as follows:

$$\begin{cases} 0 & (\text{if } a^{\gamma_0 + \gamma_1 + \cdots + \gamma_{\ell-1}} \neq 1) \\ \beta(x^{\alpha_0} + a^{\gamma_0} x^{\alpha_1} + a^{\gamma_0+\gamma_1} x^{\alpha_2} + \cdots + a^{\gamma_0+\gamma_1+\cdots+\gamma_{\ell-2}} x^{\alpha_{\ell-1}}) & (\text{otherwise}), \end{cases}$$

where $\beta$ runs over all elements of $\mathbb{F}_p$. The solution space $V_{\bar{\alpha}}$ of the equation (4) is $\{0\}$ if $a^{\gamma_0 + \gamma_1 + \cdots + \gamma_{\ell-1}} \neq 1$ and, otherwise, forms one-dimensional subspace of $T_{\bar{\alpha}}$ generated by $x^{\alpha_0} + a^{\gamma_0} x^{\alpha_1} + a^{\gamma_0+\gamma_1} x^{\alpha_2} + \cdots + a^{\gamma_0+\gamma_1+\cdots+\gamma_{\ell-2}} x^{\alpha_{\ell-1}}$.

We describe the proposed algorithm in Table 2.

## 4 Experimental results

In this section, we implement the proposed method for various cases, which is listed in the following tables (Tables 3 – 6). We note some remarks: (1) We compute 100 times for each case and list the average time. (2) The numbers in the parentheses imply the number of irreducible factors of the polynomials to be factored.

All computations are performed on a 3 GHz Pentium IV with 0.99 Gb RAM. The language is C with Borland C++ compiler 5.5.1 and with no mathematical library.

We see, from these tables, that the proposed method is faster than the original one for all cases. Especially, the difference between the running time of these two methods becomes very large in the case where

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

| |
|---|
| **Input:** A binomial $x^n - a$ over $\mathbb{F}_p$ with $p \not| n$.<br>**Output:** A basis $B$ of the solution space $V$ of $h(x)^p \equiv h(x) \pmod{x^n - a}$. |
| **Step 1:** $B \leftarrow \{1\}$,<br>$G \leftarrow \{1, 2, \ldots, n-1\}$. |
| **Step 2:** **if** $G = \emptyset$, return $B$. |
| **Step 3:** $i_0 \leftarrow \min\{i \mid i \in G\}$,<br>$G \leftarrow G \setminus \{i_0\}$,<br>$j \leftarrow i_0$,<br>$f \leftarrow x^j$,<br>$b \leftarrow 1$. |
| **Step 4:** Compute the integers $k,\ r$<br>such that $jp = kn + r$ with $0 \le r < n$.<br>$b \leftarrow b \cdot a^k \bmod p$. |
| **Step 5:** **while** $r \neq i_0$<br>$G \leftarrow G \setminus \{r\}$,<br>$f \leftarrow f + b \cdot x^r$,<br>$j \leftarrow r$.<br>Compute the integers $k,\ r$<br>such that $jp = kn + r$ with $0 \le r < n$.<br>$b \leftarrow b \cdot a^k \bmod p$.<br>**end while** |
| **Step 6:** **if** $b = 1$, $B \leftarrow B \cup \{f\}$.<br>**goto** Step 2. |

the number of irreducible factors of $x^n - a$ is two. We further observe that the running time of both the original method and the proposed one becomes shorter when we rearrange the basis $\{h_1(x), \ldots, h_k(x)\}$ of $V$ so that $\deg h_i(x) \le \deg h_j(x)$ for $1 \le i \le j \le k$ (Step 1 in Table 1).

## 5 Conclusion and future works

In this paper, we described an improvement of the Berlekamp algorithm for binomials $x^n - a$ over prime fields $\mathbb{F}_p$. More precisely, we proposed a method for solving the equation $h(x)^p \equiv h(x) \pmod{x^n - a}$ directly. We leave the comparison of our method with other factorization methods, for example the Cantor and Zassenhaus method [3, 4], and further improvements, for example applications to larger base fields and to more general cases (e.g., trinomials). These are our future works.

