Hardware Implementation of Compact AES S-box

Xiaoqiang ZHANG, Ning WU, Gaizhen YAN, and Liling DONG

***Abstract*—In this paper, a detailed study on compact Advanced Encryption Standard (AES) S-box implementation has been proposed. Firstly, We firstly map the Multiplicative Inverse (MI) in S-box from finite filed GF(2^8) to composite field GF(($(2^4)^2$) by Composite Field Arithmetic (CFA) technology, and derive out the expressions for all sub-operations over GF(2^4) in MI. Then a novel multi-term-pattern common subexpression elimination algorithm is proposed to reduce redundant resources in these sub-operations over GF(2^4) and isomorphic mapping matrices. Both polynomial basis and normal basis for CFA are discussed in this paper. The optimization results show that the S-boxes based on normal basis has smaller area and shorter critical path than the one based on polynomial basis. Compared with previous works, the normal basis based S-boxes proposed in this paper can achieve the shortest critical path and the minimal area-delay-product.**

***Index Terms*—Advanced Encryption Standard (AES), S-box, Composite Field Arithmetic (CFA), Common Subexpression Elimination (CSE), polynomial basis, normal basis**

I. Introduction

INFORMATION security issues have become increasingly prominent as information technology developed [1]. Data encrypted by ciphers is widely used to ensure security [2-3]. Advanced Encryption Standard (AES) is established by the National Institute of Standards and Technology (NIST) to replace the original Data Encryption Standard (DES) in 2001 [4]. It is one of the most important symmetric block ciphers. The block length of the AES algorithm is 128 bits with the key lengths of 128, 192 or 256 bits. Full computation of the AES encryption requires 10, 12 or 14 rounds, and each round contains four transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey. The SubBytes transformation, commonly known as S-box, is a nonlinear substitution that guarantees a better security of the AES encryption. It takes the most resources and consumes the most power in the implementation. Hence, the hardware implementation efficiency of the AES encryption in terms of area, speed, security and power consumption mainly depends on the implementation of S-box [5].

In general, the AES encryption is applied in resource-limited systems, such as wireless sensor networks [6]-[7] and radio frequency identifiers [8], in which the compact AES hardware implementation is highly desirable. Among these implementations, S-boxes implemented with composite field arithmetic (CFA) have the smallest size [9]-[10]. Over years, several CFA-based S-boxes have been proposed [11]-[19], and they usually base on polynomial basis and normal basis [11]. Summarizing from the previous works [12], Canright presented the smallest S-box based on normal basis [13]. However, the critical path was long in Canright's work. On the other hand, Zhang *et al.* proposed an AES S-box based on polynomial basis with the shortest critical path to date [14]. However, their work requires a large area.

In this paper, to reduce the area consumption and shorten the critical path required for AES S-box, CFA technology is employed to map the Multiplicative Inverse (MI) in AES S-box from finite filed $GF(2^8)$ to composite field $GF(((2^4)^2)$. Expressions of all sub-operations over $GF(2^4)$ in MI are derived out based on polynomial basis and normal basis, respectively. Furthermore, a new *Multi-Term-Pattern* Common Subexpression Elimination (MTP-CSE) algorithm is proposed to reduce redundant resources in these sub-operations and isomorphic mapping matrices.

The rest of this paper is organized as follows. Based on polynomial basis and normal basis, two architectures for hardware implementation of S-box are derived in Section II. In section III, the expressions for all sub-operations over $GF(2^4)$ in MI are derived out. A new MTP-CSE algorithm is proposed to reduce redundant resources in the sub-operations over $GF(2^4)$ and isomorphic mapping matrices. In Section V, two architectures of S-box are evaluated and compared with previous works. Conclusions are given in Section VI.

II. Composite Field Implementation of S-box

The AES S-box is defined as the MI module over the finite field $GF(2^8)$ followed by an affine transformation [15]. The S-box is calculated by using

$$F = M(X^{-1}) + V, \tag{1}$$

where X is 8-bit input vector, M is an 8×8 constant binary matrix, and V is an 8-bit constant vector. M and V are defined as

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad V = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

This work was supported by the National Natural Science Foundation of China (61376025), Industry-academic Joint Technological Innovations Fund Project of Jiangsu Province (BY2013003-11) he Funding of Jiangsu Innovation Program for Graduate Education (No. KYLX_0273), and the Fundamental Research Funds for the Central Universities.

Xiaoqiang ZHANG, Ning WU, Gaizhen YAN, and Liling DONG are with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing 210016, China (e-mail: zxq198111@qq.com; wunee@nuaa.edu.cn; xucs_yan@126.com; 820365078@qq.com).

The process of affine transformation is defined as follows: X^{-1} is multiplied by M first, and then added with a constant vector V.

This study focuses mainly on approaches to simplify the MI over $GF(2^8)$. The architecture of the S-box using the CFA technique is shown in Fig. 1.

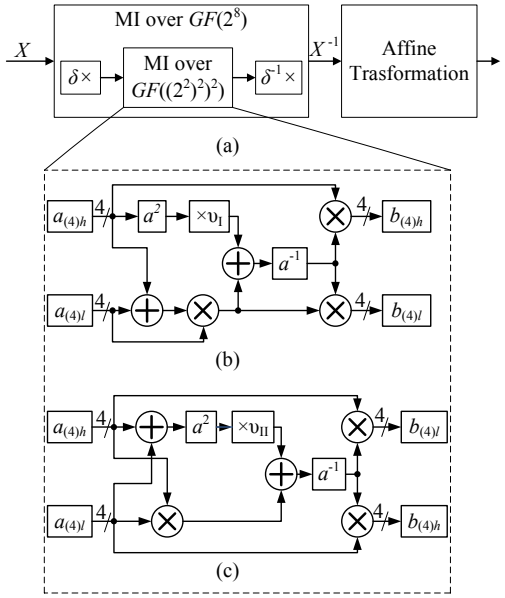

Fig. 1 Architecture of S-box using the CFA technique: (a) architecture of S-box; (b) architecture of MI based on polynomial basis; (c) architecture of MI based on normal basis.

