Application of the Composite Field in the Design of an Improved AES S-box Based on Inversion

Zhao Wang, Xiao Zhang, Sitao Wang, Zhisong Hao and Zhiming Zheng
School of Mathematics and Systems Science & LMIB
Beijing University of Aeronautics and Astronautics
Beijing, China
Email: wangzhao@smss.buaa.edu.cn, xiao.zh@buaa.edu.cn, wang_sitao@smss.buaa.edu.cn,
haozhisong2004@sina.com, zzheng@pku.edu.cn.

***Abstract*—The hardware implementation of the Substitution-Box (S-box) of the Advanced Encryption Standard (AES) always employs composite field $GF((2^n)^2)$ to obtain better efficiency. In this paper, an improved class of S-boxes by direct inversion in composite field is presented, and the choice of the subfield leading to the most efficient implementation is discussed. Eliminating the field isomorphic transformations, such a composite field is easier to fix and the resulting hardware implementation is more efficient than that of AES S-box. Some common cryptographic characteristics for the composite field based S-boxes are examined, and it turns out that direct inversion in composite field does not weaken the cryptographic characteristics. In addition, a demonstration for the immunity against the potential algebraic attack on AES with the replacement of our S-box is given, and it is proven that the revised AES is even more secure than the original AES against the algebraic attack. As a result of this work, it could be predicted that the isomorphism implies equal immunity from certain cryptanalysis. Our S-box is suitable for the area-limited hardware production.**

***Keywords*–AES; Composite field; S-box; Hardware implementation.**

I. Introduction

The Substitution-Box (S-box) is a basic component of symmetric key algorithms, and should always be carefully chosen to create strong confusion and to resist certain kinds of cryptanalysis. The multiplicative inversion mapping over Galois Field are frequently employed due to their ideal cryptographic characteristics [1]. Most of the recent S-boxes in block ciphers, such as the Advanced Encryption Standard (AES) [2], Camellia (NESSIE and CRYPTREC winner) [3], CLEFIA (developed by SONY) [4] and SMS4 (used in the Chinese National Standard for Wireless LAN WAPI) [5] are created based on the inversion on $GF(2^8)$. Currently, the GF-inversion has become one of the most popular algebraic tools in block ciphers, and its hardware implementation, especially targeted for AES is still a worldwide challenge.

Among so many state-of-art designs to implement GF-inversion, one general idea is to employ the composite field representation [6]. The fields $GF(2^8)$, $GF((2^4)^2)$ and $GF(((2^2)^2)^2)$ are linear isomorphic to each other, so that the isomorphisms can be achieved by simple matrix multiplications, which means that finding the inverse in $GF(2^8)$ can be changed into calculating the low-cost addition, multiplication, square and inversion in $GF(2^4)$ [7] or $GF((2^2)^2)$ [8].

Mentens et al. [9] used $GF(((2^2)^2)^2)$, examined all possible choices for irreducible polynomials generating the extension field and all the transformation matrices mapping to the corresponding $GF(((2^2)^2)^2)$ representation, and pointed out the area of [8] was still 5% from the minimum. Later, Canright [10] considered not only polynomial bases but also normal bases, with 432 cases including all in [9], and get the most compact S-box up to date. However, the critical path delay, substructure sharing and use of NOR and NOT gates were all ignored, and then Zhang and Parhi [11] improved it and got a seemingly better result. Nikova and Rijmen [12] decomposed $GF(2^8)$ to $GF((2^4)^2)$ using normal bases and the result could compete with that in [10]. For the other recent architectures, Nogamni et al. [13] suggest mixing normal and polynomial bases for the reduction of the critical path delay of S-box.

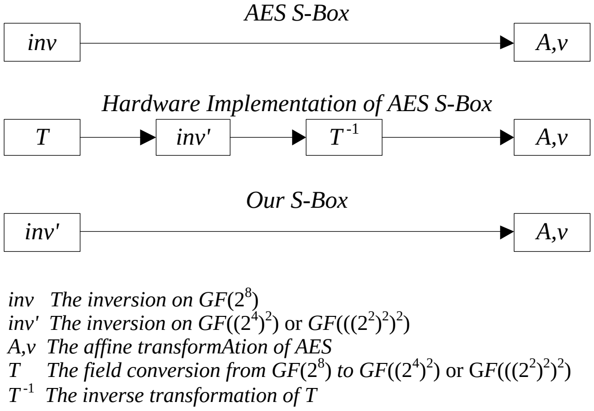
AES S-Box
inv
A,v
Hardware Implementation of AES S-Box
T
inv'
T^{-1}
A,v
Our S-Box
inv'
A,v
inv The inversion on $GF(2^8)$
inv' The inversion on $GF((2^4)^2)$ or $GF(((2^2)^2)^2)$
A,v The affine transformAtion of AES
T The field conversion from $GF(2^8)$ to $GF((2^4)^2)$ or $GF(((2^2)^2)^2)$
T^{-1} The inverse transformation of T

Figure 1. Structures of the different S-boxes in this paper.

In this paper, we try to do the inversion directly in composite field; see Figure 1. Then, the effect caused by transformation matrixes can be ignored, and the choice of irreducible polynomial for field extension would be easier.

The outline of this paper is as follows. In Section II, the applications of composite field in S-box implementation is introduced. In Section III, our new S-box based on inversion in composite field is described. The complete cryptographical analysis of this new S-box and some potential algebraic attacks are given in Section IV and Section V, followed by the conclusion.

II. Preliminaries

In this section, we show the applications of the composite field for the hardware implementation of AES S-box.