**Table 3.** Running time (ms) for factoring $x^n - 1$ over $\mathbb{F}_2$

| the value of $n$ with $x^n - 1$ | 501 | 601 | 701 | 801 | 901 | 1001 | 1101 | 1201 | 1301 | 1401 | 1501 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (number of irreducible factors) | (6) | (25) | (2) | (27) | (12) | (27) | (6) | (5) | (2) | (5) | (10) |
| original method [2] | 7.9 | 25.7 | 8.2 | 19.5 | 38.1 | 51.4 | 45.0 | 69.5 | 32.1 | 60.9 | 132.3 |
| proposed method | 4.0 | 20.9 | 0.1 | 9.8 | 25.0 | 35.4 | 24.0 | 43.4 | 0.4 | 23.5 | 88.9 |

**Table 4.** Running time (ms) for factoring $x^n - 2$ over $\mathbb{F}_3$

| the value of $n$ with $x^n - 2$ | 500 | 601 | 700 | 800 | 901 | 1000 | 1100 | 1201 | 1300 | 1400 | 1501 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (number of irreducible factors) | (14) | (9) | (30) | (18) | (7) | (26) | (62) | (5) | (50) | (54) | (9) |
| original method [2] | 6.5 | 33.2 | 30.0 | 14.3 | 57.8 | 30.6 | 60.6 | 95.1 | 123.2 | 99.6 | 244.8 |
| proposed method | 2.9 | 27.6 | 18.4 | 3.5 | 45.4 | 12.8 | 40.6 | 66.2 | 91.4 | 61.8 | 197.5 |

## 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 – 177, 1977.
2. E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, 1968.
3. D. G. Cantor and H. Zassenhaus, *A new algorithm for factoring polynomials over finite fields*, Math. Comp,, **36**, pp. 587 – 592, 1981.
4. H. Cohen, *A Course in Computational Algebraic Number Theory*, Graduate Texts in Math., vol. **138**, Springer-Verlag, Berlin Heidelberg, 1993.
5. K. Geddes, S. Czapor and G. Labahn, *Algorithms for Computer Algebra*, Kluwer Academic Publishers, 1992.
6. A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
7. 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.

**Table 5.** Running time (ms) for factoring $x^n - 3$ over $\mathbb{F}_5$

| the value of $n$ with $x^n - 3$ | 501 | 602 | 703 | 804 | 901 | 1002 | 1103 | 1204 | 1301 | 1402 | 1503 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (number of irreducible factors) | (5) | (17) | (22) | (21) | (7) | (9) | (2) | (17) | (3) | (5) | (8) |
| original mathod [2] | 13.7 | 50.1 | 61.8 | 25.3 | 100.4 | 60.0 | 25.7 | 95.9 | 121.2 | 128.2 | 97.0 |
| proposed method | 9.3 | 44.2 | 53.9 | 14.0 | 85.3 | 40.4 | 0.9 | 61.7 | 85.3 | 85.0 | 47.5 |

**Table 6.** Running time (ms) for factoring $x^n - 3$ over $\mathbb{F}_7$

| the value of $n$ with $x^n - 3$ | 500 | 600 | 703 | 801 | 904 | 1007 | 1100 | 1200 | 1303 | 1401 | 1504 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (number of irreducible factors) | (46) | (52) | (83) | (2) | (68) | (21) | (78) | (100) | (2) | (3) | (12) |
| original method [2] | 44.3 | 12.1 | 60.1 | 12.9 | 80.3 | 55.0 | 169.0 | 58.9 | 38.9 | 89.2 | 66.2 |
| proposed method | 41.4 | 7.5 | 54.5 | 0.6 | 69.3 | 35.0 | 149.0 | 34.8 | 0.6 | 45.1 | 15.3 |