S-box Optimization for SM4 Algorithm

Yuan Zhu, Fang Zhou, Ning Wu, Yasir

Abstract—**This paper proposes a highly optimized S-box of SM4 algorithm for low-area and high-speed embedded application. A novel methodology is adopted for S-box implementation based on Composite Field Arithmetic (CFA) and mixed basis. The optimization result shows that the S-box based on mixed basis has shorter critical path than S-boxes based on normal basis and polynomial basis. Compared with previous works, the mixed basis based S-box proposed in this paper can achieve the shortest critical path. Besides, the operations over $GF((2^2)^2)$ and the constant matrix multiplications are optimized by Delay-Aware Common Sub-expression Elimination (DACSE) algorithm. ASIC implementation using static 180 nm @ 1.8 V yield an area reduction of 35.57% as compared to direct implementation.**

Index Terms—**SM4 algorithm, S-box, Composite Field Arithmetic (CFA), mixed basis**

I. INTRODUCTION

SM4 algorithm is a group symmetric cipher algorithm announced by Chinese government in January 2006 and it has been widely used in various fields of information security, such as wireless local area network (WLAN), Wireless LAN Authentication and Privacy Infrastructure (WAPI), storage device and the smart card system [1]-[2]. As the SM4 algorithm is mostly used in high-speed and resource-constrained applications, it is very necessary to design and implement short-delay and compact circuit of SM4. The implementation of S-box is the most expensive part in terms of the required hardware. Therefore, the short-delay and compact S-box is the key component of the SM4 algorithm.

The design and optimization of SM4 S-box have been studied in detail. The S-box implemented with LUT achieves high throughput but acquires large area. In [3], the N-dimensional hypercube method was introduced to construct S-box. Although the method reduced the area, it is difficult to implement in hardware because of complex derivation process. In [4]-[5], S-boxes based on polynomial basis and normal basis were introduced. Their optimizations for S-box focused on the hardware overhead at the cost of throughput. Therefore, when the SM4 algorithm is used for high-throughput and area-constrained devices, a short-delay and compact S-box is required for SM4 hardware implementation.

Manuscript received July 03, 2017; revised July 24, 2017. This work was supported in part by the National Natural Science Foundation of China (61376025), the Fundamental Research Funds for the Central Universities (NS2017023) and the Natural Science Foundation of Jiangsu Province (BK20160806).

Y. Zhu is with College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211100, China (e-mail: zhuyuan93@ qq.com).

F. Zhou is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211100, China(e-mail: zfnuaa@nuaa.edu.cn).

N. Wu is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211100, China.

Yasir is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211100, China.

This study focuses on the optimization of S-box for SM4 and the major contributions include:

- Coalescence design and implementation based on CFA technology [6] and mixed basis [7].
- The MI over $GF((2)^2)^2$ and constant matrix multiplications are optimized by DACSE algorithm [8].

II. BACKGROUND OF SM4 S-BOX ON CFA

The algebraic expression is shown as (1) and properties of the S-box for SM4 algorithm has been analyzed in [9].

$$S(x) = I\left(x \bullet A + C\right) \bullet A + C \tag{1}$$

where $\boldsymbol{I}$ is the MI over $GF(2^8)$. $\boldsymbol{A}$ is the affine matrix and $\boldsymbol{C}$ is row vector. $\boldsymbol{x}$ is the input of S-box and $\boldsymbol{S}(\mathbf{x})$ is the corresponding output. The S-box function involves a pre-affine transformation, MI over $GF(2^8)$ and a post-affine transformation. $\boldsymbol{A}$ and $\boldsymbol{C}$ are shown in (2).

$$A = \begin{bmatrix} 1&1&1&0&0&1&0&1 \\ 1&1&1&1&0&0&1&0 \\ 0&1&1&1&1&0&0&1 \\ 1&0&1&1&1&1&0&0 \\ 0&1&0&1&1&1&1&0 \\ 0&0&1&0&0&1&1&1 \\ 1&0&0&1&0&1&1&1 \\ 1&1&0&0&1&0&1&1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{2}$$

Since direct calculation of the MI over $GF(2^8)$ is a complicated and difficult task, we adopt the CFA to reduce the hardware complexity by mapping the MI over $GF(2^8)$ into composite field $GF(((2^2)^2)^2)$.