The AES S-box involves an inversion mapping in $GF(2^8)$ followed by a $GF(2)$-affine transformation, here $GF(2^8) = GF(2)/(O(z))$, $O(z) = z^8 + z^4 + z^3 + z + 1$. Denote the AES S-box by $S_1(x)$,

$$S_1(x) = \boldsymbol{A} \cdot x^{-1} + \boldsymbol{v} = \boldsymbol{A} \cdot inv(x) + \boldsymbol{v}, \qquad (1)$$

where $inv(x)$ denotes $GF(2^8)$ inversion of x. Denote the binary representation of any $x \in GF(2^8)$ by $(x_7, x_6, \cdots, x_0)$ with x_7 the most significant bit. And

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \boldsymbol{v} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}. \qquad (2)$$

The basic idea of employing composite field comes from the field isomorphism $GF(2^n) \cong GF((2^{n/2})^2) \cong GF(((2^{n/4})^2)^2)$. However, all the basic arithmetics in $GF((2^{n/2})^2)$ are actually the ones in $GF(2^{n/2})$. Each element $\Delta \in GF((2^m)^2) = GF(2^m)/(x^2 + \alpha x + \beta)$ where $\alpha, \beta \in GF(2^m)$, can be expressed as $\Delta = \delta_1 x + \delta_0$, with $\delta_1, \delta_0 \in GF(2^m)$. The multiplicative inversion of Δ can be obtained via

$$\Delta^{-1} = (\delta_1 \Gamma)x + (\delta_0 + \delta_1 \alpha)\Gamma, \Gamma = (\delta_1^2 \beta + \delta_1 \delta_0 \alpha + \delta_0^2)^{-1}. \quad (3)$$

$GF(2^8)$ can be mapped to either $GF((2^4)^2)$ or $GF(((2^2)^2)^2)$. The specific operation comprises of the following 3 steps:

(I) Field linear isomorphic transformation T maps each $GF(2^8)$ element into the composite field $GF((2^4)^2)$ (or $GF(((2^2)^2)^2)$).

(II) Inversion in composite field.

(III) Linear transformation T^{-1}.

The operations using $GF((2^4)^2)$ representation can be expressed as

$$S_1(x) = \boldsymbol{A} \cdot \boldsymbol{T}_1^{-1} \cdot inv'(\boldsymbol{T}_1 \cdot x) + \boldsymbol{v}, \qquad (4)$$

where $inv'(x)$ denotes $GF((2^4)^2)$ inversion of x. In [8], there is another architecture by mapping $GF(2^8)$ to $GF(((2^2)^2)^2)$, and the operations can be expressed as

$$S_1(x) = \boldsymbol{A} \cdot \boldsymbol{T}_2^{-1} \cdot inv''(\boldsymbol{T}_2 \cdot x) + \boldsymbol{v}, \qquad (5)$$

where $inv''(x)$ denotes $GF(((2^2)^2)^2)$ inversion of x.

III. Improved S-boxes by inversion in Composite Field

A. *The Specification of the Improved S-boxes*

According to the designers of AES [2], the choice of $GF(2^8)$ inversion only comprises cryptographical reasons but barely covers implementation efficiency. It provides very ideal input-output correlation amplitude and difference propagation probability, and the following $GF(2)$-affine transformation (A, v) complicates the algebraic expression of $S_1(x)$. However, one may find it difficult to build the optimum hardware architecture for $S_1(x)$ on composite field. We believe that three reasons might lead to such a situation:

(I) *Computational Complexity*. As revealed by the Extended Euclid's Algorithm [14], GF-inversion is essentiality more complex than any other basic GF arithmetic such as multiplication.

(II) *Overfull Factors*. Introducing the composite field does reduce the area cost, however, additional factors arise, such as the coefficients of the irreducible polynomial generating the composite field. The following factors must be considered:

a) Subfield multiplication of two multiplicands;
b) Subfield squaring;
c) Subfield constant multiplication;
d) Matrixes of $\boldsymbol{T}$ and $\boldsymbol{A} \cdot \boldsymbol{T}^{-1}$.

(III) *How to Define the "optimum"?* The criteria to build the optimum architecture of the previous contributions [7]–[13] are unambiguous, which can not be achieved simultaneously. For example, the throughput usually contradicts the area since compact construction always causes more critical path delays, and certain criterion is hard to be judged or quantified. As shown by Mentens et al. [9], further area reduction can be achieved by substructure sharing or introducing NOR gates or NOT gates, so the matrix for $\boldsymbol{T}$ with the least number of "1" might not be the best choice.

Recognizing the comparability among (1), (4) and (5), we could try to overlook the field isomorphism T_i and T_i^{-1} and directly carry out the multiplicative inversion in the isomorphic composite field. So, two new S-boxes are obtained in (6) and (7), and only the irreducible polynomials are to be fixed.

$$S_2(x) = \boldsymbol{A} \cdot inv'(x) + \boldsymbol{v}, x \in GF((2^4)^2)$$
$$\begin{cases} GF((2^4)^2) & = \quad GF(2^4)/(M(x)) \\ M(x) & = \quad x^2 + m_0 x + m \\ GF(2^4) & = \quad GF(2)/(N(y)) \\ N(y) & = \quad y^4 + n_2 y^3 + n_1 y^2 + n_0 y + n \end{cases} \qquad (6)$$

$$S_3(x) = \boldsymbol{A} \cdot inv''(x) + \boldsymbol{v}, x \in GF(((2^2)^2)^2)$$
$$\begin{cases} GF(((2^2)^2)^2) & = \quad GF((2^2)^2)/(P(x)) \\ P(x) & = \quad x^2 + p_0 x + p \\ GF((2^2)^2) & = \quad GF(2^2)/(Q(y)) \\ Q(y) & = \quad y^2 + q_0 y + q \\ GF(2^2) & = \quad GF(2)/(R(z)) \\ R(z) & = \quad z^2 + r_0 z + r \end{cases} \qquad (7)$$

First of all, fix $R(z) = z^2 + z + 1$ since it is the only irreducible polynomial over $GF(2)$ with degree 2. Then, set $m_0 = 1 \in GF(2^4)$ in (8) and $p_0 = 1 \in GF((2^2)^2)$, $q_0 = 1 \in GF(2^2)$ in (9) since they would reduce more multiplications than m and p, q, similar to the AES settings in Section II.

