Mixed Bases for Efficient Inversion in $\mathbb{F}_{((2^2)^2)^2}$ and Conversion Matrices of SubBytes of AES

Yasuyuki Nogami, Kenta Nekado, Tetsumi Toyota, Naoto Hongo, and Yoshitaka Morikawa

Graduate School of Natural Science and Technology, Okayama University
3-1-1, Tsushima–naka, Kita–ku, Okayama, Okayama 700-8530, Japan
nogami@trans.cne.okayama-u.ac.jp

Abstract. A lot of improvements and optimizations for the hardware implementation of SubBytes of Rijndael, in detail *inversion* in $\mathbb{F}_{2^8}$ have been reported. Instead of the Rijndael original $\mathbb{F}_{2^8}$, it is known that its isomorphic tower field $\mathbb{F}_{((2^2)^2)^2}$ has a more efficient inversion. For the towerings, several kinds of bases such as polynomial and normal bases can be used in *mixture*. Different from the meaning of this *mixture* of bases, this paper proposes another *mixture* that contributes to the reduction of the critical path delay of SubBytes. To the $\mathbb{F}_{(2^2)^2}$–inversion architecture, for example, the proposed *mixture* inputs and outputs elements represented with normal and polynomial bases, respectively.

1 Introduction

SubBytes of the Advanced Encryption Standard (AES), that is Rijndael, uses arithmetic operations in $\mathbb{F}_{2^8}$, especially *inversion* [8]. From the viewpoint of *hardware implementation*, it is said that *tower field* technique efficiently works and then a lot of efficient techniques have been reported [4, 9]. In detail, instead of the Rijndael original $\mathbb{F}_{2^8}$, its *isomorphic* tower field $\mathbb{F}_{((2^2)^2)^2}$ is efficiently applied for calculating an inversion in SubBytes. According to Canright's work [2], there are 432 possible combinations of the modular polynomials and bases for constructing tower field $\mathbb{F}_{((2^2)^2)^2}$. Morioka et al's work [7] adopted only polynomial bases and Canright's work [2] did only normal bases; however, the difference causes little influence for the critical path delays. For example, another efficient construction that is introduced at **Sec**. 2.4 of this paper has the same *critical path delay*. It uses two normal bases and one polynomial basis for the towering bases in *mixture*. Different from the meaning of this *mixture* of bases, this paper proposes to use normal and polynomial bases in *mixture*.

When the tower field $\mathbb{F}_{((2^2)^2)^2}$ is used in SubBytes, it needs certain conversion matrices between the Rijndael original $\mathbb{F}_{2^8}$ and the tower field $\mathbb{F}_{((2^2)^2)^2}$. A few papers [2, 6] have discussed the efficiency of conversion matrices. Most of those previous works just discuss the number of 1's in the conversion matrices; however, this paper focuses on their critical path delays only, in detail, the Hamming weights of the row vectors of the conversion matrices. It has been experimentally shown that there are some rare conversion matrices whose row vectors all have the Hamming weights less than or equal to 4. It is very important for

the hardware implementation. For such efficient conversion matrices, Canright's approach [2] such as *greedy* algorithm and *tree structure analysis* will be also applied to decrease the number of 1's in the matrices.

The *mixture* of bases proposed in this paper, in brief *mixed bases*, means the following usage of two different bases such as polynomial and normal bases. Let $A = a_0\beta + a_1\beta^4, a_0, a_1 \in \mathbb{F}_{2^2}$ be a non–zero element represented with normal basis $\{\beta, \beta^4\}$ in $\mathbb{F}_{(2^2)^2}$, where β is a zero of $g(x) = x^2 + x + \alpha$ and α is a zero of $e(x) = x^2 + x + 1$. Then, calculate its inverse $D = A^{-1}$ in $\mathbb{F}_{(2^2)^2}$ as

$$D = A^{-1} = (a_0\beta + a_1\beta^4)^{-1} = d_0 + d_1\beta,\ d_0, d_1 \in \mathbb{F}_{2^2}. \tag{1}$$

The most important point is that the input A is represented with normal basis $\{\beta, \beta^4\}$ but the output D is represented with polynomial basis $\{1, \beta\}$. This paper especially applies the *mixed bases* to the inversions in $\mathbb{F}_{(2^2)^2}$ and $\mathbb{F}_{((2^2)^2)^2}$. It is shown that the former contributes to the reduction of the critical path delay of $\mathbb{F}_{((2^2)^2)^2}$–inversion and the latter connects the $\mathbb{F}_{((2^2)^2)^2}$–inversion to some efficient conversion matrices. As previously introduced, the conversion matrices have smaller critical path delays and they are quite rare cases. In addition, it is shown that the use of the *mixed bases* has little influence to the number of gates needed for the logical architectures but reduces the critical path delays.

2 Preliminaries

This section briefly introduces the conventional construction of an inversion in tower field $\mathbb{F}_{((2^2)^2)^2}$ for the use in S-Box (SubBytes) of AES. In detail, let us review the adopted bases, modular polynomials, calculation procedure of an inversion in $\mathbb{F}_{((2^2)^2)^2}$, and then Morioka's work [7], Canrigt's work [2], and another efficient construction. Since the tower field is used with two conversion matrices for the isomorphism between $\mathbb{F}_{2^8}$ and $\mathbb{F}_{((2^2)^2)^2}$, the conventional viewpoints of the efficiency of conversion matrices are also introduced.