In CFA technology, an isomorphic mapping matrix is demanded to map the input vector from the 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 $GF(2^8)$. So the S-box based on CFA technique can be expressed as:

$$S(x) = \left(T^{-1}\left(I\left(x \bullet A + C\right) \bullet T\right)\right) \bullet A + C \tag{3}$$

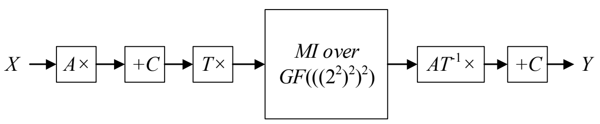

Fig. 1. Architecture of SM4 s-box using the CFA technique

where $\boldsymbol{T}$ is the isomorphic mapping matrix and $\boldsymbol{T}^{-1}$ is inverse matrix of $\boldsymbol{T}$. Generally, matrix $\boldsymbol{T}^{-1}$ and matrix $\boldsymbol{A}$ are merged into a single matrix to reduce the hardware resources. The

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

architecture of the S-box using the CFA technique is shown in Fig. 1.

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

In CFA technology, the MI over $GF(2)^8$ is built iteratively from GF(2) by using the following irreducible polynomials:

$$\begin{cases} GF\left(2^8\right) \to GF\left(\left(\left(2^2\right)^2\right)^2\right): \; g(\gamma)=\gamma^2+\gamma+v \quad (v=\alpha^2\beta) \\ GF\left(\left((2)^2\right)^2\right) \to GF\left((2)^2\right): f(\beta)=\beta^2+\beta+\alpha \\ GF\left((2)^2\right) \to GF(2): e(\alpha)=\alpha^2+\alpha+1 \end{cases} \tag{4}$$

The coefficients $\{v=(0100)_2,\ \lambda=(10)_2\}$ are used in this paper.

A. MI in Composite Domain

According to the first irreducible polynomial in (4), the MI over $GF(2^8)$ is decomposed into $GF(((2^2)^2)^2)$ as (5).

$$\begin{aligned} A^{-1} &= \left(AA^{16}\right)^{-1} A^{16} \\ &= \left\{a_l a_h \left(\gamma+\gamma^{16}\right)^2 + \left(a_l^2+a_h^2\right)\gamma\gamma^{16}\right\}^{-1}\left(a_h\gamma+a_l\gamma^{16}\right) \\ &= \left\{a_l a_h + \left(a_l+a_h\right)^2 v\right\}^{-1}\left\{a_l+\left(a_l+a_h\right)\gamma\right\} \\ &= b_l + b_h\gamma = B \end{aligned} \tag{5}$$

where A can be expressed as $A=a_l\gamma+a_h\gamma^{16}$, a_l, $a_h \in GF((2^2)^2)$. B can be represented as $B=b_l+b_h\gamma$, b_l, $b_h \in GF((2^2)^2)$. The architecture of MI over $GF(((2^2)^2)^2)$ is shown in Fig. 2.

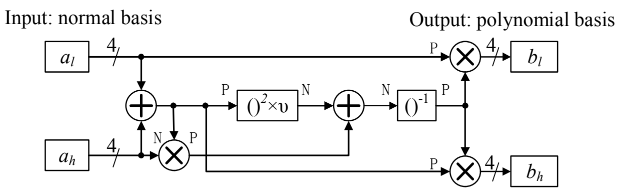

Fig. 2. MI over GF($((2^2)^2)^2$)

As shown in Fig. 2, the MI over $GF(((2^2)^2)^2)$ includes two additions, a MI, a square, a constant multiplication and three multiplications. All of the operations are over $GF((2^2)^2)$. The addition over $GF((2^2)^2)$ is defined as bitwise XOR gates. The square and constant multiplication over $GF((2^2)^2)$ can be deduced from multiplication over $GF((2^2)^2)$, and they are usually joint into a single block to reduce the number of gates.