For $S_2(x)$, there are only three choices for $N(y)$, and for $M(x)$, only m is to be decided. When $N(y)$ is fixed, there are only limited candidate values for m to make $M(x)$ irreducible. The best value for m can be found through comparing the gate counts of multiplying m. For $S_3(x)$, since $R(z)$, p_0 and q_0 are fixed, we find there are only eight choices of (p,q) to make $P(x)$ and $Q(y)$ irreducible, so the best (p,q) can also be fixed through simple comparison. Consequently, all the settings can

TABLE I. NUMBER OF XOR GATE FOR T AND $A \cdot T_1^{-1}$

Design	**Field**	**Original**	**After Optimizations**
[7]	$GF((2^4)^2)$	43	Not available
[8]	$GF(((2^2)^2)^2)$	45	Obtained by greedy algorithm
[9]	$GF(((2^2)^2)^2)$	38	No optimization
[11]	$GF(((2^2)^2)^2)$	36	28 (Gate) + 6 (Critical Path)
[11]	$GF(((2^2)^2)^2)$	38	27 (Gate) + 6 (Critical Path)
S_2	$GF((2^4)^2)$	**32**	**18** (Gate) + **4** (Critical Path)
S_3	$GF(((2^2)^2)^2)$	**32**	**18** (Gate) + **4** (Critical Path)

be fixed:

$$S_2(x) = \boldsymbol{A} \cdot inv'(x) + \boldsymbol{v}, x \in GF((2^4)^2)$$

$$\begin{cases} GF((2^4)^2) & = & GF(2^4)/(M(x)) \\ M(x) & = & x^2 + x + \{1001\}_2 \\ GF(2^4) & = & GF(2)/(N(y)) \\ N(y) & = & y^4 + y + 1 \end{cases} \tag{8}$$

$$S_3(x) = \boldsymbol{A} \cdot inv''(x) + \boldsymbol{v}, x \in GF(((2^2)^2)^2)$$

$$\begin{cases} GF(((2^2)^2)^2) & = & GF((2^2)^2)/(P(x)) \\ P(x) & = & x^2 + x + \{1100\}_2 \\ GF((2^2)^2) & = & GF(2^2)/(Q(y)) \\ Q(y) & = & y^2 + y + \{10\}_2 \\ GF(2^2) & = & GF(2)/(R(z)) \\ R(z) & = & z^2 + z + 1 \end{cases} \tag{9}$$

B. *Performance of Hardware Implementation*

Defined in composite filed, the inversion function in both S_2 and S_3 could be implemented directly by (3) without field isomorphism.

Observing the differences between (4) and (8), or (5) and (9), our improvement on efficiency is manifest. Most of the previous architectures for S_1 based on composite field need to do at least two matrix multiplications, $\boldsymbol{T}$ and $\boldsymbol{A} \cdot \boldsymbol{T}_1^{-1}$, while our S-boxes only need to multiply the matrix $\boldsymbol{A}$. The number of XOR gate for the two matrix multiplications of the previous architectures are listed in Table I. Note that [10] [12] [13] used different bases, while [11] provided us two optimal settings.)

The total number of XOR gate within the multiplication of $\boldsymbol{A}$ is only 32. Our S-box, S_2 or S_3 has only one regular-structured matrix $\boldsymbol{A}$, while the hardware implementation of S_1 has to deal with two irregular structured matrixes, $\boldsymbol{T}$ and $\boldsymbol{A} \cdot \boldsymbol{T}_1^{-1}$. Further optimization would reach less gate counts and critical path. If $\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x}$ where $\boldsymbol{A}$ is defined in (2), then

$$\begin{cases} y_7 = x_5 + X_2 + X_1 \\ y_6 = x_6 + X_6 \\ y_5 = x_1 + X_6 \\ y_4 = X_4 + X_3 \\ y_3 = x_7 + x_3 + X_5 \\ y_2 = X_5 + X_2 \\ y_1 = X_7 + X_3 \\ y_0 = x_4 + x_0 + X_7 \end{cases} \quad with \quad \begin{cases} X_7 = x_5 + X_2 \\ X_6 = x_5 + X_4 \\ X_5 = X_3 + x_2 \\ X_4 = X_1 + x_2 \\ X_3 = x_1 + x_0 \\ X_2 = x_7 + x_6 \\ X_1 = x_4 + x_3 \end{cases} . \tag{10}$$

Apparently, there are totally 18 XORs in (10), and 4 XORs in the critical path. From Table I, we can see that our S-boxes have a great advantage over the known results.

TABLE II. TEST RESULTS ON ANF OF S_i

Terms	f_7	f_6	f_5	f_4	f_3	f_2	f_1	f_0	**Sum**
S_1	110	112	114	131	136	145	133	132	1013
S_2	130	119	131	117	126	132	132	129	1016
S_3	119	114	132	126	126	126	128	134	1005

The reduction in our S-box also optimises the countermeasure against side-channel attack [15], such as differential power analysis [16], which uses statistical analysis of physical quantities to deduce certain information about the secret key. S_1 can be effectively masked under composite field, so do S_2 and S_3. The only difference is that we only need to mask A for S_2 and S_3, which is clearly more effective. Also for S_3, the inversion on $GF(((2^2)^2)^2)$ can be split into that on $GF((2^2)^2)$, and can be split into $GF(2^2)$, where $x^{-1} = x^2$, and inversion becomes linear. Hence, S_3 might be easier to be masked.

Our S-boxes are suitable for the encryption(decryption) within the area-limited hardware productions, such as flash memory cards, smart cards and mobile phones. Furthermore, the idea to employ the composite filed to construct the S-box is highly recommended in the design of the lightweight block cipher [17].

IV. Cryptographic Characteristics of S_2 and S_3

In this Section, a security evaluation of S_2 and S_3 will be given by comparing some common cryptographic characteristics with those of S_1. Denote $S(x) = (f_{m-1}(x_{n-1}, \cdots, x_0), ..., f_0(x_{n-1}, \cdots, x_0)) : GF(2^n) \mapsto GF(2^m)$ as the S-box transformation, with $f_i(x), m-1 \geq i \geq 0$ the n-tuple Boolean function of the i_{th} output bit.