2.1 Extension field $\mathbb{F}_{2^8}$ and its tower construction $\mathbb{F}_{((2^2)^2)^2}$

8-bit inputs and outputs of the S-Box are dealt as elements in binary field of extension degree 8, that is $\mathbb{F}_{2^8}$. Among the arithmetic operations in the binary field, *inversion* plays an important role in SubBytes. In detail, the SubBytes calculates the multiplicative inverse A^{-1} of a non–zero input element $A \in \mathbb{F}_{2^8}^*$ and then carries out a certain Affine transformation. For the preparation of $\mathbb{F}_{2^8}$, AES [8] originally adopts an irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ as the modular polynomial; however, it is well known that its isomorphic *tower* construction $\mathbb{F}_{((2^2)^2)^2}$ achieves a more efficient inversion together with Itoh–Tsujii inversion algorithm (ITA) [5]. In detail, first construct $\mathbb{F}_{2^2}$ by using the irreducible polynomial $e(x) = x^2 + x + 1$ over $\mathbb{F}_2$ [1], then construct $\mathbb{F}_{(2^2)^2}$ by using a certain irreducible polynomial $f(x)$ of degree 2 over $\mathbb{F}_{2^2}$, and then construct $\mathbb{F}_{((2^2)^2)^2}$ by using a certain irreducible polynomial $g(x)$ of degree 2 over $\mathbb{F}_{(2^2)^2}$. Thus, the efficiencies of the arithmetic operations in $\mathbb{F}_{((2^2)^2)^2}$ are closely related to the

[1] Any other irreducible polynomials of degree 2 over $\mathbb{F}_2$ does not exist.

selection of the modular polynomials and the bases for the towerings. For example, polynomial and normal bases are efficient for multiplication and Frobenius mapping, respectively. In the case that the characteristic is equal to 2 such as AES, Frobenius mapping is equivalent to squaring.

2.2 Morioka's construction [7]

Conventional works such as [4] have often referred to Morioka et al.'s construction [7] for achieving efficient inversion in $\mathbb{F}_{((2^2)^2)^2}$. Morioka's work [7] adopts $e(x) = x^2+x+1$ with its polynomial basis $\{1, \alpha\}$ for $\mathbb{F}_{2^2}$, $f(x) = x^2+x+\alpha$ with its polynomial basis $\{1, \beta\}$ for $\mathbb{F}_{(2^2)^2}$, and $g(x) = x^2 + x + \lambda,\ \lambda = \alpha^2\beta$ with its polynomial basis $\{1, \gamma\}$ for $\mathbb{F}_{((2^2)^2)^2}$, where $\alpha \in \mathbb{F}_{2^2}$, $\beta \in \mathbb{F}_{(2^2)^2}$, and $\gamma \in \mathbb{F}_{((2^2)^2)^2}$ are zeros of $e(x)$, $f(x)$, and $g(x)$, respectively. Note that it adopts polynomial bases for all towerings. Its *critical path delays* are summarized in **Table** 1.

2.3 Canright's construction [2]

Different from Morioka's work, Canright's work [2] adopts $e(x) = x^2+x+1$ with its normal basis $\{\alpha, \alpha^2\}$ for $\mathbb{F}_{2^2}$, $f(x) = x^2+x+\alpha$ with its normal basis $\{\beta, \beta^4\}$ for $\mathbb{F}_{(2^2)^2}$, and $g(x) = x^2 + x + \lambda,\ \lambda = \alpha^2\beta$ with its normal basis $\{\gamma, \gamma^{16}\}$ for $\mathbb{F}_{((2^2)^2)^2}$, where $\alpha \in \mathbb{F}_{2^2}$, $\beta \in \mathbb{F}_{(2^2)^2}$, and $\gamma \in \mathbb{F}_{((2^2)^2)^2}$ are zeros of $e(x)$, $f(x)$, and $g(x)$, respectively. It is noted that it adopts normal bases for all towerings. Its *critical path delays* are summarized in **Table** 1.

2.4 Another efficient construction

This section introduces another efficient construction. Different from Morioka's and Canright's works, it adopts $e(x) = x^2 + x + 1$ with its normal basis $\{\alpha, \alpha^2\}$ for $\mathbb{F}_{2^2}$, $f(x) = x^2+x+\alpha$ with its polynomial basis $\{1, \beta\}$ for $\mathbb{F}_{(2^2)^2}$, and $g(x) = x^2 + x + \lambda,\ \lambda = \alpha^2\beta$ with its normal basis $\{\gamma, \gamma^{16}\}$ for $\mathbb{F}_{((2^2)^2)^2}$, where $\alpha \in \mathbb{F}_{2^2}$, $\beta \in \mathbb{F}_{(2^2)^2}$, and $\gamma \in \mathbb{F}_{((2^2)^2)^2}$ are zeros of $e(x)$, $f(x)$, and $g(x)$, respectively. Its *critical path delays* are summarized in **Table** 1.

The improvements proposed in this paper are started from this construction, thus in what follows let us briefly review its arithmetic operations in $\mathbb{F}_{2^2}$, $\mathbb{F}_{(2^2)^2}$, and $\mathbb{F}_{((2^2)^2)^2}$. Their calculation architectures are summarized in **App**. A.

Arithmetic operations in $\mathbb{F}_{2^2}$ In the same of Canright's work [2], construct $\mathbb{F}_{2^2}$ with the modular polynomial $e(x) = x^2 + x + 1$ and its normal basis $\{\alpha, \alpha^2\}$ as follows. According to the coefficients of $e(x)$ whose zero is α, $\alpha + \alpha^2 = 1$ and $\alpha^3 = 1$. Let $A = a_0\alpha + a_1\alpha^2$, $B = b_0\alpha + b_1\alpha^2$, $a_0, a_1, b_0, b_1 \in \mathbb{F}_2$, a multiplication $C = AB$ becomes as follows (**Fig**. 7).

$$\begin{aligned} AB &= (a_0\alpha + a_1\alpha^2)(b_0\alpha + b_1\alpha^2) \\ &= a_1b_1\alpha + a_0b_0\alpha^2 + (a_1b_0 + a_0b_1)(\alpha + \alpha^2) \\ &= \{(a_0 + a_1)(b_0 + b_1) + a_0b_0\}\,\alpha + \{(a_0 + a_1)(b_0 + b_1) + a_1b_1\}\,\alpha^2 \\ &= c_0\alpha + c_1\alpha^2 = C. \end{aligned} \tag{2}$$

For a non–zero element A in $\mathbb{F}_{2^2}$, Frobenius mapping with respect to $\mathbb{F}_2$, that is squaring, is equivalent to inversion as follows (**Fig**. 8).

$$A^2 = A^{-1} = (a_0\alpha + a_1\alpha^2)^2 = a_0\alpha^2 + a_1\alpha^4 = a_1\alpha + a_0\alpha^2. \tag{3}$$

Times α and times α^2 are carried out as follows (**Fig**. 9).