As shown in Fig. 1, the computation of S-box includes two parts, namely the MI over $GF(2^8)$ and the affine transformation. The MI over $GF(2^8)$ can be decomposed into composite field $GF(((2^2)^2)^2)$ to reduce the hardware complexity by CFA technology [14]. The decomposing is based either on polynomial basis or on normal basis, for easy presentation in the following, the polynomial basis based S-box is denoted as Case I, and the normal basis based S-box, as Case II. An isomorphic mapping matrix is required to map the input vector from finite field $GF(2^8)$ to the composite field $GF(((2^2)^2)^2)$, and its inverse matrix is required to revert the computing results to $GF(2^8)$. The S-box based on the CFA technique can be expressed as follows

$$F = M(\delta^{-1}(\delta X)^{-1}) + V, \tag{2}$$

where δ is the isomorphic mapping matrix and δ^{-1} is inverse matrix of δ, both them are 8×8 binary matrix. Usually, matrix δ^{-1} and matrix M are merged into a single matrix $M\delta^{-1}$ to reduce hardware resource.

The irreducible polynomial for composite field $GF((2^4)^2)$, $GF((2^2)^2)$ and $GF(2^2)$ are denoted as follows [13]:

$$\begin{cases} GF((2^4)^2): f_1(x) = x^2 + \tau x + \upsilon \\ GF((2^2)^2): f_2(x) = x^2 + Tx + N, \\ GF(2^2): f_3(x) = x^2 + x + 1 \end{cases} \tag{3}$$

where coefficients $\{\tau, \upsilon\} \in GF(2^4)$, coefficients $\{T, N\} \in GF(2^2)$, and the values of coefficients $\{\tau, \upsilon\}$ must be satisfied that $f_1(x)$ is irreducible over composite field $GF((2^4)^2)$, and the values of coefficients $\{T, N\}$ must be satisfied that $f_2(x)$ is irreducible over composite field $GF((2^2)^2)$. Coefficients $\{\tau, T\}$ are usually opted unity to simplify the architecture of MI [13].

According CFA technology, the MI over $GF((2^4)^2)$ in Case I [14] and Case II [13] can be expressed as in (4) and (5), respectively,

$$\begin{aligned} A_{(8)\rm I}^{-1}(X) = &\left(a_{(4)h}^2\upsilon_{\rm I} + a_{(4)l}\left(a_{(4)h} + a_{(4)l}\right)\right)^{-1} \\ &\times\left(a_{(4)h}X + \left(a_{(4)h} + a_{(4)l}\right)\right), \end{aligned} \tag{4}$$

$$\begin{aligned} A_{(8)\rm II}^{-1}(Y) = &\left(\left(a_{(4)h} + a_{(4)l}\right)^2\upsilon_{\rm II} + a_{(4)h}a_{(4)l}\right)^{-1} \\ &\times\left(a_{(4)l}Y^{16} + a_{(4)h}Y\right), \end{aligned} \tag{5}$$

where $A_{(8)\rm I}(X)=a_{(4)h}X+a_{(4)l}$, $A_{(8)\rm II}(Y)=a_{(4)h}Y^{16}+a_{(4)l}Y$, and where $A_{(8)} \in GF(2^8)$, $\{a_{(4)h}, a_{(4)l}\} \in GF(2^4)$, and polynomial basis ($X$, 1) and normal basis ($Y^{16}$, Y) for $GF((2^4)^2)$ are chosen in (4) and (5), respectively. The corresponding architecture of MI in Case I and Case II are shown in Fig. 1(b) and Fig. 1(c), respectively.

In the $GF(2^p)$ field, the additions are defined as bitwise exclusive OR operations, then the additions in Fig. 1(b) and Fig. 1(c) can be easily implemented by XOR gates. The architectures of other operation modules are depended on the value of irreducible polynomial coefficients in (3). We choose $\{\upsilon_{\rm I}=(1100)_2, N_{\rm I}=(10)_2\}$ in Case I [14], and $\{\upsilon_{\rm II}=(0001)_2, N_{\rm II}=(10)_2\}$ in Case II [13] to illustrate our methods in this paper.

III. Sub-Operations over GF(2^8) in MI

In this section, the expressions of sub-operations over $GF(2^8)$ in MI are derived out based on CFA technology. The sub-operations include multiplication, MI, square, and Constant Coefficient Multiplication (CCM).

A. Multiplication over $GF(2^4)$

Multiplication over $GF(2^4)$ can be decomposed into composite field $GF((2^2)^2)$ to implement, and the expressions in Case I [14] and in Case II [13] are represented in (6) and (7), respectively.

$$\begin{aligned} &A_{(4)\rm I}(X)B_{(4)\rm I}(X) \\ &= \left(a_{(2)h}b_{(2)h} + \left(a_{(2)h} + a_{(2)l}\right)\left(b_{(2)h} + b_{(2)l}\right)\right)X \\ &\quad + \left(a_{(2)h}b_{(2)h}N_{\rm II} + a_{(2)l}b_{(2)l}\right), \end{aligned} \tag{6}$$

$$\begin{aligned} &A_{(4)\rm II}(Y)B_{(4)\rm II}(Y) \\ &= \left(a_{(2)h}b_{(2)h} + \left(a_{(2)h} + a_{(2)l}\right)\left(b_{(2)h} + b_{(2)l}\right)N_{\rm II}\right)Y^4 \\ &\quad + \left(a_{(2)l}b_{(2)l} + \left(a_{(2)h} + a_{(2)l}\right)\left(b_{(2)h} + b_{(2)l}\right)N_{\rm II}\right)Y, \end{aligned} \tag{7}$$

where $\begin{cases} A_{(4)\rm I}(X) = a_{(2)h}X + a_{(2)l} \\ B_{(4)\rm I}(X) = b_{(2)h}X + b_{(2)l} \end{cases}$, $\begin{cases} A_{(4)\rm II}(Y) = a_{(2)h}Y^4 + a_{(2)l}Y \\ B_{(4)\rm II}(Y) = b_{(2)h}Y^4 + b_{(2)l}Y \end{cases}$,

and where $\{A_{(4)}, B_{(4)}\} \in GF(2^4)$, $\{a_{(2)h}, a_{(2)l}, b_{(2)h}, b_{(2)l}\} \in GF(2^2)$, and polynomial basis ($X$, 1) and normal basis ($Y^4$, Y) for $GF((2^2)^2)$ is chosen in (6) and (7), respectively.