There are two kind multiplications in this paper, mixed multiplication and general multiplication. In mixed multiplication operation, the input and output are represented by polynomial basis and normal basis, respectively. And for the general multiplication operation, the input and output are all denoted by polynomial basis.

Derivations of these operations are described in the following sections.

a) MI over $GF((2^2)^2)$

In $GF((2^2)^2)$ domain, the input of the MI is denoted by the normal basis $\{\beta, \beta^4\}$ and the output is represented by the polynomial basis $\{1,\beta\}$. Its inverse $B = A^{-1}$ is calculated in (6) and architecture of MI is shown in Fig. 3.

$$\begin{aligned} A^{-1} &= \left(AA^4\right)^{-1} A^4 \\ &= \left\{\left(a_l\beta+a_h\beta^4\right)\left(a_h\beta+a_l\beta^4\right)\right\}^{-1}\left(a_h\beta+a_l\beta^4\right) \\ &= \left\{a_l a_h + \left(a_l+a_h\right)^2\alpha\right\}^{-1}\left(\left(a_l+a_h\right)\beta+a_l\right) \\ &= b_l + b_h\beta = B \end{aligned} \tag{6}$$

Where A can be expressed as $A=a_l\beta+a_h\beta^4$, $a_l,a_h \in GF(2^2)$. B can be represented as $B=b_l+b_h\beta$, b_l, $b_h \in GF(2^2)$. By using the third irreducible polynomial in (4), the MI over $GF((2^2)^2)$ is further decomposed into $GF((2)^2)$ as (7).

$$B=(a_l\beta+a_h\beta^4)^{-1}=\begin{cases} b_3=a_0+a_1+a_2+a_3+a_1a_2+a_0a_3 \\ \qquad +a_0a_1a_2+a_0a_2a_3 \\ b_2=a_0+a_2+a_1a_2+a_0a_3+a_0a_1a_3 \\ \qquad +a_1a_2a_3 \\ b_1=a_0+a_1+a_1a_2+a_1a_3+a_0a_1a_2 \\ b_0=a_0+a_0a_2+a_1a_2+a_1a_3+a_0a_1a_3 \end{cases} \tag{7}$$

Where $a_l=a_0\alpha+a_1\alpha^2, a_h=a_2\alpha+a_3\alpha^2$, $a_0, a_1, a_2, a_3 \in GF(2)$.

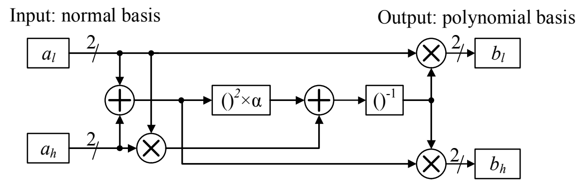

Fig. 3. MI over GF($(2^2)^2$)

The MI over $GF((2^2)^2)$ is optimized by the DACSE algorithm, and the optimized result includes 13 XOR gates and 8 AND gates, which needs 51 equivalent gates, with a reduction of 44.26% in terms of the total area occupancy compared with the direct implementation.

b) Constant Multiplied by Square over $GF((2^2)^2)$

The input and output of constant multiplied by square operation are denoted by the polynomial basis $\{1, \beta\}$ and normal basis $\{\beta, \beta^4\}$, respectively. Constant multiplied by square is calculated as (8) and its architecture is mentioned in Fig. 4.