$$\alpha A = a_0\alpha^2 + a_1\alpha^3 = a_1\alpha + (a_0 + a_1)\alpha^2, \tag{4a}$$
$$\alpha^2 A = a_0\alpha^3 + a_1\alpha^4 = (a_0 + a_1)\alpha + a_0\alpha^2. \tag{4b}$$

Arithmetic operations in $\mathbb{F}_{(2^2)^2}$ In the same of Morioka et al.'s work [7], construct $\mathbb{F}_{(2^2)^2}$ with the modular polynomial $g(x) = x^2 + x + \alpha$ and its polynomial basis $\{1, \beta\}$. Thus, the arithmetic operations and calculation procedures become as follows. Let $A = a_0 + a_1\beta,\ B = b_0 + b_1\beta, a_0, a_1, b_0, b_1 \in \mathbb{F}_{2^2}$, a multiplication $C = AB$ in $\mathbb{F}_{(2^2)^2}$ is carried out as follows (**Fig**. 10).

$$\begin{aligned} AB &= (a_0 + a_1\beta)(b_0 + b_1\beta) \\ &= (a_0b_0 + a_1b_1\alpha) + \{(a_0 + a_1)(b_0 + b_1) + a_0b_0\}\beta \\ &= c_0 + c_1\beta = C. \end{aligned} \tag{5}$$

Frobenius mapping of A with respect to $\mathbb{F}_{2^2}$, that is 4–th power operation, becomes as follows.

$$A^{2^2} = a_0 + a_1\beta^4 = a_0 + a_1(\beta + 1) = (a_0 + a_1) + a_1\beta. \tag{6}$$

The square of A is calculated as follows (**Fig**. 11).

$$A^2 = a_0^2 + a_1^2\beta^2 = a_0^2 + a_1^2(\beta + \alpha) = (a_0^2 + a_1^2\alpha) + a_1^2\beta. \tag{7}$$

Let A be a non–zero element in $\mathbb{F}_{(2^2)^2}$, its inverse $D = A^{-1}$ is calculated by ITA as follows (**Fig**. 12).

$$\begin{aligned} A^{-1} &= (AA^4)^{-1}A^4 \\ &= \left\{(a_0 + a_1\beta)(a_0 + a_1\beta^4)\right\}^{-1}((a_0 + a_1) + a_1\beta) \\ &= \left\{a_0(a_0 + a_1) + a_1^2\alpha\right\}^{-1}((a_0 + a_1) + a_1\beta) \\ &= d_0 + d_1\beta = D. \end{aligned} \tag{8}$$

Times $\lambda = (\alpha + 1)\beta = \alpha^2\beta$, that is the constant term of the modular polynomial $g(x)$, is carried out as follows (**Fig**. 13).

$$\alpha^2\beta A = a_0\alpha^2\beta + a_1\alpha^2\beta^2 = a_0\alpha^2\beta + a_1\alpha^2(\beta + \alpha) = a_1 + (a_0 + a_1)\alpha^2\beta. \tag{9}$$

Inversion in $\mathbb{F}_{((2^2)^2)^2}$ In the same of Canright's construction [2], construct $\mathbb{F}_{((2^2)^2)^2}$ with the modular polynomial $g(x) = x^2 + x + \lambda,\ \lambda = \alpha^2\beta$ with its

normal basis $\{\gamma, \gamma^{16}\}$. Let $A = a_0\gamma + a_1\gamma^{16}$, $a_0, a_1 \in \mathbb{F}_{(2^2)^2}$ be a non–zero element in $\mathbb{F}_{((2^2)^2)^2}$, ITA calculates its inverse $D = A^{-1}$ as follows (**Fig**. 14).

$$\begin{aligned} A^{-1} &= (AA^{16})^{-1}A^{16} \\ &= \left\{a_0a_1(\gamma+\gamma^{16})^2 + (a_0^2+a_1^2)\gamma\gamma^{16}\right\}^{-1}\left(a_1\gamma + a_0\gamma^{16}\right) \\ &= \left\{a_0a_1 + (a_0+a_1)^2\lambda\right\}^{-1}\left(a_1\gamma + a_0\gamma^{16}\right) \\ &= d_0\gamma + d_1\gamma^{16} = D. \end{aligned} \tag{10}$$

Efficiencies of various tower fields One of typical features of Morioka's work [7] is that all of the towering bases are polynomial bases such as $\{1, \alpha\}$ for $\mathbb{F}_{2^2}$. As introduced in Canright's work [2], not only polynomial bases but also normal bases are available for the towering bases and it is said that there are 432 possible combinations. Canright's work [2] has introduced an efficient construction of tower field $\mathbb{F}_{((2^2)^2)^2}$ that uses normal bases for all towerings. As introduced in [2], it will be one of the best combinations for tower field $\mathbb{F}_{((2^2)^2)^2}$; however, such *good* constructions of *inversion* in tower field $\mathbb{F}_{((2^2)^2)^2}$ have a comparable compactness. According to his detail report [3], the best *inversion* introduced in [2] and that shown in **Sec**. 2.4 have almost the same compactness. In addition, the improvements and optimizations introduced in Morioka et al.'s and Canright's works [7], [2] will be also efficiently applied to the inversions shown in this paper. Thus, this paper focuses on the inversion in $\mathbb{F}_{((2^2)^2)^2}$ and the conversion matrices with the viewpoint of *critical path delay* and without discussing the compactness.

2.5 Conversion matrices with the viewpoint of conjugates

As shown in **Fig**. 1 and the following **Eqs**. (12), when the inversion in the isomorphic *tower* field $\mathbb{F}_{((2^2)^2)^2}$ is applied to SubBytes instead of that of the Rijndael original $\mathbb{F}_{2^8}$, the input 8–bit vector needs to be converted to the corresponding element in $\mathbb{F}_{((2^2)^2)^2}$. Then, after calculating its inverse in $\mathbb{F}_{((2^2)^2)^2}$, the result needs to be returned to the Rijndael–original vector representation. Thus, two conversion matrices together with a certain Affine transformation are required before and after the inversion in $\mathbb{F}_{((2^2)^2)^2}$ (**Fig**. 1).