The multiplication over $GF(2^2)$ can be further decomposed into $GF((2)^2)$, and the expressions in Case I [14] and in Case II [13] are represented in (8) and (9), respectively.

$$A_{(2)\mathrm{I}}(X)B_{(2)\mathrm{I}}(X) = \left(a_h b_h + \left(a_h + a_l\right)\left(b_h + b_l\right)\right)X + \left(a_h b_h + a_l b_l\right), \tag{8}$$

$$A_{(2)\mathrm{II}}(Y)B_{(2)\mathrm{II}}(Y) = \left(a_h b_h + \left(a_h + a_l\right)\left(b_h + b_l\right)\right)Y^2 + \left(a_l b_l + \left(a_h + a_l\right)\left(b_h + b_l\right)\right)Y, \tag{9}$$

where $\begin{cases} A_{(2)\mathrm{I}}(X) = a_h X + a_l \\ B_{(2)\mathrm{I}}(X) = b_h X + b_l \end{cases}$, $\begin{cases} A_{(2)\mathrm{II}}(Y) = a_h Y^2 + a_l Y \\ B_{(2)\mathrm{II}}(Y) = b_h Y^2 + b_l Y \end{cases}$, and where $\{A_{(2)}, B_{(2)}\} \in GF(2^2)$, $\{a_h, a_l, b_h, b_l\} \in GF(2)$, and polynomial basis $(X, 1)$ and normal basis (Y^2, Y) for $GF((2)^2)$ is chosen in (8) and (9), respectively.

Over the $GF(2)$ field, a multiplication is defined as a AND operation. Substituting (8) in (6), and substituting (9) in (7), we can obtain that the expressions of multiplication over $GF(2^4)$ in Case I and in Case II are given by (10) and (11), repectively, where $\{a_3, a_2, a_1, a_0\}$ are four bits of $A_{(4)} \in GF(2^4)$, so as $B_{(4)}$ and $C_{(4)}$.

B. MI over $GF(2^4)$

MI over $GF((2^2)^2)$ in Case I [14] and in Case II [13] are represented in (12) and (13), respectively,

$$A_{(4)\mathrm{I}}^{-1}(X) = \left(a_{(2)h}^2 N_{\mathrm{I}} + a_{(2)l}\left(a_{(2)h} + a_{(2)l}\right)\right)^{-1} \times \left(a_{(2)h}X + \left(a_{(2)h} + a_{(2)l}\right)\right), \tag{12}$$

$$A_{(4)\mathrm{II}}^{-1}(Y) = \left(\left(a_{(2)h} + a_{(2)l}\right)^2 N_{\mathrm{II}} + a_{(2)h}a_{(2)l}\right)^{-1} \times \left(a_{(2)l}Y^4 + a_{(2)h}Y\right). \tag{13}$$

MI over $GF(2^2)$ can be further decomposed into $GF((2)^2)$, the expressions in Case I [14] and in Case II [13] are represented in (14) and (15), respectively,

$$A_{(2)\mathrm{I}}^{-1}(X) = \left(a_h^2 + a_l\left(a_h + a_l\right)\right)^{-1}\left(a_h X + \left(a_h + a_l\right)\right) = a_h X + \left(a_h + a_l\right), \tag{14}$$

$$A_{(2)\mathrm{II}}^{-1}(Y) = \left(\left(a_h^2 + a_l^2\right) + a_h a_l\right)^{-1}\left(a_l Y^2 + a_h Y\right) = \left(a_l Y^2 + a_h Y\right), \tag{15}$$

where over $GF(2)$, $a = a^{-1} = a^2$, $a \in GF(2)$. According to (8) and (14), we can obtain that the expression of MI over $GF(2^4)$ in Case I is given by

TABLE I
THE OPTIMIZATION RESULTS OF $\times\upsilon$ AND a^2 OPERATION

Approach	a^2 and $\times\upsilon$	Resources Consumption	Critical Path
Case I	Separate	8 XOR	4 XOR
	Merged	4 XOR	2 XOR
Case II	Separate	13 XOR	4 XOR
	Merged	3 XOR	1 XOR

$$D_{(4)\mathrm{I}} = A_{(4)\mathrm{I}}^{-1} = \begin{cases} d_3 = a_3a_2a_1 + a_3a_0 + a_3 + a_2 \\ d_2 = a_3a_2a_1 + a_3a_2a_0 + a_3a_0 + a_2a_1 + a_2 \\ d_1 = a_3a_2a_1 + a_3a_1a_0 + a_2a_0 + a_3 + a_2 + a_1 \\ d_0 = a_3a_2a_1 + a_3a_2a_0 + a_3a_1a_0 + a_2a_1a_0 + a_3a_1 \\ \quad + a_3a_0 + a_2a_1 + a_2 + a_1 + a_0 \end{cases}, \tag{16}$$

and according (9) and (15), the expression of MI over $GF(2^4)$ is given by

$$D_{(4)\mathrm{II}} = A_{(4)\mathrm{II}}^{-1} = \begin{cases} d_3 = a_2a_1a_0 + a_3a_1 + a_2a_1 + a_1 + a_0 \\ d_2 = a_3a_1a_0 + a_3a_1 + a_2a_1 + a_2a_0 + a_0 \\ d_1 = a_3a_2a_0 + a_3a_1 + a_3a_0 + a_3 + a_2 \\ d_0 = a_3a_2a_1 + a_3a_1 + a_3a_0 + a_2a_0 + a_2 \end{cases}, \tag{17}$$