A. *Non-Linearity, Differential Distribution, Algebraic Degree, and Algebraic Complexity*

By simple calculations, the Non-Linearity (NL) [18], the differential distributions [18], and the algebraic degree [18] of both S_2 and S_3 stay the same as S_1, and they show almost the same number of terms in their algebraic normal form (ANF) [18]; see Table II. In terms of algebraic complexity, since the structure of both S_2 and S_3 are entirely the same as S_1, restricting the polynomial in each own field makes more sense. It has been proven that every S-box with the form $\boldsymbol{A} \cdot x^{-1} + \boldsymbol{v}$ has only 9 terms in its polynomial expression, so does the $GF((2^4)^2)$ polynomial of S_2 and $GF(((2^2)^2)^2)$ polynomial of S_3, which equally show the ability against the interpolation attack [19].

B. *Algebraic Immunity*

Algebraic Immunity comes from the algebraic attack [20]. For an $n \times n$ S-box, it is defined as $\Gamma = ((t-r)/n)^{\lceil t/r \rceil}$, where r denotes the total number of linear independent equations, and t denotes the number of monomials appearing in the equations, including the constant terms.

For S_1 in (1), where $b = inv(a),\ a, b \in GF(2^8)$, one can find $r = 24$ bi-affine equations between a_i and b_j. The first set of eight equations comes from simplifying the following polynomial in the bases $\{1, z, z^2, \cdots, z^7\}$:

$$\Big(\sum_{i=0}^{7} a_i z^i\Big) \cdot \Big(\sum_{i=0}^{7} b_i z^i\Big)\ mod\ O(z) = 1. \tag{11}$$

The second set of eight equations is derived from simplifying any one equation from the group of the following $GF(2^8)$ equations:

$$a = a^2 \cdot b, a^2 = a^4 \cdot b^2, \cdots, a^{128} = a \cdot b^{128}. \tag{12}$$

Since these eight $GF(2^8)$-equations in (12) are linearly equivalent with each other, every two different $GF(2^8)$-equations from (12) will generate two different but linearly dependent sets of 8 $GF(2)$-equations between $\{a_i\}$ and $\{b_j\}$. The remaining 8 equations comes from the symmetry with respect to the exchange of a and b [20]. Adding the affine relationship $\boldsymbol{c} = \boldsymbol{A} \cdot \boldsymbol{b} + \boldsymbol{v}$, all the $\{b_j\}$ can be replaced by $\{c_k\}$, and then totally 24 bi-affine equations between $\{a_i\}$ and $\{c_k\}$ are obtained. The monomials of the system are: $\{1, a_0, a_1, \cdots, a_7, c_0, c_1, \cdots, c_7, a_0c_0, a_0c_1, \cdots, a_7c_7\}$, therefore $t = 1 + 8 + 8 + 8 \times 8 = 81$.

For S_2, $b = inv'(a)$, we have

$$\Big[\Big(\sum_{i=4}^{7} a_i y^{i-4}\Big)x + \sum_{i=0}^{3} a_i y^i\Big] \cdot \Big[\Big(\sum_{i=4}^{7} b_i y^{i-4}\Big)x + \sum_{i=0}^{3} b_i y^i\Big] \equiv 1 \ mod \ M(x) \ mod \ N(y) \tag{13}$$

Unlike (11), the bases for (13) are $\{1, y, y^2, y^3, x, yx, y^2x, y^3x\}$, still $n = 8$ equations are get. In the same way, one can get another eight equations from any one of the group (12) defined in $GF((2^4)^2)$, and eight more by exchanging a and b. Our simulation proved that these $r = 24$ equations are linearly independent, and if adding eight more from expanding any one equation from the $GF((2^4)^2)$ group (12) (or exchanging a and b), does not change the rank of the system. Our simulation shows that for S_2, $r = 24$. Similarly, t stays unchanged. Therefore, the Algebraic Immunity of S_2 is the same as S_1. For S_3, the result is also the same.

C. Influence on AES

Considering the coherence for the calculational field for the sake of the analysis of the algebraic attack, we suggest all the computation in AES being defined in the same field according to the chosen S-box, which means that the matrix multiplication for the MixColumn operation will be done in $GF((2^4)^2)$ if S_2 is used or in $GF(((2^2)^2)^2)$ if S_3 is used. Denote the AES with S_1 replaced by S_2 and $GF((2^4)^2)$ matrix multiplication for MixColumn by AES$_{cf}$, and similarly with S_3 and $GF(((2^2)^2)^2)$ matrix multiplication by AES$_{tf}$.

By then, we can conclude that both AES$_{cf}$ and AES$_{tf}$ are immune against linear attack, differential attack and interpolation attack. While for the other attacks not related to S-boxes, such as the square attack [21], the collisions attack [22] and related-key attack [23], both AES$_{cf}$ and AES$_{tf}$ have the same immunity as AES.

V. Algebraic Attack on AES$_{cf}$ and AES$_{tf}$

Our improvement on S_2 or S_3 is simply the change of field. In order to deeper demonstrate the advantage of the composite field, a concrete algebraic attack on AES$_{cf}$ and AES$_{tf}$ will be given. There are two ways to develop the algebraic attack, one is put forward by N. T. Courtois and J. Pieprzyk in Asia Crypt 2002 based on a $GF(2)$-system [20], and the other is found by S. Murphy and M. J. Robshaw in CRYPTO 2002 with only simple algebraic operations in $GF(2^8)$ [24]. The $GF(2^8)$-system created in [24] is less complicated than the $GF(2)$-system derived in [20], which indicates that any change in the field evolved during the encryption should be considered in algebraic attack, that is why three case are discussed below.

A. The $GF(2)$-Algebraic Attack

Firstly, put AES$_{cf}$ in the $GF(2)$-system. From Section IV-B, the $GF(2)$-system of S_2 is very similar to that of S_1, for they have the same algebraic lmmunity. For AddRoundkey and ShiftRows, their $GF(2)$-system is apparently equal in scale. It is easily seen that in $GF(2^8)$ or $GF((2^4)^2)$, constant multiplication can be represented by a 8-order $GF(2)$-matrix-vector multiplication. The operation in MixColumn is equivalent to a 32-order $GF(2)$-matrix-vector multiplication. In [20], it was proved that the $GF(2)$-system of AES is too complicated to be solved, therefore, AES$_{cf}$ is safe from the $GF(2)$ algebraic attack.