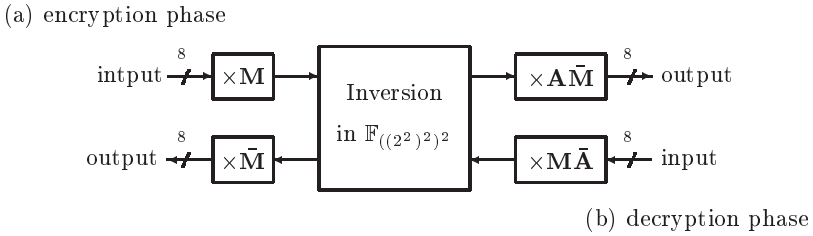


Fig. 1. Sharing the inversion for encryption/decryption with conversion matrices

In detail, let $\{1, \omega, \cdots, \omega^6, \omega^7\}$ be the polynomial basis of $\mathbb{F}_{2^8}$, where ω is a zero of the modular polynomial $x^8 + x^4 + x^3 + x + 1$, the Rijndael originally represents 8–bit vector as an element $\tilde{X}$ in $\mathbb{F}_{2^8}$ as follows.

$$\tilde{X} = \tilde{x}_0 + \tilde{x}_1\omega + \cdots + \tilde{x}_6\omega^6 + \tilde{x}_7\omega^7 = (\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_6, \tilde{x}_7). \tag{11}$$

Then, SubBytes for *encryption* phase calculates

$$\tilde{Z}^{\mathcal{T}} = \mathbf{A}\left(\bar{\mathbf{M}}\left(\left(\mathbf{M}\tilde{X}^{\mathcal{T}}\right)^{-1}\right)\right) + (0,1,1,0,0,0,1,1)^{\mathcal{T}}, FS \tag{12a}$$

where $\mathbf{M}$, $\bar{\mathbf{M}} = \mathbf{M}^{-1}$, and $\mathbf{A}$ denote the conversion, inverse conversion, and Affine transformation matrices, respectively. Thus, $X = \mathbf{M}\tilde{X}$ becomes an element in the tower field $\mathbb{F}_{((2^2)^2)^2}$ and then its inverse X^{-1} is efficiently calculated in $\mathbb{F}_{((2^2)^2)^2}$. As understood from **Eq**. (12a), $\mathbf{A}\bar{\mathbf{M}}$ is precomputed.

Inversely, SubBytes for *decryption* phase calculates

$$\tilde{X}^{\mathcal{T}} = \bar{\mathbf{M}}\left(\left(\mathbf{M}\left(\bar{\mathbf{A}}\tilde{Z}^{\mathcal{T}} + (0,0,0,0,0,1,0,1)^{\mathcal{T}}\right)\right)^{-1}\right).BS \tag{12b}$$

In this case, $Z = \mathbf{M}\left(\bar{\mathbf{A}}\tilde{Z}^{\mathcal{T}} + (0,0,0,0,0,1,0,1)^{\mathcal{T}}\right)$ becomes an element in the tower field $\mathbb{F}_{((2^2)^2)^2}$ and then its inverse Z^{-1} is efficiently calculated in $\mathbb{F}_{((2^2)^2)^2}$. In the same of the encryption phase, $\mathbf{M}\bar{\mathbf{A}}$ and $\mathbf{M}(0,0,0,0,0,1,0,1)^{\mathcal{T}}$ are precomputed. Note here that the inversions in *encryption* phase **Eq**. (12a) and *decryption* phase **Eq**. (12b) can be carried out in the same procedure such as **Fig**. 14. Thus, previous works such as Canright's [2] works have mostly focused on the compact construction of inversion in tower field $\mathbb{F}_{((2^2)^2)^2}$ but not together with the efficiency of the conversion matrices in detail.

In the case of the efficient construction shown in **Sec**. 2.4, for example, the conversion matrices are given as follows (**Table** 1).

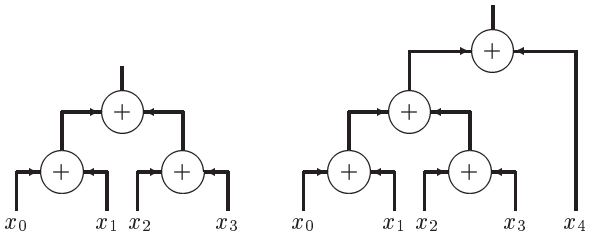
$$\mathbf{M} = \begin{bmatrix} 1&0&1&0&0&1&0&0 \\ 1&0&0&0&1&0&1&0 \\ 0&0&1&1&1&1&0&0 \\ 0&1&0&1&0&1&0&0 \\ 1&1&0&1&0&1&0&0 \\ 1&0&1&1&0&0&0&1 \\ 0&1&0&0&0&1&1&1 \\ 0&0&1&0&1&0&1&0 \end{bmatrix} \cdot \quad \mathbf{A}\bar{\mathbf{M}} = \begin{bmatrix} 1&1&0&1&1&1&0&0 \\ 0&1&1&1&0&1&1&1 \\ 0&1&0&0&0&1&1&0 \\ 1&0&0&1&1&1&1&1 \\ 0&1&1&0&0&0&1&0 \\ 1&0&0&0&0&0&1&0 \\ 0&0&0&1&1&0&1&0 \\ 0&1&1&1&0&0&1&1 \end{bmatrix} \cdot \tag{13a}$$

$$\bar{\mathbf{M}} = \begin{bmatrix} 0&1&0&0&0&0&0&0 \\ 1&0&1&0&1&0&1&0 \\ 1&0&1&0&0&1&1&1 \\ 0&1&1&0&1&0&1&1 \\ 1&0&1&0&0&1&0&0 \\ 1&0&0&1&0&1&1&0 \\ 0&1&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&1 \end{bmatrix} \cdot \quad \mathbf{M}\bar{\mathbf{A}} = \begin{bmatrix} 1&1&0&1&0&0&0&0 \\ 1&0&1&0&0&1&0&0 \\ 1&1&0&0&1&0&0&1 \\ 1&1&0&1&0&1&1&1 \\ 1&0&0&1&0&0&0&0 \\ 0&1&0&1&0&0&1&1 \\ 1&1&0&0&0&0&0&0 \\ 0&0&0&1&1&0&0&0 \end{bmatrix} \cdot \tag{13b}$$

Efficiency of conversion matrices These conversion matrices are easily determined but they are not uniquely determined because the modular polynomials such as $e(x) = x^2 + x + 1$ have conjugate elements as zeros. In detail, in the case of **Sec**. 2.2, since α has its conjugate α^2 with respect to $\mathbb{F}_2$, $\{1, \alpha^2\}$ can be the basis of $\mathbb{F}_{2^2}$. In the same, $\{1, \beta^4\}$ and $\{1, \gamma^{16}\}$ can be the *towering* bases of $\mathbb{F}_{(2^2)^2}$ and $\mathbb{F}_{((2^2)^2)^2}$, respectively. Thus, there are 8 variants for each matrix and they play the same role on the connection to Rijndael original $\mathbb{F}_{2^8}$. Most of previous works such as Mentens's work [6] have basically focused on the number of 1's in the conversions matrices to evaluate their efficiencies.

