An Efficient Algorithm for Computing Inverses in $GF(2^m)$ Using Dual Bases

Hyeong Seon Yoo, Seok Ung Yoon, and Eui Sun Kim

Department of Computer Science, Inha University,
Incheon, 402-751, Korea
hsyoo@inha.ac.kr

Abstract. This paper propose a new multiplicative inverse algorithm for Galois field $GF(2^n)$ whose elements are represented by optimal normal bases type II. The efficiency of the arithmetic algorithms depends on the basis and many foregoing papers use either polynomial or optimal normal basis. A normal basis element is always possible to rewrite canonical basis form. The proposed algorithm combines normal basis and canonical basis. It is shown that the suggested algorithm is suitable for implementation and reduces the computation time to 5–10 % of the normal basis algorithm.

1 Introduction

There has been an increasing attention in the design of fast arithmetic operations in the Galois field $GF(2^n)$ which have applications in coding theory and cryptosystem. In these applications, efficient arithmetic algorithms and hardware structures are crucial factors for good performance. The efficiency of these operations is closely related in the way of presenting element bases. Polynomial basis and normal basis are the two presenting methods, and both approach have pros and cones. Basic properties of normal basis and multiplication with minimal coefficients in the finite field $GF(2^n)$ are studied [1–3].

The most popular inversion algorithms are either based on Fermat's little theorem, or on the Euclidean algorithm or related ones [–6]. For extension fields $GF(q^m)$, the Itoh and Tsujii inversion algorithm is very effective [4, 5]. De Win et al. studied software implementation for arithmetic operations in $GF(2^n)$ [7]. Koç et al published papers about multiplication algorithms in finite field [8, 9]. There is also a paper addressing this issue [10]. And software impelmenting issue is covered in a book by Rosing [11].

In this paper we restrict to the extension fields that can be obtained by the optimal normal basis type II. Our idea in this paper is that if we combine the optimal normal basis and the shifted form of the canonical basis. The optimal normal basis can be very effective for squaring operations and the shifted form of the canonical basis can be easily adopted for multiplication process. The suggested inversion algorithm is shown to be efficient and easily implemented.

2 Arithmetic Operations in Optimal Normal Basis

The field $GF(2^n)$ is viewed as an n -dimensional vector space defined over $GF(2)$. And it is well known that a normal basis exists in every finite field [1]. A normal basis in $GF(2^n)$ is given as equation (1).

$$M = \{\beta, \beta^2, \beta^{2^2}, \dots, \beta^{2^{n-1}}\} \text{ for some } \beta \in GF(2^n) \quad (1)$$

An optimal normal basis of type II for this field is constructed using a normal element $\beta = \gamma + \gamma^{-1}$, where γ is a primitive $(2n+1)$ th root of unity and $\gamma^i \neq 1$ for any $1 \leq i < 2n+1$. In this basis, squaring of an element is just a cyclic shift of the basis..

Using this normal element, the normal basis M can be expressed as equation (2).

$$M = \{\gamma + \gamma^{-1}, \gamma^2 + \gamma^{-2}, \gamma^{2^2} + \gamma^{-2^2}, \dots, \gamma^{2^{(n-1)}} + \gamma^{-2^{(n-1)}}\} \quad (2)$$

It is known that a basis element of the form $\gamma^{2^i} + \gamma^{-2^i}$ for $i \in [1, m]$ can be written uniquely as $\gamma^j + \gamma^{-j}$ for $j \in [1, m]$ [8]. The normal basis M has an equivalent form N which is a shifted form of the canonical basis.

$$N = \{\gamma + \gamma^{-1}, \gamma^2 + \gamma^{-2}, \gamma^3 + \gamma^{-3}, \dots, \gamma^n + \gamma^{-n}\} \quad (3)$$

Construction of basis N is done by a systematic transformation of M . If the exponent 2^i is bigger than n , change it by $2n+1-2^i$. The basis N can be simplified using the relation $\beta_i = \gamma^i + \gamma^{-i}$ and a number can be expressed as eq.(4).