B. The $GF((2^4)^2)$-Algebraic Attack

In [24], AES is embedded within the Big Encryption System (BES) which uses algebraic operations in $GF(2^8)$ and can be described by a system of multivariate quadratic equations in $GF(2^8)$ simpler than the $GF(2)$-system in [20]. Analogously, we could embed AES$_{cf}$ within a BES-like cipher, and get a $GF((2^4)^2)$-system of multivariate quadratic equations. Even though this is better than the $GF(2)$-system of AES$_{cf}$, the solvability remains the same as that of the $GF(2^8)$ system of BES because these two systems are equal in scale.

C. The $GF(2^4)$-Algebraic Attack

To be more accurate, the arithmetic within AES$_{cf}$ is in $GF(2^4)$ rather than $GF((2^4)^2)$. So, we may try to split the round function of AES$_{cf}$ in $GF(2^4)$. (Most of the notations from [24] will be used below with the same indication.)

First of all, embed AES$_{cf}$ within another BES-like cipher called BES$_{cf}$. Define a mapping ϕ from $GF(2^4)$ to $(GF(2^4))^4, \phi(a) = (a^{2^0}, a^{2^1}, a^{2^2}, a^{2^3})$, and regard $\boldsymbol{a} \in (GF((2^4)^2))^{16}$ (the state variable of AES$_{cf}$) as a column vector, where

$$\boldsymbol{a} = (a_{00}, \cdots, a_{30}, a_{01}, \cdots, a_{31}, \cdots, a_{33})^T. \tag{14}$$

Each $a_{ij} \in GF((2^4)^2)$ can be split as $a_{ij} = a_{ij1}x + a_{ij0}$, so that the state space of AES$_{cf}$ is actually $(GF(2^4))^{32}$. The function ϕ can be extended to the state space of AES$_{cf}$: $\phi(\boldsymbol{a}) = (\phi(a_{001}), \phi(a_{000}), \cdots, \phi(a_{331}), \phi(a_{330}))$, and the state space of BES$_{cf}$ is $(GF(2^4))^{128}$. Using $\phi(a_{ijm}) = (b_{ijm0}, \cdots, b_{ijm3}), i, j = 0, \cdots, 3, m = 1, 0$, every BES$_{cf}$ state vector $\boldsymbol{b}$ can be denoted as

$$\begin{aligned}\boldsymbol{b} = (&b_{0010}, \cdots, b_{0013}, b_{0000}, \cdots, b_{0003},\\ &\cdots, b_{3310}, \cdots, b_{3313}, b_{3300}, \cdots, b_{3303})^T.\end{aligned} \tag{15}$$

Our aim is to give every operation of AES$_{cf}$ a $GF(2^4)$ expression, and extend it to BES$_{cf}$ using the way of [24].

As noted in [24], the additive constant 0X63 in S_1 (noted by $\boldsymbol{v}$ in (1)) can be removed by incorporating it within a modified key schedule, and so does S_2 for AES$_{cf}$. Our reductions are as follows.

1) Subkey addition: This is the same as BES, just a bitwise XOR operation on $GF(2^4)^{128}$.

2) S-box inversion: Since the inversion of S_2 is defined on $GF((2^4)^2)$, it will act on each pair of $(a_{ij1}, a_{ij0}), i, j = 0, \cdots, 3$. From (3), the $GF(2^4)$ expression is get

$$\begin{cases} inv'(a_{ij1}, a_{ij0}) = (a_{ij1} \cdot t^{-1}, (a_{ij1} + a_{ij0}) \cdot t^{-1}),\\ t = a_{ij1}^2 \cdot \lambda + a_{ij1} \cdot a_{ij0} + a_{ij0}^2. \end{cases}$$

For every eight consecutive elements $(b_{ij10}, \cdots, b_{ij03})$ from $\boldsymbol{b}$, the S-box inversion of BES$_{cf}$ can be expressed as:

$$\begin{cases} t_m = b_{ij1(m+1)} \cdot \lambda^{2^m} + b_{ij1m} \cdot b_{ij0m} + b_{ij0(m+1)}, \\ b_{ij1m} \mapsto t_m^{-1} \cdot b_{ij1m}, \\ b_{ij0m} \mapsto t_m^{-1} \cdot (b_{ij1m} + b_{ij0m}). \end{cases}$$

Here, $m = 0, \cdots, 3$, and $m+1$ is interpreted modulo 4.

3) S-box linear operation: Use Lagrange interpolation in $GF((2^4)^2)$, one can get

$$\begin{aligned} l(a) =& '06'a +' 4B'a^2 +' F6'a^4 +' 89'a^8 \\ &+' 46'a^{16} +' C0'a^{32} +' 8F'a^{64} +' 24'a^{128}, \end{aligned} \tag{16}$$

where $l(a)$ denotes the $GF(2)$ matrix multiplication of S_2. Furthermore, set $a = a_1x + a_0$ and by only simple calculations on $GF(2^4)$, (16) can be converted from $GF((2^4)^2)$ to $GF(2^4)$:

$$l(a_1x + a_0) = \Big[\sum_{i=0}^{3}(l_i a_1^{2^i} + l_{i+4}a_0^{2^i})\Big]x + \sum_{i=0}^{3}(l_{i+4}a_1^{2^i} + l_i a_0^{2^i}),$$

where $(l_0, \cdots, l_7) = ('0', 'B', '9', 'D', '4', '8', '7', 'A')$. Also the following vector form is used,

$$\begin{aligned} a_1 &\mapsto (l_0, \cdots, l_7) \cdot \tilde{\boldsymbol{a}}, \\ a_0 &\mapsto (l_4, \cdots, l_7, l_0, \cdots, l_3) \cdot \tilde{\boldsymbol{a}}, \\ \tilde{\boldsymbol{a}} &= (a_1, a_1^2, a_1^4, a_1^8, a_0, a_0^2, a_0^4, a_0^8)^T. \end{aligned}$$