$$\begin{aligned} A^2 v &= \alpha^2\beta(a_l+a_h\beta)^2 \\ &= \alpha^2\beta(a_l^2+a_h^2\beta^2) \\ &= a_l^2\alpha^2\beta + a_h^2\alpha^2(\alpha\beta^4+\beta) \\ &= \alpha^2\left(a_l^2+a_h^2\right)\beta + a_h^2\beta^4 \\ &= b_l\beta + b_h\beta^4 = B \end{aligned} \tag{8}$$

where A can be expressed as $A=a_l+a_h\beta$, $a_l,a_h \in GF(2^2)$. B can be represented as $B=b_l\beta+b_h\beta^4$, b_l, $b_h \in GF(2^2)$. The constant multiplied by square $p = a^2v$ is further decomposed into $GF((2)^2)$ as (9), using the third irreducible polynomial in (4).

$$p=(a_l+a_h\beta)^2\upsilon=\begin{cases} p_3=a_2 \\ p_2=a_3 \\ p_1=a_0+a_1+a_2+a_3 \\ p_0=a_0+a_2 \end{cases} \tag{9}$$

where $a_l=a_0\alpha+a_1\alpha^2, a_h=a_2\alpha+a_3\alpha^2$, $a_0, a_1, a_2, a_3 \in GF(2)$.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

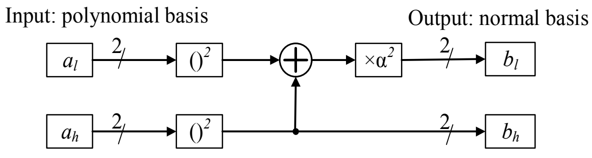

Fig. 4. Constant multiplied by square over $GF((2^2)^2)$

The optimized result consumes 3XOR gates with an area reduction of 25% compared with the direct implementation.

c) Mixed Multiplication over $GF((2^2)^2)$

Mixed multiplication operation needs a non–zero input represented with the polynomial basis $\{1, \beta\}$ and its output is expressed by normal basis $\{\beta, \beta^4\}$. Mixed multiplication $\hat{M}_4$ shown in Fig. 5 is calculated as (10).

$$\begin{aligned} \hat{M}_4 &= AB \\ &= (a_l + a_h\beta)(b_l + b_h\beta) \\ &= \{(a_l + a_h)(b_l + b_h) + a_h b_h \alpha\}\beta + (a_l b_l + a_h b_h \alpha)\beta^4 \\ &= c_l\beta + c_h\beta^4 = C. \end{aligned} \tag{10}$$

where A can be expressed as $A=a_l+a_h\beta$, $a_l,a_h \in GF(2^2)$. B can be represented in the same way. C can be represented as $C=c_l\beta+c_h\beta^4$, c_l, $c_h \in GF(2^2)$. By using the third irreducible polynomial in (4), mixed multiplication $C=AB$ is further decomposed into $GF((2)^2)$ as (11).

$$C = AB = \begin{cases} c_3 = (a_1b_0 + a_0b_1) + (a_2b_3 + a_3b_2) + a_3b_3 + a_0b_0 \\ c_2 = (a_1b_0 + a_0b_1) + a_3b_3 + a_2b_2 + a_1b_1 \\ c_1 = (a_1b_0 + a_0b_1) + (a_2b_0 + a_0b_2) + (a_3b_0 + a_0b_3) \\ \qquad + (a_2b_1 + a_1b_2) + a_3b_3 + a_2b_2 + a_0b_0 \\ c_0 = (a_1b_0 + a_0b_1) + (a_3b_0 + a_0b_3) + (a_2b_1 + a_1b_2) \\ \qquad + (a_3b_1 + a_1b_3) + (a_2b_3 + a_3b_2) + a_2b_2 + a_1b_1 \end{cases} \tag{11}$$

where $a_l=a_0\alpha+a_1\alpha^2, a_h=a_2\alpha+a_3\alpha^2$, $b_l=b_0\alpha+b_1\alpha^2$, $b_h=b_2\alpha+b_3\alpha^2$, $a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3 \in GF(2)$.