C. Square over $GF(2^4)$ and CCM over $GF(2^4)$

From the (8) and (9), we can obtain that square over $GF(2^4)$ in Case I and in Case II can be expressed as follows:

$$E_{(4)\mathrm{I}} = A_{(4)\mathrm{I}}^2 = \begin{cases} e_3 = a_3 \\ e_2 = a_3 + a_2 \\ e_1 = a_2 + a_1 \\ e_0 = a_3 + a_1 + a_0 \end{cases}, \tag{18}$$

$$E_{(4)\mathrm{II}} = A_{(4)\mathrm{II}}^2 = \begin{cases} e_3 = a_3 + a_2 + a_1 \\ e_2 = a_2 + a_1 + a_0 \\ e_1 = a_3 + a_1 + a_0 \\ e_0 = a_3 + a_2 + a_0 \end{cases}. \tag{19}$$

And CCM over $GF(2^4)$ in Case I and in Case II can be expressed as follows:

$$C_{(4)\mathrm{I}} = A_{(4)\mathrm{I}}B_{(4)\mathrm{I}} = \begin{cases} c_3 = a_0b_0 + \left(a_1 + a_0\right)\left(b_1 + b_0\right) + \left(a_2 + a_0\right)\left(b_2 + b_0\right) + \left(a_3 + a_2 + a_1 + a_0\right)\left(b_3 + b_2 + b_1 + b_0\right) \\ c_2 = a_1b_1 + a_0b_0 + \left(a_3 + a_1\right)\left(b_3 + b_1\right) + \left(a_2 + a_0\right)\left(b_2 + b_0\right) \\ c_1 = a_3b_3 + a_0b_0 + \left(a_3 + a_2\right)\left(b_3 + b_2\right) + \left(a_1 + a_0\right)\left(b_1 + b_0\right) \\ c_0 = a_2b_2 + a_1b_1 + a_0b_0 + \left(a_3 + a_2\right)\left(b_3 + b_2\right) \end{cases}. \tag{10}$$

$$C_{(4)\mathrm{II}} = A_{(4)\mathrm{II}}B_{(4)\mathrm{II}} = \begin{cases} c_3 = a_3b_3 + \left(a_3 + a_2\right)\left(b_3 + b_2\right) + \left(a_2 + a_0\right)\left(b_2 + b_0\right) + \left(a_3 + a_2 + a_1 + a_0\right)\left(b_3 + b_2 + b_1 + b_0\right) \\ c_2 = a_2b_2 + \left(a_3 + a_2\right)\left(b_3 + b_2\right) + \left(a_2 + a_0\right)\left(b_2 + b_0\right) + \left(a_3 + a_1\right)\left(b_3 + b_1\right) \\ c_1 = a_1b_1 + \left(a_1 + a_0\right)\left(b_1 + b_0\right) + \left(a_2 + a_0\right)\left(b_2 + b_0\right) + \left(a_3 + a_2 + a_1 + a_0\right)\left(b_3 + b_2 + b_1 + b_0\right) \\ c_0 = a_0b_0 + \left(a_1 + a_0\right)\left(b_1 + b_0\right) + \left(a_2 + a_0\right)\left(b_2 + b_0\right) + \left(a_3 + a_1\right)\left(b_3 + b_1\right) \end{cases}. \tag{11}$$

$$F_{(4)\mathrm{I}}=A_{(4)\mathrm{I}}v_{\mathrm{I}}=\begin{cases} f_3=a_2+a_0 \\ f_2=a_3+a_2+a_1+a_0 \\ f_1=a_3 \\ f_0=a_2 \end{cases}, \quad (20)$$

$$F_{(4)\mathrm{II}}=A_{(4)\mathrm{II}}v_{\mathrm{II}}=\begin{cases} f_3=a_3+a_1 \\ f_2=a_2+a_0 \\ f_1=a_3+a_0 \\ f_0=a_2+a_1+a_0 \end{cases}. \quad (21)$$

Square over $GF(2^4)$ and CCM over $GF(2^4)$ can be merged into one single module $a^2\times v$ to save resources consumption and shorten critical path. According (18) and (20), the ${A_{(4)\mathrm{I}}}^2\times v_{\mathrm{I}}$ can be expressed as follows:

$$G_{(4)\mathrm{I}}=A_{(4)\mathrm{I}}^2v_{\mathrm{I}}=\begin{cases} g_3=a_2+a_1+a_0 \\ g_2=a_3+a_0 \\ g_1=a_3 \\ g_0=a_3+a_2 \end{cases}. \quad (22)$$

According (19) and (21), the ${A_{(4)\mathrm{II}}}^2\times v_{\mathrm{II}}$ can be expressed as follows:

$$G_{(4)\mathrm{II}}=A_{(4)\mathrm{II}}^2v_{\mathrm{II}}=\begin{cases} g_3=a_2+a_0 \\ g_2=a_3+a_1 \\ g_1=a_1+a_0 \\ g_0=a_0 \end{cases}. \quad (23)$$

Table I summarizes the resources consumption and critical path required for implementation of quare over $GF(2^4)$, CCM over $GF(2^4)$, and the merged module $a^2\times v$. As shown in Table I, compared to separate implementations, the the merged module $a^2\times v$ saves up to 50% and 76.92% for the resources consumption in Case I and Case II, respectively, and shortens by 50% and 75% for the critical path in Case I and Case II, respectively.

IV. Optimization Based on MTP-CSE Algorithm

Common Subexpression Elimination (CSE) is an effective method to reduce the redundancy circuits in many applications. The idea of a CSE algorithm is to identify patterns (common subexpressions) that are present in expressions more than once and replace them with a single variable. It has been proven that how to select a pattern to eliminate is an NP-complete problem [20]. An MTP-CSE algorithm was proposed in [21]. Based on this algorithm, another more efficient MTP-CSE algorithm is proposed to reduce the gate count in S-box in this paper. The details of the proposed algorithm are described below.