The shifted form has a good property over normal basis since it is possible to present index by a small number less than n . The two equivalent bases can be changed.

$$A = \sum_{i=1}^n a_i \beta_i, \quad a_i \in GF(2) \quad (4)$$

Let A and B be expresses in the canonical basis, then the product C can be written as equation (5).

$$\begin{aligned} C &= A \times B \\ &= \left(\sum_{i=1}^n a_i \beta_i \right) \times \left(\sum_{j=1}^n b_j \beta_j \right) \\ &= \sum_{i=1}^n \sum_{j=1}^n a_i b_j (\gamma^{i-j} + \gamma^{-(i-j)}) + \sum_{i=1}^n \sum_{j=1}^n a_i b_j (\gamma^{i+j} + \gamma^{-(i+j)}) \end{aligned} \quad (5)$$

In the above equation if the exponent $i+j$ is greater than n , it will be changed to $2n+1-(i+j)$ by using the identity $\gamma^{2n+1} = 1$.

One simple, inversion algorithm in optimal normal basis is using the Fermat's theorem and continuously factoring the exponent. Let's start with basic notation of inverse of a number A^{-1} .

$$A \cdot A^{-1} = 1 \quad (6)$$

Using the Fermat's theorem the inverse is expressed as (7).

$$A^{-1} = A^{2^n-2} \quad (7)$$

From the exponent, we get equation (8), [11].

$$-1 \equiv 2^n - 2 \pmod{n+1} \quad (8)$$

The equation (8) can be factored as (9).

$$2^n - 2 = 2(2^{n-1} - 1) \quad (9)$$

Let's denote $r_0 = n - 1$. If r_0 is even, it can be factored as eq. (10).

$$2^{r_0} - 1 = (2^{r_0/2} - 1)(2^{r_0/2} + 1) \quad (10)$$

Since it is in the exponent, right side of eq.(7) means eq.(11).

$$\left(\left(A^{2^{r_0/2}-1} \right)^{2^{r_0/2}+1} \right)^2 \quad (11)$$

The above equation means a digit rotation and a multiplication. If it is odd, we can factor as eq. (12).

$$2^{r_0} - 1 = 2 \left(2^{\frac{r_0-1}{2}} - 1 \right) \left(2^{\frac{r_0-1}{2}} + 1 \right) + 1 \quad (12)$$

Since the second-parenthesis term is easy to calculate, we concentrate on the first parenthesis in both equations (10) and (12). If we let the exponent $r_0/2$ or $(r_0-1)/2$ be r_1 , we have the same problem as before. This factoring process continues till the i -th exponent r_i in the first parenthesis equals 1. If we calculate elements in 155 bits, the number of factoring process will be 8. It needs 1 rotation and 1 multiplication for even exponent, and it doubles for odd exponent. So the total arithmetic operations are 11 rotations and 10 multiplications.

3 A New Dual Bases Algorithm

In this paper we suggest to combine the optimal normal basis and the shifted form of the canonical basis. The change of basis is easily done by table lookup process. The computation of inverse needs number of squarings and multiplications. Since two bases have superior properties in squarings and the canonical basis, it is suggested to change basis for two different operations.

The computation steps are followings, Algorithm 1. First check the exponent r_i , continue the process if it is greater than 1. Start with optimal normal basis type and compute a squaring process. Next step is to change the basis into the canonical basis

by table lookup, this process is carried out quickly. Compute the multiplications. If the exponent is odd, change the basis to the optimal type and follow the squaring-multiplication step.

Algorithm 1. A dual bases inversion procedure

```

ON: optimal normal basis,
CB: canonical basis
1. start with  $n - bit$  number  $A$ , ON type;
2. factor as eq. (10) or eq. (12);
3. if the exponent  $r_i > 1$ ;
4.   compute squaring;
5.   change the basis to CB;
6.   compute multiplication;
7. . change the basis to ON;
8.   if  $r_i$  is odd;
9.     compute squaring;
10.    change the basis to CB;
11.    compute multiplication;
12. . change the basis to ON;
13. endif;
14. endif;

```

4 An Example

It is common that the number of bits in elliptic curve cryptography is between 150 and 200. In this range we can construct 15 optimal normal bases and 10 cases belong to type II. We tested number of bits 173, 174, 179, 183, 186, 189 and 191 cases. And two cases of element are selected. For case 1, non-zero bit points are 0, 64 and 128. And the second case, there are 6 non-zero points such as 0, 32, 64, 96, 128 and 160.