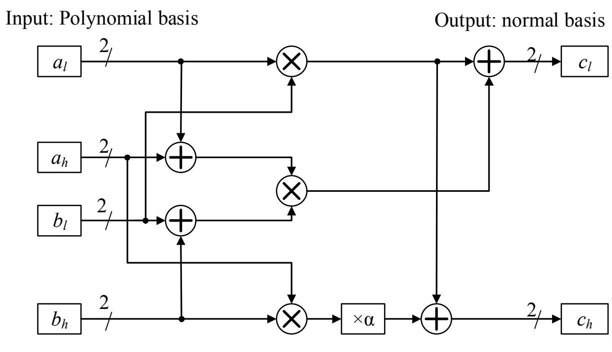

Fig. 5. Mixed multiplication over $GF((2^2)^2)$

DACSE algorithm is adopted to optimize the mixed multiplication over $GF((2^2)^2)$, and the optimized circuit requires 81 equivalent gates. Compared with the direct implementation, which needs 21 XOR gates and 27 AND gates, it makes 61.5(43.16%) gates reduction in total area cost.

d) *General Multiplication over* $GF((2^2)^2)$

The input and output of general multiplication are all denoted by the polynomial basis $\{1, \beta\}$. General multiplication M_4 which is constructed as Fig. 6 is calculated in (12).

$$\begin{aligned} M_4 &= AB \\ &= (a_l + a_h\beta)(b_l + b_h\beta) \\ &= (a_l b_l + a_h b_h \alpha) + \{a_h b_h + a_l b_h + a_h b_l\}\beta \\ &= (a_l b_l + a_h b_h \alpha) + \{(a_l + a_h)(b_l + b_h) + a_l b_l\}\beta \\ &= c_l + c_h\beta = C. \end{aligned} \tag{12}$$

where A can be expressed as $A=a_l+a_h\beta$, $a_l,a_h \in GF(2^2)$. B and C can be represented in the same way. By using the third irreducible polynomial in (4), general multiplication $C=AB$ is further decomposed into $GF((2)^2)$ as (13).

$$C = AB = \begin{cases} c_3 = (a_2b_0 + a_0b_2) + (a_3b_0 + a_0b_3) + (a_2b_1 + a_1b_2) \\ \qquad + (a_2b_3 + a_3b_2) + a_2b_2 \\ c_2 = (a_3b_0 + a_0b_3) + (a_2b_1 + a_1b_2) + (a_3b_1 + a_1b_3) \\ \qquad + (a_2b_3 + a_3b_2) + a_3b_3 \\ c_1 = (a_1b_0 + a_0b_1) + (a_2b_3 + a_3b_2) + a_3b_3 + a_0b_0 \\ c_0 = (a_1b_0 + a_0b_1) + a_3b_3 + a_2b_2 + a_1b_1 \end{cases} \tag{13}$$

where $a_l=a_0\alpha+a_1\alpha^2, a_h=a_2\alpha+a_3\alpha^2$, $b_l=b_0\alpha+b_1\alpha^2$, $b_h=b_2\alpha+b_3\alpha^2$, $a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3 \in GF(2)$.

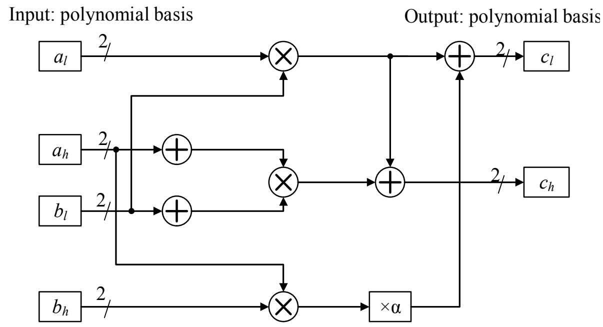

Fig. 6. General multiplication over $GF((2^2)^2)$

The general multiplication optimized by DACSE occupies 75 equivalent gates, with a reduction of 33.33% as compared with the direct implementation.