The extension to BES$_{cf}$ requires a 128×128 $GF(2^4)$ matrix $\boldsymbol{Lin_B}$, a block diagonal matrix with 16 identical blocks, that is, $\boldsymbol{Lin_B} = Diag_{16}(\boldsymbol{L_B})$, where $\boldsymbol{L_B} = \begin{pmatrix} \boldsymbol{L_{B1}} & \boldsymbol{L_{B2}} \\ \boldsymbol{L_{B2}} & \boldsymbol{L_{B1}} \end{pmatrix}$ and

$$\boldsymbol{L_{B1}} = \begin{pmatrix} l_0^{2^0} & l_1^{2^0} & l_2^{2^0} & l_3^{2^0} \\ l_1^{2^1} & l_2^{2^1} & l_3^{2^1} & l_0^{2^1} \\ l_2^{2^2} & l_3^{2^2} & l_0^{2^2} & l_1^{2^2} \\ l_3^{2^3} & l_0^{2^3} & l_1^{2^3} & l_2^{2^3} \end{pmatrix}, \boldsymbol{L_{B2}} = \begin{pmatrix} l_4^{2^0} & l_5^{2^0} & l_6^{2^0} & l_7^{2^0} \\ l_5^{2^1} & l_6^{2^1} & l_7^{2^1} & l_4^{2^1} \\ l_6^{2^2} & l_7^{2^2} & l_4^{2^2} & l_5^{2^2} \\ l_7^{2^3} & l_4^{2^3} & l_5^{2^3} & l_6^{2^3} \end{pmatrix}.$$

4) ShiftRows: This can be represented as a 128×128 $GF(2^4)$ matrix $\boldsymbol{R_B}$ when we only need to ensure every two vector conjugates (8 elements) are moved as a single entity.

5) MixColumn: We have assumed this operation on $GF((2^4)^2)$. For AES$_{cf}$ it can be represented as a 8×8 $GF(2^4)$ matrix $\boldsymbol{C_A}$, that is

$$\begin{pmatrix} a_{0i1} \\ a_{0i0} \\ a_{1i0} \\ a_{1i1} \\ a_{2i0} \\ a_{2i1} \\ a_{3i0} \\ a_{3i1} \end{pmatrix} \mapsto \begin{pmatrix} y & 0 & y+1 & 0 & 1 & 0 & 1 & 0 \\ 0 & y & 0 & y+1 & 0 & 1 & 0 & 1 \\ 1 & 0 & y & 0 & y+1 & 0 & 1 & 0 \\ 0 & 1 & 0 & y & 0 & y+1 & 0 & 1 \\ 1 & 0 & 1 & 0 & y & 0 & y+1 & 0 \\ 0 & 1 & 0 & 1 & 0 & y & 0 & y+1 \\ y+1 & 0 & 1 & 0 & 1 & 0 & y & 0 \\ 0 & y+1 & 0 & 1 & 0 & 1 & 0 & y \end{pmatrix} \cdot \begin{pmatrix} a_{0i1} \\ a_{0i0} \\ a_{1i0} \\ a_{1i1} \\ a_{2i0} \\ a_{2i1} \\ a_{3i0} \\ a_{3i1} \end{pmatrix}.$$

Here, y is a root of $N(y)$ that defines $GF(2^4)$ in (8). To maintain the conjugacy for extension to BES$_{cf}$, four matrixes are needed: $p = 0, \cdots, 3, \boldsymbol{C_B^{(p)}} =$

$$\begin{pmatrix} y^{2^p} & 0 & (y+1)^{2^p} & 0 & 1 & 0 & 1 & 0 \\ 0 & y^{2^p} & 0 & (y+1)^{2^p} & 0 & 1 & 0 & 1 \\ 1 & 0 & y^{2^p} & 0 & (y+1)^{2^p} & 0 & 1 & 0 \\ 0 & 1 & 0 & y^{2^p} & 0 & (y+1)^{2^p} & 0 & 1 \\ 1 & 0 & 1 & 0 & y^{2^p} & 0 & (y+1)^{2^p} & 0 \\ 0 & 1 & 0 & 1 & 0 & y^{2^p} & 0 & (y+1)^{2^p} \\ (y+1)^{2^p} & 0 & 1 & 0 & 1 & 0 & y^{2^p} & 0 \\ 0 & (y+1)^{2^p} & 0 & 1 & 0 & 1 & 0 & y^{2^p} \end{pmatrix}.$$

and $\boldsymbol{C_B^{(0)}} = \boldsymbol{C_A}$ and if $(b_0, \cdots, b_7)^T = \boldsymbol{C_A} \cdot (a_0, \cdots, a_7)^T$, then $(b_0^{2^p}, \cdots, b_7^{2^p})^T = \boldsymbol{C_B^{(p)}} \cdot (a_0^{2^p}, \cdots, a_7^{2^p})^T$. The whole operation can be represented as a 128×128 $GF(2^4)$ matrix $\boldsymbol{Mix_B}$ and by a simple basis re-ordering $\boldsymbol{Mix_B}$ is a block diagonal of sixteen 8×8 matrices.

6) Key Schedule: The key length for AES$_{cf}$ is sixteen bytes, while for BES$_{cf}$ it is sixty-four bytes. The 64-byte BES$_{cf}$ key will generate eleven subkeys with the same length. Additionally, the embedded image of the AES$_{cf}$ key k_A is the BES$_{cf}$ key $k_B = \phi(k_A)$, then for every round subkey $(k_B)_i = \phi((k_A)_i)$, the same as BES.