A. MTP-CSE Algorithm

In the MTP-CSE algorithm, the patterns with most variables and highest occurring frequency are extracted to eliminate at each iteration. In most case, the number of the candidate patterns, which meet the selection criterion, is often more than one. Unlike the algorithm proposed in [21], which randomly selected a candidate pattern to eliminate, a greedy algorithm is used in our algorithm to check all possible patterns and find out the best set of patterns with the minimal area. The details of the new MTP-CSE algorithm are summarized in **Algorithm 1,** where the initial value of N is the number of input variables.

Algorithm 1: MTP-CSE algorithm

Input: the equations of arithmetic module;

Output: the gate count needed by optimized module; the eliminated patterns;

1. Compute the occurrence frequency of N-term patterns in the equations.
2. Establish a list for the N-term patterns with the highest frequency.
3. Select a pattern from the highest frequency list to be eliminated.
4. Replace all of the selected patterns in the equations with a new variable.
5. Append the selected pattern as a new equation to be further optimized.
6. Repeat Steps 2-5 for the next iteration until no recurring N-term patterns exist.
7. Let $N=N-1$, repeat Step 2-6 until $N<2$.
8. Compute the gate count needed by optimized module.
9. Select a different set of patterns from the highest frequency list to eliminate by repeating Steps 2-8, until all sets of patterns have been checked.
10. Find out the minimal gate count, and output the corresponding eliminated patterns.

An example in the following will illustrate the application of **Algorithm 1**. Considering a linear transform as follows:

$$\begin{bmatrix} y_3 \\ y_2 \\ y_1 \\ y_0 \end{bmatrix}=\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\times\begin{bmatrix} x_3 \\ x_2 \\ x_1 \\ x_0 \end{bmatrix}=\begin{bmatrix} x_3+x_2+x_1 \\ x_2+x_1 \\ x_1+x_0 \\ x_3+x_2+x_1+x_0 \end{bmatrix}, \quad (24)$$

The optimization process for (24) by the MTP-CSE algorithm can be expressed as in (25). As shown in (25), the new MTP-CSE algorithm takes two iterations to eliminate the patterns. Patterns "$x_3+x_2+x_1$" and "x_2+x_1" are identified at first iteration and second iteration, respectively, and they are replaced by new variables "z_0" and "z_1", respectively.

$$\begin{cases} y_3=x_3+x_2+x_1 \\ y_2=x_2+x_1 \\ y_1=x_1+x_0 \\ y_0=x_3+x_2+x_1+x_0 \end{cases} \xrightarrow{z_0=x_3+x_2+x_1} \begin{cases} y_3=z_0 \\ y_2=x_2+x_1 \\ y_1=x_1+x_0 \\ y_0=z_0+x_0 \\ z_0=x_3+x_2+x_1 \end{cases}$$

$$\xrightarrow{z_1=x_2+x_1} \begin{cases} y_3=z_0 \\ y_2=z_1 \\ y_1=x_1+x_0 \\ y_0=z_0+x_0 \\ z_0=x_3+z_1 \\ z_1=x_2+x_1 \end{cases}. \quad (25)$$

A straightforward implementation of the linear transformation requires 7 XOR gates. However, there are only 4 XOR gates required after optimization by the CSE algorithm. The reduction of XOR gates can be up to 42.85%.

TABLE II
TOTAL GATE COUNT AND CRITICAL PATH OF $\times\upsilon$, a^2, AND $a^2\times\upsilon$ OVER $GF(2^4)$

		Multiplication over $GF(2^4)$				Multiplicative Inversion over $GF(2^4)$				Matrix δ		Matrix $M\delta^{-1}$	
		Resources Consumption		Critical Path		Resources Consumption		Critical Path		Resources Consumption	Critical Path	Resources Consumption	Critical Path
		XOR	AND	XOR	AND	XOR	AND	XOR	AND	XOR	XOR	XOR	XOR
Case I	Original	32	16	4	1	21	25	4	2	18	3	25	3
	Optimized	20	9	4	1	13	8	4	2	12	3	15	3
Case II	Original	44	16	4	1	16	18	3	2	24	3	17	3
	Optimized	20	9	4	1	12	8	3	2	14	3	11	3

B. The Optimization of Multiplication over $GF(2^4)$

According to (10), the straightforward implementation of the multiplication over $GF(2^4)$ needs 32 XOR gates and 16 AND gates in Case I, and according to (11), it needs 44 XOR gates and 16 AND gates in Case II. The new MTP-CSE method can be employed to reduce both XOR gates and AND gates in the multiplication modules. The optimization process for (10) and (11) by the new MTP-CSE algorithm is divided into two steps: the *additive* common terms are firstly eliminated, then the *multiplicative* common terms are eliminated. After optimized by MTP-CSE algorithm, the multiplication over $GF(2^4)$ can be expressed as (26) and (27) in Case I and Case II, respectively.

The gate count used in optimized multiplication over $GF(2^4)$ is listed in Table II. In Case I, the optimized multiplication needs 20 XOR gates and 9 AND gates, compared with the straightforward implementations, it is up to 37.5% and 43.75%, respectively, reduced by MTP-CSE algorithm. In Case II, the optimized multiplication also needs 20 XOR gates and 9 AND gates, compared with the straightforward implementations, it is up to 54.54% and 43.75%, respectively, reduced by MTP-CSE algorithm. As shown in Table II, both in Case I and Case II, the critical path in optimized multiplication is same as the straightforward implemented ones.

In Case I, as shown in Fig. 1(b), there are three multiplications in the architecture of MI. According to (26), four XOR gates can be shared between two multiplications if they have a same input. Two of multiplications in Fig. 1(b) have a same input, and then they can share four XOR gates between them. The same goes for Case II, according to (27), four XOR gates can be shared between two multiplications too, if they have a same input. As shown in Fig. 1(c), there are also three multiplications in the architecture of MI in Case II, and there is a same input between any two multiplications, so total 12 XOR gates can be shared among the multiplications.