B. Calculation for Mapping Matrix

The calculation procedure of the mapping matrix is shown in Fig. 7 and specific steps are described as follows:

1) Determine the irreducible polynomial coefficients υ and find the minimum root w of $P_{28}(w)=0$.
2) Calculate other seven roots from the smallest root $\beta_0=w$ in accordance with (14).
3) Generate a mapping matrix T_i according to (15).

$$\beta_i = \beta_0^{2^i}, (\mathrm{i} = 0 \sim 7) \tag{14}$$

$$T_i = \left[1, \beta_i, \beta_i^2, \beta_i^3, \beta_i^4, \beta_i^5, \beta_i^6, \beta_i^7\right] \tag{15}$$

The mapping matrix $\boldsymbol{T}$ can be calculated only when operations over $GF(((2^2)^2)^2)$, $GF((2^2)^2)$ and $GF(2^2)$ are represented by normal basis, polynomial basis and normal basis, respectively. Since the output of MI over $GF(((2^2)^2)^2)$ based on mixed basis is expressed by polynomial basis, it is necessary to convert polynomial basis into normal basis. Therefore, we need to multiply the matrix $\boldsymbol{T'}$ before performing the post-affine operation. So (3) is converted to (16).

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

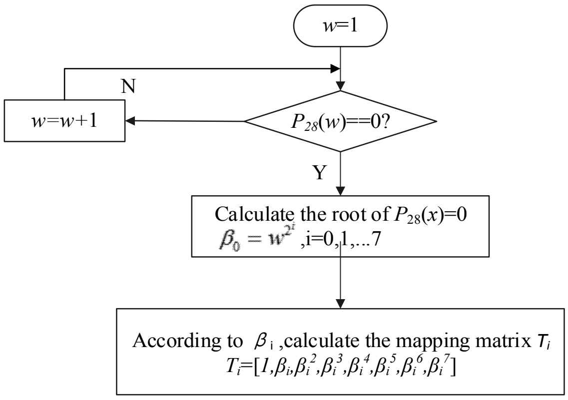

Fig. 7. Algorithm flowchart of mapping matrix

$$S(x) = I\left(\left(x \bullet A + C\right) \bullet T\right) \bullet \left(AT^{-1}T'\right) + C \tag{16}$$

One group of constant matrices is shown as (17) and (18).

$$T = \begin{bmatrix} 0&1&0&0&0&1&0&0 \\ 1&1&1&0&0&1&0&0 \\ 0&0&0&1&1&1&0&1 \\ 0&0&0&0&0&0&0&1 \\ 1&1&1&0&1&0&0&0 \\ 0&0&0&1&0&1&1&0 \\ 0&1&1&0&1&1&1&1 \\ 0&1&0&1&0&0&0&1 \end{bmatrix}, \quad T' = \begin{bmatrix} 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&0&0&1&0&0&0 \\ 0&1&0&0&0&1&0&0 \\ 0&0&1&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \end{bmatrix} \tag{17}$$

$$\overline{T} = \begin{bmatrix} 0&1&1&0&0&1&1&0 \\ 0&1&1&0&1&0&0&1 \\ 1&0&1&0&0&1&1&0 \\ 0&1&1&1&1&0&0&0 \\ 1&0&1&0&0&0&0&1 \\ 1&1&1&0&1&0&0&1 \\ 1&0&0&1&0&1&0&1 \\ 0&0&0&1&0&0&0&0 \end{bmatrix} \quad A\overline{T}T' = \begin{bmatrix} 0&0&1&0&1&1&0&1 \\ 1&0&0&0&1&0&0&1 \\ 1&0&0&0&1&0&1&1 \\ 0&0&1&1&1&0&1&1 \\ 0&1&1&0&1&1&1&0 \\ 0&0&1&1&1&1&1&1 \\ 0&1&0&0&1&1&1&0 \\ 1&1&0&0&0&1&1&0 \end{bmatrix} \tag{18}$$

Optimized by DACSE algorithm, the required hardware resources for optimized matrices are 58 XOR gates, with an area reduction of 34.09% compared with the direct implementation.

IV. Implement Results and Analysis

In this paper, the S-box for SM4 algorithm is designed by CFA technology and mixed basis. The critical path of S-box based on different basis are shown in Table I.