7) A multivariate quadratic $GF(2^4)$-system for BES$_{cf}$: As usual, the S-box linear operation, ShiftRows and MixColumn can be combined into one Matrix denoted by $\boldsymbol{M_B} = \boldsymbol{Mix_B} \cdot \boldsymbol{R_B} \cdot \boldsymbol{Lin_B}$. Denote $\boldsymbol{p}, \boldsymbol{c} \in (GF(2^4))^{128}$ as the plaintext and ciphertext, $\boldsymbol{k}_i \in (GF(2^4))^{128}, i = 0, \cdots, 10$ as the eleven BES$_{cf}$ subkeys, and the state vectors before and after the i^{th} invocation of the $GF((2^4)^2)$ inversion layer by $\boldsymbol{w}_i \in (GF(2^4))^{128}$ and $\boldsymbol{x}_i \in (GF(2^4))^{128}$, $i = 0, \cdots, 9$ respectively, with $\boldsymbol{t}_i \in (GF(2^4))^{64}$ a temporary variable during the $GF((2^4)^2)$ inversion. For each vector above except $\boldsymbol{t}_i$, four subscripts are given, $(j, m, p, q), j, m, q = 0, \cdots, 3, p = 0, 1$, where j, m indicate the $(4*j+m)^{th}$ component corresponding $a_{jm} \in GF((2^4)^2)$ in (14), p indicates one of the two $GF(2^4)$ segments of a_{jm} and q represents the coordinate of conjugate. For $\boldsymbol{t}_i$, the subscript p is discarded since $\boldsymbol{t}_i$ is used only for $GF((2^4)^2)$ inversion where both the $GF(2^4)$ segments of a_{jm} mingle. The BES$_{cf}$ encryption can then be described by the following $GF(2^4)$ systems:

$$0 = w_{0,(j,m,p,q)} + p_{(j,m,p,q)} + k_{0,(j,m,p,q)}, \tag{17}$$

for $i = 0, \cdots, 9$,

$$\begin{aligned} 0 =& t_{i,(j,m,q)} + w_{i,(j,m,1,q+1)} \cdot \lambda^{2^q} \\ &+ w_{i,(j,m,1,q)} \cdot w_{i,(j,m,0,q)} + w_{i,(j,m,0,q+1)}, \end{aligned} \tag{18}$$

$$0 = t_{i,(j,m,q)} \cdot x_{i,(j,m,1,q)} + w_{i,(j,m,1,q)}, \tag{19}$$

$$\begin{aligned} 0 =& t_{i,(j,m,q)} \cdot x_{i,(j,m,0,q)} + w_{i,(j,m,1,q)} \\ &+ w_{i,(j,m,0,q)}, \end{aligned} \tag{20}$$

$$0 = w_{i,(j,m,p,q)}^2 + w_{i,(j,m,p,q+1)}, \tag{21}$$

$$0 = x_{i,(j,m,p,q)}^2 + x_{i,(j,m,p,q+1)}, \tag{22}$$

$$0 = t_{i,(j,m,q)}^2 + t_{i,(j,m,q+1)}, \tag{23}$$

for $i = 1, \cdots, 9$,

$$\begin{aligned} 0 =& w_{i,(j,m,p,q)} + k_{i,(j,m,p,q)} + \\ &\sum_{(j',m',p',q')} \alpha_{(j',m',p',q')} \cdot x_{i-1,(j',m',p',q')}, \end{aligned} \tag{24}$$

$$\begin{aligned} 0 =& c_{(j,m,p,q)} + k_{10,(j,m,p,q)} + \\ &\sum_{(j',m',p',q')} \beta_{(j',m',p',q')} \cdot x_{9,(j',m',p',q')}. \end{aligned} \tag{25}$$

TABLE III. ITEMS FOR THE $GF(2^4)$ SYSTEM OF BES$_{cf}$

Eq.	Num.	Property	Increased Variables	Increased Quadratic Terms
(17)	128	linear	256	0
(18)	640	quadratic	1792	640
(19)	640	quadratic	640	640
(20)	640	quadratic	640	640
(21)	1280	quadratic	0	1280
(22)	1280	quadratic	0	1280
(23)	640	quadratic	0	640
(24)	1152	linear	1152	0
(25)	128	linear	128	0

TABLE IV. ITEMS FOR THE SYSTEM OF BES AND BES$_{cf}$

Items	BES [24]	BES$_{cf}$
Equations in total	5248	6528
Linear Equations	1408	1408
Quadratic Equations	3840	5120
Terms in total	7808	9728
Quadratic Terms	3840	5120
State Variables	2560	3200
Key Variables	1408	1408

Note that the equations in (21), (22) and (23) indicate conjugacy. In (24) and (25), $\alpha_{(j',m',p',q')}$ and $\beta_{(j',m',p',q')}$ denote the elements in $\boldsymbol{M_B}$ and $\boldsymbol{M_B^*} = \boldsymbol{R_B} \cdot \boldsymbol{Lin_B}$ respectively, and $q+1$ is interpreted modulo 4. The numbers of items of the $GF(2^4)$ systems are listed in Table III.

The system contains 6528 equations, of which 1408 are linear and 5120 are (extremely sparse) quadratic equations. The system comprises 9728 terms made from 3200 state variables and 1408 key variables, of which 4608 are linear terms (state variables and key variables), 3200 are square terms and 1920 are quadratic terms. The details are listed in Table IV.

The effectiveness for the algebraic attack lies in the solvability of the system. Courtois and Pieprzyk [20] present a method called XSL to solve the $GF(2)$-system for AES, and it is also available for the $GF(2^8)$-system for BES. However, up till now, there exists no authentic estimation for XSL attack, so it is of no worth to give a complete comparison of the $GF(2^8)$ attack for BES and the $GF(2^4)$ attack for BES$_{cf}$. However, we find three evidences leading to the insolvability of the $GF(2^4)$-system for BES$_{cf}$:

(I) The $GF(2^4)$-system has more terms and more equations. The complexity of the XSL algorithm is on average $O(T^\omega)$ [20]; here T denotes the number of terms.