This paper focuses on that every Hamming weight of row vectors of **M** shown in **Eq**. (13a) is smaller than or equal to 4. It is very important for the hardware implementation. For example, let us consider the following vector multiplications (inner products). Its hardware calculation will be implemented as **Fig**. 2.

$$(1,1,1,1,0,0,0,0)(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)^{\mathcal{T}}, \tag{14a}$$
$$(1,1,1,1,1,0,0,0)(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)^{\mathcal{T}}. \tag{14b}$$



Eq. (14a) **Eq**. (14b)

Fig. 2. Implementations of **Eq**. (14a) and **Eq**. (14b)

Thus, in the case of **Eq**. (13a), since every Hamming weight of row vectors of **M** is smaller than or equal to 4, it is efficiently implemented as shown in **Fig**. 2 and then its critical path delay becomes 2 T_{X}, where in what follows T_X and T_A denote the delays of XOR and AND, respectively. Such an efficient conversion matrix is a quite rare case, therefore, as shown in **Eqs**. (13), **M** has the efficiency but the other matrices such as $\mathbf{A}\bar{\mathbf{M}}$ do not (**Table** 1).

Since it has been introduced that the Hamming weights of the matrices are reduced by some techniques such as *tree structure* [2], this paper does not discuss the weights of matrices into detail. Then, from the viewpoint of *critical path delay*, this paper proposes an efficient inversion in $\mathbb{F}_{((2^2)^2)^2}$ and conversion matrices to which *polynomial* and *normal* bases are used in *mixture*.

Table 1. Comparison of the efficiencies of three constructions

construction		# of 1's	critical path delay[‡]
Morioka et al. [7]	$\mathbf{M}$	32	$3\ T_{\mathrm{X}}$
	$\mathbf{A}\bar{\mathbf{M}}$	29	$3\ T_{\mathrm{X}}$
	inv. in $\mathbb{F}_{((2^2)^2)^2}$	–	$17\ T_{\mathrm{X}} + 4\ T_{\mathrm{A}}$
	$\bar{\mathbf{M}}$	27	$\mathbf{2}\ T_{\mathrm{X}}$
	$\mathbf{M}\bar{\mathbf{A}}$	29	$3\ T_{\mathrm{X}}$
Canright [2]	$\mathbf{M}$	32	$3\ T_{\mathrm{X}}$
	$\mathbf{A}\bar{\mathbf{M}}$	25	$3\ T_{\mathrm{X}}$
	inv. in $\mathbb{F}_{((2^2)^2)^2}$	–	$\mathbf{15}\ T_{\mathrm{X}} + \mathbf{4}\ T_{\mathrm{A}}$
	$\bar{\mathbf{M}}$	29	$3\ T_{\mathrm{X}}$
	$\mathbf{M}\bar{\mathbf{A}}$	26	$3\ T_{\mathrm{X}}$
another efficient construction	$\mathbf{M}$	28	$\mathbf{2}\ T_{\mathrm{X}}$
	$\mathbf{A}\bar{\mathbf{M}}$	33	$3\ T_{\mathrm{X}}$
	inv. in $\mathbb{F}_{((2^2)^2)^2}$	–	$\mathbf{15}\ T_{\mathrm{X}} + \mathbf{4}\ T_{\mathrm{A}}$
	$\bar{\mathbf{M}}$	31	$3\ T_{\mathrm{X}}$
	$\mathbf{M}\bar{\mathbf{A}}$	26	$3\ T_{\mathrm{X}}$

3 Main proposal

This paper proposes an efficient architecture for *inversion* in tower field $\mathbb{F}_{((2^2)^2)^2}$ to which, different from Morioka et al.'s proposal [7] and Canright's approaches [2], polynomial and normal bases are used in *mixture*, in brief *mixed bases*. Especially based on the *inversion* in $\mathbb{F}_{((2^2)^2)^2}$ constructed as **Fig**. 14, the *mixed bases* are mainly applied to two calculation parts: I_4 and I_8. In detail, denote their new versions by $\hat{I}_4$ and $\hat{I}_8$, respectively,

- $\hat{I}_4$ has the input and output for $\mathbb{F}_{(2^2)^2}$–elements represented with normal basis $\{\beta, \beta^4\}$ and polynomial basis $\{1, \beta\}$, respectively,
- $\hat{I}_8$ has the input and output for $\mathbb{F}_{((2^2)^2)^2}$–elements represented with normal basis $\{\gamma, \gamma^{16}\}$ and polynomial basis $\{1, \gamma\}$, respectively.

Then, the critical path delay for *encryption* phase of SubBytes of AES becomes

$$2\ T_{\mathrm{X}} + (\ 14\ T_{\mathrm{X}} + 4\ T_{\mathrm{A}}\ ) + 2\ T_{\mathrm{X}}. \tag{15}$$

Together with the meaning of the *mixed* bases, in what follows, several improvements using *mixed bases* especially at I_4 and I_8 are shown in detail. Note here that the modular polynomials and bases are as introduced in **Sec**. 2.4.

3.1 Mixed bases for I_4 of Fig. 14

As also introduced in [2], it is often said that *inversion* with normal basis is more efficient than that with polynomial basis because several Frobenius mappings

are needed in ITA–based inversion. Inversely, it is often said that *multiplication* with polynomial basis is more efficient than that with normal basis because Karatsuba–based multiplication needs polynomial multiplications [1].