In Table I, the implementations of SM4 S-boxes based on polynomial basis and normal basis are realized according to [4] and [5], respectively. From the delay shown in Table I, the optimized S-boxes has been reduced by 22.5% and 17.05% in the terms of total delay, as compared to S-box based on polynomial basis and normal basis, respectively.

The critical path of MIs over $GF(((2^2)^2)^2)$ have a decisive effect on the critical path of whole S-box, because the critical path of mapping matrices of SM4 S-box are 3 XOR gates with no further optimization. Therefore, the comparison of critical path of MI can explain the performance of the whole S-box. Because of the MIs of AES S-box and SM4 S-box over $GF(((2^2)^2)^2)$ have the same structure, their critical path can be compared directly. Comparisons of the critical path between our works and selected previous works are summarized in Table II. Compared with these works, the S-box based on mixed basis in this paper has the shortest critical path.

TABLE I
THE CRITICAL PATH OF S-BOX IN COMPOSITE DOMAIN

Basis Used	Construction	Critical Path	
		AND	XOR
Polynomial	$T\times$	—	3
	$AT^{-1}\times$	—	3
	MI over $GF(((2^2)^2)^2)$	4	17
	S-box	4	28
Normal	$T\times$	—	3
	$AT^{-1}\times$	—	3
	MI over $GF(((2^2)^2)^2)$	4	15
	S-box	4	26
ours	$T\times$	—	3
	$AT^{-1}T'\times$	—	3
	MI over $GF(((2^2)^2)^2)$	4	10
	S-box	4	21

TABLE II
THE CRITICAL PATH OF MI OVER $GF(((2^2)^2)^2)$

Works	Basis Used	Critical Path	
		AND	XOR
[10]	Polynomial	4	13
[11]	Polynomial	4	17
[12] case I	Polynomial	4	14
[13]	Polynomial	4	14
[12] case II	Normal	4	12
[14]	Normal	4	14
[15]	Normal	4	14
[16]	Normal	3	14
[7]	Mixed	4	14
ours	Mixed	4	10

In order to reduce the area cost of S-box, the designed S-box is optimized by DACSE algorithm. The type and quantity of logic gates as well as the total number of transistors in the direct implementation and optimization by DACSE are listed in Table III, respectively.

TABLE III
THE AREA COST AND CRITICAL PATH BY EACH PART OF THE CFA-BASED S-BOX

Module	Direct				Optimized by DACSE			
	AND	XOR	Gates	Critical Path	AND	XOR	Gates(Reduction)	Critical Path
A	——	32	96	3 T_{XOR}	——	21	63(34.38%)	3 T_{XOR}
T	——	19	57	3 T_{XOR}	——	14	43(26.32%)	3 T_{XOR}
$AT^{-1}T'$	——	27	81	3 T_{XOR}	——	13	39(51.85%)	3 T_{XOR}
C	——	10	30	2 T_{XOR}	——	10	30	2 T_{XOR}
MI over $GF(((2^2)^2)^2)$	114	108	495	10 T_{XOR}+4 T_{AND}	50	80	315(36.36%)	10 T_{XOR}+4 T_{AND}
s-box	114	196	759	21 T_{XOR}+4 T_{AND}	50	138	489(35.57%)	21 T_{XOR}+4 T_{AND}

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

In Table III, T_{XOR} and T_{AND} denote the delays of XOR gates and AND gates, respectively. As shown in Table III, the optimized circuit of MI over $GF(((2^2)^2)^2)$ is reduced by 36.36%. The area reduction for S-box is up to 35.57%.

V. CONCLUSIONS

This paper proposed a highly optimized S-box based on CFA and mixed basis for SM4 algorithm. Compared with S-boxes based on polynomial basis and normal basis, the proposed S-box has shortest critical path of 21 XOR gates and 4 AND gates. The MI over $GF(((2^2)^2)^2)$ has the shortest delay compared with previous works that based on the normal basis, polynomial basis or mixed basis. Besides, the designed S-box was optimized by DACSE algorithm to reduce the area cost. As compared to the direct implementation, the area reduction of MI over $GF(((2^2)^2)^2)$ and the optimized S-box are up to 36.36% and 35.57% using 180 nm 1.8 V COMS technology.