C. The Optimization of MI Module over $GF(2^4)$

Similar as the multiplication over $GF(2^4)$, the MI over $GF(2^4)$ can also be optimized by MTP-CSE algorithm. According to (16), the straightforward implementation of MI needs 21 XOR gates and 25 AND gates in Case I, and according to (17), it needs 16 XOR gates and 18 AND gates in Case II. The optimized MI in Case I and in Case II can be expressed as (28) and (29), respectively,

$$D_{(4)\mathrm{I}} = A_{(4)\mathrm{I}}^{-1} = \begin{cases} d_3 = \left(\left(\{a_3a_1\}_{n1}a_2 + a_2\right)_{m2} + a_3a_0\right)_{m1} + a_3 \\ d_2 = \left(a_3\{a_2a_0\}_{n0} + a_2a_1 + m_1\right)_{m0} \\ d_1 = \left(n_1a_0 + a_1\right)_{m3} + a_3 + m_2 + n_0 \\ d_0 = n_0a_1 + a_0 + m_0 + m_3 + n_1 \end{cases}, \quad (28)$$

$$D_{(4)\mathrm{II}} = A_{(4)\mathrm{II}}^{-1} = \begin{cases} d_3 = \left(\{a_3a_1\}_{e1} + a_2a_1\right)_{d0} + \{a_2a_0\}_{e0}a_1 \\ \qquad + a_1 + a_0 \\ d_2 = e_1a_0 + a_0 + d_0 + e_0 \\ d_1 = \left(a_3a_0 + e_1\right)_{d1} + a_3e_0 + a_3 + a_2 \\ d_0 = e_1a_2 + a_2 + d_1 + e_0 \end{cases}. \quad (29)$$

The gate count used in optimized MI is listed in Table II too. In Case I, the optimized MI needs 13 XOR gates and 8 AND gates, compared with the straightforward implementations, it is up to 38.10% and 68%, respectively, reduced by MTP-CSE algorithm. In Case II, the optimized MI needs 12 XOR gates and 8 AND gates, compared with the straightforward implementations, it is up to 25% and 55.56%, respectively, reduced by MTP-CSE algorithm. The critical path in optimized MI is same as the straightforward implemented ones both in Case I and Case II.

$$C_{(4)\mathrm{I}} = A_{(4)\mathrm{I}}B_{(4)\mathrm{I}} = \begin{cases} c_3 = \left[\left[a_0b_0\right]_{m7} + \left(a_2 + a_0\right)\left(b_2 + b_0\right)\right]_{m0} + \left[\left(a_1 + a_0\right)_{m3}\left(b_1 + b_0\right)_{m4}\right]_{m2} + \left(m_5 + m_3\right)\left(m_6 + m_4\right) \\ c_2 = \left[a_1b_1\right]_{m8} + \left(a_3 + a_1\right)\left(b_3 + b_1\right) + m_0 \\ c_1 = a_3b_3 + \left[\left(a_3 + a_2\right)_{m5}\left(b_3 + b_2\right)_{m6} + m_7\right]_{m1} + m_2 \\ c_0 = a_2b_2 + m_8 + m_1 \end{cases}. \quad (26)$$

$$C_{(4)\mathrm{II}} = A_{(4)\mathrm{II}}B_{(4)\mathrm{II}} = \begin{cases} c_3 = a_3b_3 + \left[\left(a_3 + a_2\right)_{m5}\left(b_3 + b_2\right)_{m6}\right]_{m2} + \left[\left[\left(a_2 + a_0\right)\left(b_2 + b_0\right)\right]_{m3} + \left(m_5 + m_7\right)\left(m_6 + m_8\right)\right]_{m0} \\ c_2 = a_2b_2 + \left[\left(a_3 + a_1\right)\left(b_3 + b_1\right) + m_3\right]_{m1} + m_2 \\ c_1 = a_1b_1 + \left[\left(a_1 + a_0\right)_{m7}\left(b_1 + b_0\right)_{m8}\right]_{m4} + m_0 \\ c_0 = a_0b_0 + m_1 + m_4 \end{cases}. \quad (27)$$

D. The Optimization of Isomorphic Mapping Matrices

For a fixed set of coefficients in irreducible polynomials in (3), there exist multiple isomorphic mapping between $GF(2^8)$ and $GF(((2^2)^2)^2)$ [14]. In Case I, for $\{\upsilon_I=(1100)_2, N_I=(10)_2\}$, then there is a pair of isomorphic mapping matrices as follows [14]:

$$\delta_I=\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&1&1&0&0&1\\0&1&1&1&0&0&1&0\\0&0&1&0&1&1&0&1\\0&0&0&0&1&1&1&0\\0&0&1&1&0&0&0&0\\0&1&1&1&1&0&1&1\\0&0&0&0&0&1&0&1\end{bmatrix};\ \delta_I^{-1}=\begin{bmatrix}1&0&0&0&0&0&0&0\\0&0&0&0&1&1&1&1\\0&1&1&0&1&0&0&1\\0&1&1&0&1&1&0&1\\0&1&1&1&1&0&0&0\\0&1&0&1&1&0&1&1\\0&0&1&0&1&0&1&1\\0&1&0&1&1&0&1&0\end{bmatrix}.$$

In Case II, for $\{\upsilon_{II}=(0001)_2, N_{II}=(10)_2\}$, then there is a pair of isomorphic mapping matrices as follows [13]:

$$\delta_{II}=\begin{bmatrix}1&1&1&0&0&1&1&1\\0&1&1&1&0&0&0&1\\0&1&1&0&0&0&1&1\\1&1&1&0&0&0&0&1\\1&0&0&1&1&0&1&1\\0&0&0&0&0&0&0&1\\0&1&1&0&0&0&0&1\\0&1&0&0&1&1&1&1\end{bmatrix};\ \delta_{II}^{-1}=\begin{bmatrix}0&0&0&1&0&0&1&0\\1&1&1&0&1&0&1&1\\1&1&1&0&1&1&0&1\\0&1&0&0&0&0&1&0\\0&1&1&1&1&1&1&0\\1&0&1&1&0&0&1&0\\0&0&1&0&0&0&1&0\\0&0&0&0&0&1&0&0\end{bmatrix}.$$