First, let us consider an *inversion* in $\mathbb{F}_{(2^2)^2}$ with the normal basis $\{\beta, \beta^4\}$, where β is a zero of $g(x) = x^2 + x + \alpha$. Let $A = a_0\beta + a_1\beta^4$ be a non–zero element in $\mathbb{F}_{(2^2)^2}$, its inverse $D = A^{-1}$ is calculated by ITA as follows.

$$\begin{aligned} A^{-1} &= (AA^4)^{-1}A^4 \\ &= \left\{(a_0\beta + a_1\beta^4)(a_1\beta + a_0\beta^4)\right\}^{-1}\left(a_1\beta + a_0\beta^4\right) \\ &= \left\{a_0a_1 + (a_0 + a_1)^2\alpha\right\}^{-1}\left(a_1\beta + a_0\beta^4\right) \\ &= d_0\beta + d_1\beta^4 = D. \end{aligned} \tag{16a}$$

However, the following multiplications in $\mathbb{F}_{(2^2)^2}$ denoted by M_4 in **Fig**. 14 cannot accept $\mathbb{F}_{(2^2)^2}$–elements represented with the normal basis. Because, they accept ones represented with the polynomial basis $\{1, \beta\}$. Thus, consider the following inversion in $\mathbb{F}_{(2^2)^2}$ with a non–zero element $A = a_0\beta + a_1\beta^4$.

$$\begin{aligned} A^{-1} &= (AA^4)^{-1}A^4 \\ &= \left\{(a_0\beta + a_1\beta^4)(a_1\beta + a_0\beta^4)\right\}^{-1}\left(a_1\beta + a_0\beta^4\right) \\ &= \left\{a_0a_1 + (a_0 + a_1)^2\alpha\right\}^{-1}((a_0 + a_1) + a_0\beta) \\ &= d_0 + d_1\beta = D. \end{aligned} \tag{16b}$$

Based on **Eq**. (16b), the calculation architecture of the new version $\hat{I}_4$ is constructed as **Fig**. 3. It is the meaning of *mixed bases*. If I_4 in **Fig**. 14 that is constructed with the polynomial basis $\{1, \beta\}$ is replaced to the *inversion* with normal basis $\{\beta, \beta^4\}$, that is denoted by $\hat{I}_4$ (**Fig**. 3), the critical path delay of I_8 constructed as **Fig**. 14 is reduced to $14\ T_{\mathrm{X}} + 4\ T_{\mathrm{A}}$.

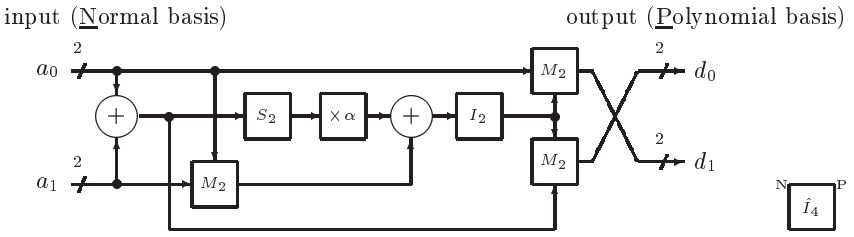


Fig. 3. Inversion in $\mathbb{F}_{(2^2)^2}$ with normal and polynomial bases ($\hat{I}_4$)

On the other hand, $\hat{I}_4$ (**Fig**. 3) needs a non–zero input represented with the normal basis $\{\beta, \beta^4\}$ in $\mathbb{F}_{(2^2)^2}$. Without increasing the critical path delay, it needs two changes at $\times\lambda$ and M_4 in **Fig**. 14 before the inversion in $\mathbb{F}_{(2^2)^2}$. Their

output elements are originally represented with the polynomial basis $\{1, \beta\}$. Thus, change them so as to output $\mathbb{F}_{(2^2)^2}$–elements represented with the *normal basis* $\{\beta, \beta^4\}$. In detail, let $A = a_0 + a_1\beta,\ B = b_0 + b_1\beta, a_0, a_1, b_0, b_1 \in \mathbb{F}_{2^2}$ and based on the following calculations, their new versions denoted by $\times\hat{\lambda}$ and $\hat{M}_4$ are constructed as **Fig**. 4 and **Fig**. 5, respectively.

$$\begin{aligned} \lambda A &= a_0\alpha^2\beta + a_1\alpha^2\beta^2 \\ &= \{a_1 + (a_0 + a_1)\alpha^2\}\beta + a_1\beta^4 \\ &= (a_1\alpha + a_0\alpha^2)\beta + a_1\beta^4. \end{aligned} \tag{17}$$

$$\begin{aligned} AB &= (a_0 + a_1\beta)(b_0 + b_1\beta) \\ &= \{(a_0 + a_1)(b_0 + b_1) + a_1b_1\alpha\}\,\beta + (a_0b_0 + a_1b_1\alpha)\beta^4 \\ &= c_0\beta + c_1\beta^4 = C. \end{aligned} \tag{18}$$

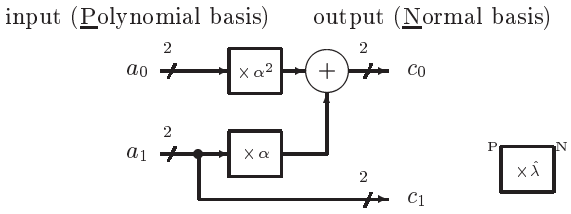


Fig. 4. Times λ in $\mathbb{F}_{(2^2)^2}$ with polynomial and normal bases ($\times\hat{\lambda}$)

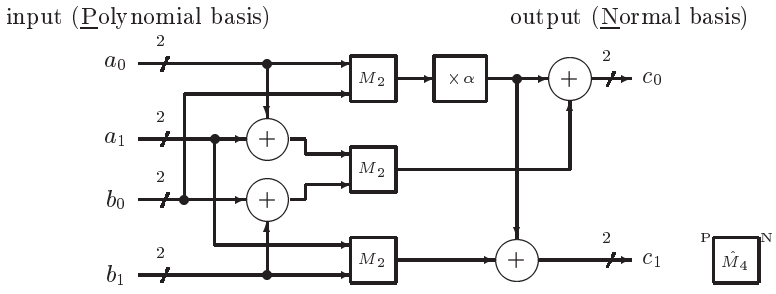


Fig. 5. Multiplication in $\mathbb{F}_{(2^2)^2}$ with polynomial and normal bases ($\hat{M}_4$)