REFERENCES

[1] W. Diffie and G. Ledin. "SMS4 Encryption Algorithm for Wireless Networks," *Iacr Cryptology Eprint Archive*, 2008.

[2] Office of State Commercial Cipher Administration. (2006). SMS4 cipher for WLAN products. [Online]. Available: http://www.oscca.gov.cn/UpFile/200621016423197990.pdf

[3] Y. Wang, L. I. Shu-Guo and T. University. "The Application of Hypercube Method in S Box of SM4," *Microelectronics & Compute*r, vol. 31, no. 7, pp. 10–17, Jul. 2014.

[4] H. Liang, L. Wu and X. Zhang. "Design and implementation of SM4 block cipher based on composite field," *Microelectronics & Computer*. vol. 32, no. 5, pp. 16–20, May. 2015.

[5] Y. H. Xu, X. F. Bai and L. Guo. "A new algorithm of S-box for hardware implementation of SMS4," *Journal of University of Science & Technology of China*. vol. 39, no. 11, pp. 1164–1170, Nov. 2009.

[6] Y. Chen, X. Zou, Z. Liu, Y. Han and Z. Zheng. "Energy-efficient and security-optimized AES hardware design for ubiquitous computing," *Journal of System Engineering and Electronic*, vol. 17, no. 4, pp. 652–658, 2008.

[7] Y. Nogami, K. Nekado, T. Toyota, N. Hongo and Y. Morikaw, "Mixed bases for efficient inversion in $F(((2)^2)^2)^2$ and conversion matrices of SubBytes of AES," *IEICE Transactions on Fundamentals of Electronics Communications & Computer Sciences,* vol. 97, no. 6, pp. 1318–1327, 2010.

[8] X. Zhang, X. Wu, F. Zhou and X. Chen, "An optimized delay-aware common subexpression elimination algorithm for hardware implementation of binary-field linear transform," *IEICE Electronics Express*, vol. 11, no. 22, pp. 20140934–20140934, 2014.

[9] F Liu, W. Ji, L. Hu, J. Ding, S. Lv and A. Pyshkin, et al, " Analysis of the SMS4 Block Cipher," *Information Security and Privacy, Australasian Conference*, vol. 4586, pp. 158–170, Jul.2007.

[10] X. Zhang and K. K. Parhi, "On the Optimum Constructions of Composite Field for the AES Algorithm," *IEEE Transactions on Circuits & Systems II Express Briefs*, vol. 53, no. 10, pp. 1153-1157,2006.

[11] N. Mentens, L. Batinan, B. Preneeland, and I. Verbauwhede, "A systematic evaluation of compact hardware implementations for the Rijndael S-box," *Cryptographers' Track at the RSA Conference. Springer Berlin Heidelberg*, vol. 3376, pp. 323-333, 2005.

[12] X. Zhang, N. Wu, G. Yan and L. Dong, "Hardware implementation of compact AES S-box," *Iaeng International Journal of Computer Science,* vol. 42, no. 2,pp. 125-131, 2015.

[13] D. Canright, "A very compact Rijndael S-box," Technical Report Collection, pp.4-5, 2005.

[14] M. M. Wong, M. L. D. Wong, A.K. Nandi and 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 let, vol. 5, no. 6, pp. 471-476, 2011.

[15] M. M. Wong, M. L. D. Wong, A. K. Nandi and 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, pp. 1151–1155, 2012.

[16] X. Zhang, N. Wu, C. Zeng, "Compact S-box Hardware Implementation with an Efficient MVP-CSE Algorithm," Lecture Notes in Engineering & Computer Science, vol. 2216, no. 1, pp. 649–654, 2015.