Numerical results are obtained by a 800MHz Pentium machine. Table 1 shows for the 3 non-zero bit case, the new algorithm is faster than the conventional one in all number of bit ranges from 2.1 to 20.6 %. Table 2 is for 6 non-zero bit case and the difference between two algorithm decreases, but it shows the nearly the same pattern as in Table 1. The difference is about 2.3 ~ 17.7 %.

Table 1. Inversion results for $A = \beta^{2^{128}} + \beta^{2^{64}} + \beta^{2^0}$ (m sec).

No. of bits	Conventional algorithm	New algorithm	% decrease
173	1.85	1.75	5.4
174	1.99	1.58	20.6
179	1.88	1.84	2.1
183	2.11	1.77	16.1
186	2.11	1.85	12.3
189	2.18	1.91	12.4
191	2.40	1.93	19.6

Table 2. Inversion results for $A = \beta^{2^{160}} + \beta^{2^{128}} + \beta^{2^{96}} + \beta^{2^{64}} + \beta^{2^{32}} + \beta^{2^0}$ (m sec).

No. of bits	Conventional algorithm	New algorithm	% decrease
173	1.85	1.74	5.9
174	2.03	1.67	17.7
179	1.92	1.79	6.8
183	2.10	1.81	13.8
186	2.14	2.09	2.3
189	2.17	1.83	15.6
191	2.38	2.25	5.4

5 Conclusions

In this paper we suggested a new inversion algorithm combining two types of bases, the optimal normal basis and the canonical basis. The computation time for the new method decreased about 2.1 ~ 20.6 % less for 3 non-zero bit case. And it decreased about 2.3 ~ 17.7 % for 6 non-zero bit case. The difference between two algorithms are less for the 6 non-zero bit case, but it shows the same tendency. The suggested new algorithm is easy to implement and can be applied in any cryptosystems.

Acknowledgements. This research was supported by University IT Research Center Project.

References

1. R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*, Cambridge University Press, Cambridge, 1986
2. J. Omura and J. Massey, "Computational methods and apparatus for finite field arithmetic," U.S. Patent No. 4587627, 1986
3. R. C. Mullin, I. M. Onyszchuk and S. A. Vanstone, "Optimal normal bases in $\text{GF}(p^n)^*$," *Discrete Applied Mathematics*, 22, 149–161, 1988
4. T. Itoh and S. Tsujii, "A fast algorithm for computing multiplicative inverses in $\text{GF}(2^m)$ using normal bases," 78, 171–177, 1988
5. J. Juajardo and C. Paar, "Itoh-Tsujii inversion in standard basis and its application in cryptography and codes," 25, 207–216, 2002
6. M. A. Hansen, M. Z. Wang and V. K. Bhargava, "A modified Massey-Omura parallel multiplier for a class of finite field," *IEEE Trans. On Computers*, 42, 10, 1278–1280, 1993
7. E. De Win, A. Bosselaers, S. Vandenberghe, P. D. Gerssem and J. Vanderwalle, "A fast software implementation for arithmetic operations in $\text{GF}(2^k)$," *ASIACRYPT '96*, LNCS 1233, 65–76, 1996
8. C. K. Koç and T. Acar, "Montgomery multiplication in $\text{GF}(2^k)$," *Design, Codes and Cryptography*, 14, 1, 57–69, 1998
9. B. Sunar and C. K. Koç, "An efficient optimal normal basis type II multiplier," *IEEE Trans. on Computers*, 50, 1, 83–87, 2001
10. D. V. Bailey and C. Paar, "Optimal extension fields for fast arithmetic in public-key algorithms," *CRYPTO '98*, LNCS 1462, 472–485, 1998
11. M. Rosing, *Implementing elliptic curve cryptography*, Manning Publ. Co., Greenwich, CT, 1999