3.2 Mixed bases for the inversion in $\mathbb{F}_{((2^2)^2)^2}$

The input and output elements for the inversion architecture constructed as **Fig**. 14 both need to be represented with the normal basis $\{\gamma, \gamma^{16}\}$. However, this paper changes only the representation of the output element to that with the polynomial basis $\{1, \gamma\}$. In detail, let $A = a_0\gamma + a_1\gamma^{16}$, $a_0, a_1 \in \mathbb{F}_{(2^2)^2}$ be a non–zero element in $\mathbb{F}_{((2^2)^2)^2}$, based on ITA, calculate its inverse $D = A^{-1}$ as

$$\begin{aligned} A^{-1} &= (AA^{16})^{-1}A^{16} \\ &= \left\{a_0a_1(\gamma+\gamma^{16})^2 + (a_0^2+a_1^2)\gamma\gamma^{16}\right\}^{-1}\left(a_1\gamma + a_0\gamma^{16}\right) \\ &= \left\{a_0a_1 + (a_0+a_1)^2\lambda\right\}^{-1}\left\{a_0 + (a_0+a_1)\gamma\right\} \\ &= d_0 + d_1\gamma = D. \end{aligned} \tag{19}$$

Note that, for a non–zero input represented with the normal basis $\{\gamma, \gamma^{16}\}$, it calculates its inverse represented with the polynomial basis $\{1, \gamma\}$. **Fig**. 6 shows its calculation architecture to which $\hat{I}_4$, $\hat{M}_4$, and $\times\hat{\lambda}$ are also applied.

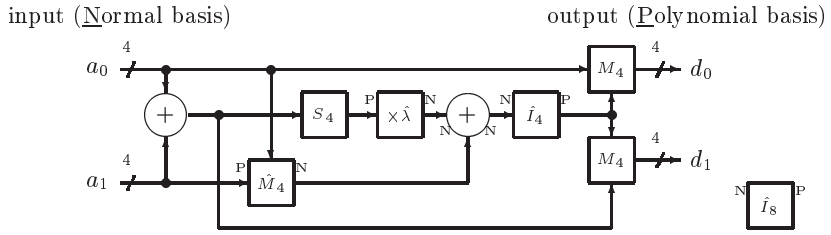


Fig. 6. Inversion in $\mathbb{F}_{((2^2)^2)^2}$ with normal and polynomial bases

As previously introduced, this inversion achieves $14\,T_{\mathrm{X}} + 4\,T_{\mathrm{A}}$; however, the last *mixed bases* used in **Eq**. (19) is not related to this efficiency. It is related to the efficiency of the conversion matrices. When the output is represented with the normal basis $\{\gamma, \gamma^{16}\}$, the calculated inverse A^{-1} is multiplied by the conversion matrix $\mathbf{A}\bar{\mathbf{M}}$ shown in **Eqs**. (13a). On the other hand, in the case of the inversion constructed as **Fig**. 6, since the output is represented with the polynomial basis $\{1, \gamma\}$, it needs to be multiplied by the following conversion matrix $\mathbf{A}\bar{\mathbf{M}}\mathbf{M}'$,

$$\mathbf{A}\bar{\mathbf{M}} \times \mathbf{M}' = \begin{bmatrix} 1&1&0&1&1&1&0&0 \\ 0&1&1&1&0&1&1&1 \\ 0&1&0&0&0&1&1&0 \\ 1&0&0&1&1&1&1&1 \\ 0&1&1&0&0&0&1&0 \\ 1&0&0&0&0&0&1&0 \\ 0&0&0&1&1&0&1&0 \\ 0&1&1&1&0&0&1&1 \end{bmatrix} \times \begin{bmatrix} 1&0&0&0&1&0&0&0 \\ 0&1&0&0&0&1&0&0 \\ 0&0&1&0&0&0&1&0 \\ 0&0&0&1&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \end{bmatrix} = \begin{bmatrix} 0&0&0&1&1&1&0&1 \\ 0&0&0&0&0&1&1&1 \\ 0&0&1&0&0&1&0&0 \\ 0&1&1&0&1&0&0&1 \\ 0&1&0&0&0&1&1&0 \\ 1&0&1&0&1&0&0&0 \\ 1&0&1&1&0&0&0&1 \\ 0&1&0&0&0&1&1&1 \end{bmatrix}, \tag{20a}$$

where $\mathbf{M}'$ is given by

$$\mathbf{M}' = \begin{bmatrix} 1&0&0&0&1&0&0&0 \\ 0&1&0&0&0&1&0&0 \\ 0&0&1&0&0&0&1&0 \\ 0&0&0&1&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \end{bmatrix} \tag{20b}$$

and it converts the vector representation with the polynomial basis $\{1, \gamma\}$ to that with the normal basis $\{\gamma, \gamma^{16}\}$. According to **Eq**. (20a), the conversion matrix $\mathbf{A}\bar{\mathbf{M}}\mathbf{M}'$ after the inversion in $\mathbb{F}_{((2^2)^2)^2}$ shown in **Fig**. 6 fortunately has the efficiency introduced in **Sec**. 2.5. Such an efficient conversion matrix is a quite rare case and it is experimentally found. Thus, the last *mixed bases* shown in **Fig**. 6 is just for obtaining this efficient conversion matrix $\mathbf{A}\bar{\mathbf{M}}\mathbf{M}'$.

3.3 Evaluation

Finally, the proposed architecture with conversion matrices, especially its *encryption* phase, has the critical path delays shown in **Table** 2.

Table 2. Critical path delays of the proposed architecture

construction			# of 1's	critical path delay[‡]
proposal	$\mathbf{M}$	**Eq**. (13a)	28	$\mathbf{2}\ T_{\mathrm{X}}$
	$\mathbf{A}\bar{\mathbf{M}}\mathbf{M}'$	**Eq**. (20a)	27	$\mathbf{2}\ T_{\mathrm{X}}$
	inv. in $\mathbb{F}_{((2^2)^2)^2}$	**Fig**. 6	–	$\mathbf{14}\ T_{\mathrm{X}} + \mathbf{4}\ T_{\mathrm{A}}$

According to the result, this paper could show that the *mixed bases* contributes to some improvements of SubBytes of AES with tower field technique.