As mention above, the affine matrix M and the isomorphic mapping matrix δ^{-1} are usually merged into a single matrix $M\delta^{-1}$ to reduce hardware resource. The merged matrices $M\delta_I^{-1}$ and $M\delta_{II}^{-1}$ are shown as follows:

$$M\delta_I^{-1}=\begin{bmatrix}1&1&1&1&0&0&1&1\\0&0&1&0&1&0&0&0\\0&0&0&0&1&1&0&0\\0&0&1&1&1&1&1&1\\1&1&0&1&0&0&1&0\\1&0&1&0&0&1&0&1\\1&0&0&1&0&1&1&1\\1&1&0&1&0&0&0&1\end{bmatrix};\ M\delta_{II}^{-1}=\begin{bmatrix}0&0&1&0&1&0&0&0\\1&0&0&0&1&0&0&0\\0&1&0&0&0&0&0&1\\1&0&1&0&1&0&0&0\\1&1&1&1&1&0&0&0\\0&1&1&0&1&1&0&1\\0&0&1&1&0&0&1&0\\0&1&0&1&0&0&1&0\end{bmatrix}.$$

By using the new MTP-CSE algorithm, the optimization process of δ_I is given by

$$\delta_I X=\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&1&1&0&0&1\\0&1&1&1&0&0&1&0\\0&0&1&0&1&1&0&1\\0&0&0&0&1&1&1&0\\0&0&1&1&0&0&0&0\\0&1&1&1&1&0&1&1\\0&0&0&0&0&1&0&1\end{bmatrix}\begin{bmatrix}x_7\\x_6\\x_5\\x_4\\x_3\\x_2\\x_1\\x_0\end{bmatrix}$$

$$=\begin{bmatrix}x_7\\x_6+x_4+x_3+x_0\\x_6+x_5+x_4+x_1\\x_5+x_3+x_2+x_0\\x_3+x_2+x_1\\x_5+x_4\\x_6+x_5+x_4+x_3+x_1+x_0\\x_2+x_0\end{bmatrix}$$

$$=\begin{bmatrix}x_7\\x_6+x_4+(x_3+x_0)_{z1}\\(x_6+x_1+(x_5+x_4)_{z2})_{z0}\\x_5+x_2+z_1\\x_3+x_2+x_1\\z_2\\z_1+z_0\\x_2+x_0\end{bmatrix}. \quad (30)$$

Other matrices are same as δ_I. The optimized results are also listed Table II. The area reductions for δ_I and $M\delta_I^{-1}$ are up to 33.33% and 40%, respectively. And the area reduction for δ_{II} and $M\delta_{II}^{-1}$ are up to 41.67% and 35.29%, respectively. The critical paths in optimized matrices are same as the straightforward implemented ones both in Case I and Case II.

V. Implement Results and Analysis

After series of architectural and algorithmic optimizations, we have obtained the gate count and critical path for each sub-operation of MI over $GF(2^8)$. Combining with Fig. 1, we can obtain the total gate count and critical path for S-box. In Case I, the S-box needs 108 XOR gates and 35 AND gates with critical path of 20 XOR gates and 4 AND gates. In Case II, the S-box needs 96 XOR gates and 36 AND with critical path of 18 XOR gates and 4 AND gates. The results show that, between two architectures of S-box proposed in this paper, the S-box in Case II have smaller gate count and shorter critical path than the one in Case I.

Comparisons of the circuit complexities between our works and selected previous works are summarized in Table III. We use the key circuit parameters in SMIC 0.18μm CMOS technology to evaluate the works in Table III. In the SMIC 0.18μm CMOS technology, the area consumption for a XOR gate is 26.6112μm^2, and it is 13.3056μm^2 for an AND gate, while the standard delay for both XOR gate and AND gate is 1ns. Using these parameters, the area, delay, and Area-Delay-Product (ADP) are compared in Table III.

Polynomial basis was also employed in [5], [14], and [17]. Compared with these works, the S-box based on polynomial basis in this paper has the smallest area. Although the critical path in our polynomial basis based S-box is a litter longer than the S-box proposed in [14], the ADP in our work is smallest among these works due to much smaller area.

TABLE III
TOTAL RESOURCES CONSUMPTION, CRITICAL PATH, AND AREA-DELAY-PRODUCT FOR S-BOX IN OUR WORKS AND THE PRIOR WORKS

Works		[5]	[14]	[17]	Case I	[13]	[16] Case III	[12] Case III	[11] Case II	Case II
Basis Used		Polynomial	Polynomial	Polynomial	Polynomial	Normal	Normal	Normal	Normal	Normal
Resources Consumption	XOR (gates)	126	120	123	108	91	117	96	93	96
	AND (gates)	36	35	36	35	36	35	36	35	35
	Area (μm^2)	3832.01	3659.04	3752.18	3339.71	2900.62	3579.21	3033.68	2940.54	3033.68
Critical Path	XOR (gates)	25	19	23	20	22	20	20	20	18
	AND (gates)	4	4	4	4	4	4	4	3	4
	Dealy (ns)	29	23	27	24	26	24	24	23	22
Area-Delay-Product ($10^3\mu m^2$•ns)		111.12	84.16	101.31	80.15	75.41	82.32	72.81	67.63	66.74

Our normal basis based S-box is compared with the ones in [11] Case II, [12] Case III, [13], and [16] Case III, which are also based on normal basis. Among these works, our normal basis based S-box has the shortest critical path. Although the area in our works is a litter bigger than the S-box proposed in [11] Case II and [13], the ADP in our work is smallest among these works due to much shorter critical path. Beyond that, our normal basis based S-box has the shortest critical path and minimal ADP among all the works in the Table III.