4 Conclusion and future work

This paper has proposed an efficient architecture for *inversion* in tower field $\mathbb{F}_{((2^2)^2)^2}$ to which, different from the conventional works, polynomial and normal bases are used in *mixture*, in brief *mixed bases*. Then, this paper has especially shown some improvements of the inversion architecture in $\mathbb{F}_{((2^2)^2)^2}$ and the conversion matrices in the *encryption* phase. As a future work, using *mixed bases*, those in the *decryption* phase should be also improved. Then, the detailed comparison with some other efficient implementations is needed. After that, a consideration for *side channel attacks* will be also required.

Acknowledgments

The authors would like to thank the anonymous referees for detailed review. We adequately appreciate their observations and helpful suggestions.

References

1. D. Bailey and C. Paar, "Optimal Extension Fields for Fast Arithmetic in Public–Key Algorithms, " Crypto'98, LNCS 1462, pp. 472–485, Springer–Verlag, 1998.
2. D. Canright, "A Very Compact S-Box for AES," Workshop on Cryptographic Hardware and Embedded Systems (CHES2005), LNCS 3659, pp. 441–455, Springer–Verlag, 2005.
3. D. Canright, Naval Postgraduate School Technical Report: NPS–MA–05–001, 2005. http://web.nps.navy.mil/ dcanrig/pub/NPS-MA-05-001.pdf
4. D. Canright and L. Batina, "A Very Compact "Perfectly Masked" S–Box for AES," Applied Cryptography and Network Security (ACNS2008), LNCS 5037, pp. 446-459, Springer–Verlag, 2008.
5. T. Itoh and S. Tsujii, "A Fast Algorithm for Computing Multiplicative Inverse in GF(2^m) using Normal Basis," Inf. Comput., vol. 78, pp. 171–177, 1988.
6. N. Mentens, Secure and Efficient Coprocessor Design for Cryptographic Applications on FPGAs, Doctor thesis, Katholieke Universiteit Leuven, 2007.
7. S. Morioka and A. Satoh, "An optimized S–box circuit arthitecture for low power AES design," Workshop on Cryptographic Hardware and Embedded Systems (CHES2002), LNCS 2523, pp. 172–186, Springer–Verlag, 2003.
8. National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), FIPS publication 197, http://csrc.nist.gov/encryption/aes/index.html, 2001.
9. A. Satoh, S. Morioka, K. Takano, and Seiji Munetoh, "A compact Rijndael hardware architecture with S-box optimization," ASIACRYPT 2001, LNCS 2248, pp. 239–254, Springer–Verlag, 2001.

A Architectures of the construction shown in Sec. 2.4

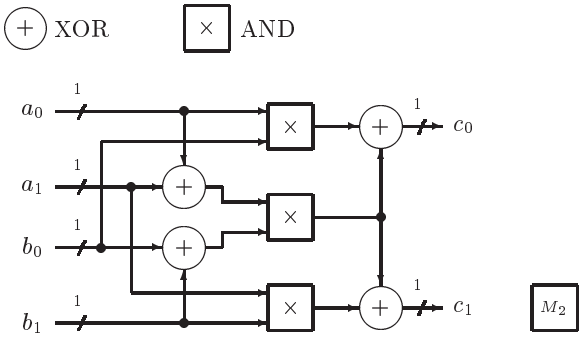


Fig. 7. Multiplication in $\mathbb{F}_{2^2}$ (M_2)

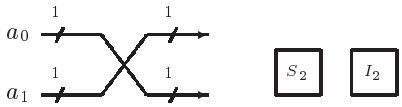


Fig. 8. Squaring (Frobenius mapping) in $\mathbb{F}_{2^2}$ $(S_2,\ I_2)$

Fig. 9. Times α and times α^2 in $\mathbb{F}_{2^2}$ $(\times\alpha,\ \times\alpha^2)$

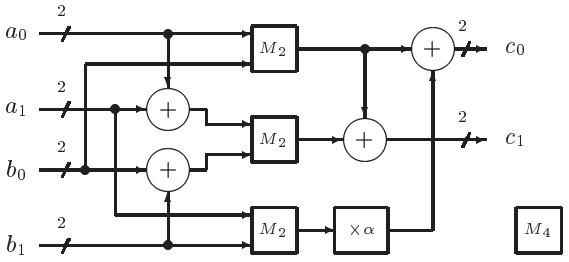


Fig. 10. Multiplication in $\mathbb{F}_{(2^2)^2}$ (M_4)

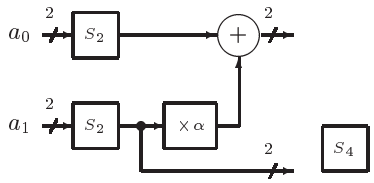


Fig. 11. Squaring in $\mathbb{F}_{(2^2)^2}$ (S_4)

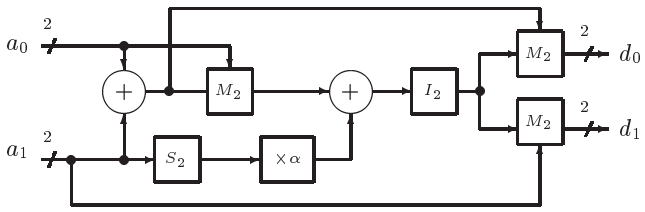


Fig. 12. Inversion in $\mathbb{F}_{(2^2)^2}$ (I_4)

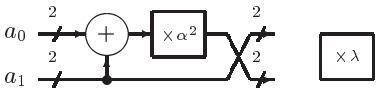


Fig. 13. Times λ in $\mathbb{F}_{(2^2)^2}$ $(\times\lambda)$

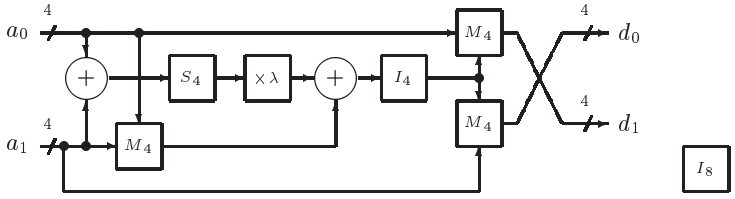


Fig. 14. Inversion in $\mathbb{F}_{((2^2)^2)^2}$