VI. CONCLUSION

In this paper, the methods for compact AES S-box implementation have been discussed. Firstly, we derive out the expressions for sub-operations over $GF(2^4)$ in MI over $GF(2^8)$ by CFA technology. Then a novel MTP-CSE algorithm is proposed to reduce redundant resources in these sub-operations. Two case of S-box have been discussed based on a polynomial basis and a normal basis, respectively. As results, the S-box that based on the polynomial basis needs 108 XOR gates and 35 AND gates with critical path of 20 XOR gates and 4 AND gates, it has the minimal area and minimal ADP compared with previous works that based on the polynomial basis. The S-box based on the normal basis needs 96 XOR gates and 36 AND gates with critical path of 18 XOR gates and 4 AND gates, it has the shortest critical path and minimal ADP compared with previous works based on the normal basis. Furthermore, our normal basis based S-box can achieve the shortest critical path and the minimal ADP among the works that selected for comparison in this paper.

The optimization methods proposed in this paper can also be applied other circuits that involved finite field arithmetic. Our future works will focus on deriving the expression for MI over $GF(2^8)$ and proposing a more efficient CSE algorithm for it.

REFERENCES

[1] T. Sivakumar, and R. Venkatesan, "A Novel Approach for Image Encryption using Dynamic SCAN Pattern," *IAENG International Journal of Computer Science*, Vol. 41, No. 2, pp. 91-101, 2014.

[2] R. E. Boriga, A. C. Dascalescu, and A. V. Diaconu, "A New Fast Image Encryption Scheme Based on 2D Chaotic Maps," *IAENG International Journal of Computer Science*, Vol. 41, No. 4, pp. 249-258, 2014.

[3] M. A. B. Younes and A. Jantan, "Image encryption using block-based transformation algorithm", *IAENG International Journal of Computer Science*, Vol. 35, No. 1, pp. 15-23, 2008.

[4] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)", *FIPS Publication 197*, Nov. 2001.

[5] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization", *In Advances in Cryptology-ASIACRYPT 2001*, LNCS 2248, 2001, pp. 239–245.

[6] O. Song, and J. Kim, "Compact Design of the Advanced Encryption Standard Algorithm for IEEE 802.15.4 Devices," *Journal of Electrical Engineering & Technology*, Vol. 6, No. 3, pp. 418–422, 2011.

[7] J. A. R. Pacheco de Carvalho, H. Veiga, C. F. Ribeiro Pacheco, and A. D. Reis, "Extended Research on Performance of IEEE 802.11 a, b, g Laboratory WPA2 Point-to-Multipoint Links," *Engineering Letters*, Vol. 23, No. 1, pp. 8-13, 2015.

[8] L. Fu, X. Shen, L. Zhu, J. Wang, "A Low-Cost UHF RFID Tag Chip with AES Cryptography Engine," *Security and Communication Networks*, Vol. 7, No. 2, pp. 365–375, February 2014.

[9] S. Morioka, A. Satoh, "A 10-Gbps Full-AES Crypto Design with a Twisted BDD S-Box Architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 7. pp. 686-691, July 2004.

[10] Y. Chen, X. Zou, Z. Liu, Y. Han, and Z. Zheng, "Energy-efficient and security-optimized AES hardware design for ubiquitous computing," *Journal of Systems Engineering and Electronics*, Vol. 19, No. 4, pp. 652–658, 2008.

[11] X. Zhang, N. Wu, C. Zeng, "Compact S-box Hardware Implementation with an Efficient MVP-CSE Algorithm," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2015*, IMECS 2015, 18-20 March, 2015, Hong Kong, pp. 649–654.

[12] M. M. Wong, M. L. D. Wong, A. K. Nandi, I. Hijazin, "Construction of Optimum Composite Field Architecture for Compact High-Throughput AES S-Boxes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. vol. 20, no. 6, 2012, pp. 1151–1155.

[13] D. Canright, "A very compact Rijndael S-box," Technical report NPS-MA-04-001, Naval Postgraduate School, 2005

[14] X. Zhang and K. K. Parhi, "On the optimum constructions of composite field for the AES algorithm," *IEEE Transaction on Circuits and systems–II: Express Briefs,* vol. 53, no. 10, pp. 1153–1157, Oct. 2006.

[15] S.-F. Hsiao, M.-C. Chen, and C.-S. Tu, "Memory-Free Low-Cost Designs of Advanced Encryption Standard Using Common Subexpression Elimination for Subfunctions in Transformations," *IEEE Transactions on Circuits and Systems—I: Regular papers*, vol. 53, no. 3, March 2006.

[16] M. M. Wong, M. L. D. Wong, A.K. Nandi, I. Hijazin, "Composite field $GF(((2^2)^2)^2)$ Advanced Encryption Standard (AES) S-box with algebraic normal form representation in the subfield inversion," *Circuits, Devices & Systems, IET*. Vol. 5, Nov. 2011, pp. 471-476.

[17] N. Mentens, L. Batinan, B. Preneeland, and I. Verbauwhede, "A systematic evaluation of compact hardware implementations for the Rijndael S-box," *In Topics in Cryptology-CT-RSA 2005*, vol. 3376, 2005, pp. 323-333.

[18] A. Rudra, P. K. Dubey, C. S. Jutla, Vijay Kumar, Josyula R. Rao, and P. Rohatgi, "Efficient Rijndael encryption implementation with composite field arithmetic," *In Cryptographic Hardware and Embedded System-CHES2001*, LNCS 2162, 2001, pp.171-184.

[19] M. Mozaffari-Kermani, R. Reyhani-Masoleh, "A low-cost S-box for the advanced encryption standard using normal basis," *IEEE Int. Conf. Electro/Information Technology 2009*, pp. 52-55.

[20] N. Chen, and Z. Y. Yan, "Cyclotomic FFTs With Reduced Additive Complexities Based on a Novel Common Subexpression Elimination Algorithm," *IEEE Trans. Signal Processing*, Vol. 57, no. 3, pp. 1010-1020, Mar. 2009.

[21] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Durackova, "A new algorithm for elimination of common subexpressions". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 1, 1999, pp.58-68.