(II) The $GF(2^4)$-system for BES$_{cf}$ has more quadratic equations than the $GF(2^8)$-system for BES. The quadratic terms of BES$_{cf}$ occupy $5120/9728 = 52.6\%$ in total, more than $3840/7808 = 49.2\%$ for BES. According to the analysis in [20] , the $GF(2^4)$-system for BES$_{cf}$ would be more complex to carry out XSL attack than the $GF(2^8)$-system for BES. In fact, most of the extra terms of BES$_{cf}$ are the new variables $\boldsymbol{t_i}$ used for the special treatment with composite field inversion.

(III) In [25], there is another definition of Algebraic Immunity for the XSL S-box. For each XSL S-box of BES$_{cf}$, the related equations are (18)-(23). First, the size $n = 8$ is fixed because each XSL S-box works on eight state variables of BES$_{cf}$. The terms appearing in (18)-(23) are: $t_{i,(j,m,q)}$, $w_{i,(j,m,p,q)}$, $x_{i,(j,m,p,q)}$, $w_{i,(j,m,1,q)} \cdot w_{i,(j,m,0,q)}$, $t_{i,(j,m,q)} \cdot x_{i,(j,m,1,q)}$, $t_{i,(j,m,q)} \cdot x_{i,(j,m,0,q)}$, $w^2_{i,(j,m,p,q)}$, $x^2_{i,(j,m,p,q)}$, $t^2_{i,(j,m,q)}$. Summing up according to the subscripts p and q, the number of terms is $t = 4+8+8+4+4+4+8+8+4 = 52$ and the number of equations $r = 4+4+4+8+8+4 = 32$, then the Algebraic Immunity for XSL S-box of BES$_{cf}$ is $\Gamma = (\frac{t-r}{n})^{\lceil \frac{t-r}{n} \rceil} = 2.5^3 = 15.625$ which is larger than 9.6 of BES. The S-box based on composite field seems more immune from the potential algebraic attack.

Furthermore, adding the key schedule, the system may not be any simpler since it has the same number of S-box substitutions as BES. By then, we can conclude that AES$_{cf}$ is immune from the $GF(2^4)$-system, and the potential algebraic attack for AES$_{cf}$ may not work.

Similarly, one can consider AES$_{tf}$ in a $GF((2^2)^2)$ system and get a $GF((2^2)^2)$-system for AES$_{tf}$, with the same scale of system of equations as the $GF(2^4)$-system for AES$_{cf}$. And even more, one can think of splitting the $GF((2^2)^2)$ system into the field $GF(2^2)$, where the basic $GF((2^2)^2)$ operations, especially the inversion, have to be replaced by the basic operations on $GF(2^2)$. However, based on what we have done before, splitting $GF((2^4)^2)$ inversion into basic operations on $GF(2^4)$, which complicates the system, one can see that the expected $GF(2^2)$-system for AES$_{cf}$ may not be any simpler.

VI. Conclusion

In this paper, we tried to change the computational field used in AES S-box, and created a new class of S-box with better efficiency while preserving the cryptographical security. Two 8×8 S-boxes S_2 and S_3 are constructed, by direct inversion in composite fields $GF((2^4)^2)$ and $GF(((2^2)^2)^2)$ respectively, combined with a $GF(2)$ affine transformation, the same used in AES S-box to give a rational comparison of the composite field. The choice of the subfield leading to the most efficient implementation is mainly discussed. By simple comparison, our new S-boxes have better hardware implementation than AES S-box. The masking strategy against differential power attack is also more convenient.

We also studied the cryptographic characteristics with such a S-box based on composite field inversion. The results show that both S_2 and S_3 have comparatively the same cryptographic characteristics with AES S-box. Thus, the replacement to composite field does not weaken the cryptographic characteristics. Moreover, we investigated whether or not those effective cryptanalysis of AES might work if our S-box took the place, especially the algebraic attack. Due to the different fields involved, algebraic attacks applied on $GF(2)$, $GF((2^4)^2)$ and $GF(2^4)$ are discussed, respectively. And we proved that with the replacement of S_2 or S_3 and the corresponding field for MixColumn operation, the revised AES, denoted by AES$_{cf}$ or AES$_{tf}$, had no effective algebraic attack and was even more solid than the original AES with S_1.

In fact, the essence of our design is just to try to overlook the underlying computational fashion and to choose the most efficient one while preserving the cryptographic characteristics. The most compact AES S-box to date was created by normal bases [11]. The advantage for normal bases is that they have very sparse matrixes in the implementation compared with polynomial bases [11], but finding inversion will be as hard as under polynomial bases. The S-box constructed on normal bases would also survive those attacks.

TABLE V. COMPOSITE FIELD IN BLOCK CIPHER

Cipher	Field	Structure
AES	$GF(2^8)$	$\boldsymbol{A} \cdot x^{-1} + \boldsymbol{v}$
SMS4	$GF(2^8)$	$\boldsymbol{A} \cdot (\boldsymbol{A} \cdot x + \boldsymbol{v})^{-1} + \boldsymbol{v}$
CLEFIA	$GF(2^8)$	$\boldsymbol{A}_2 \cdot (\boldsymbol{A}_1 \cdot x + \boldsymbol{v}_1)^{-1} + \boldsymbol{v}_2$
Camellia	$GF((2^4)^2)$	$\boldsymbol{A}_2 \cdot (\boldsymbol{A}_1 \cdot (x + \boldsymbol{v}_1))^{-1} + \boldsymbol{v}_2$

Based on our argument, we suggest composite field $GF((2^n)^2)$ in the design of block cipher. And we think that our settings for S_2 or S_3 is indeed a balance between the implementation complexity and the theoretical security. It seems that the designers of block ciphers did not truly realize the advantage of $GF((2^n)^2)$, see Table V. Even though Camellia uses composite field, the structure is the most complex. As a result, we suggest SMS4 and CLEFIA use composite field for a more efficient hardware implementation.

ACKNOWLEDGMENT

This work is supported by Major Program of National Natural Science Foundation of China (NO.11290141), the International Cooperation Project (NO.2010DFR00700), and the Fundamental Research of Civil Aircraft (NO.MJ-F-2012-04). We are grateful to the Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education for funding us to fulfil this work.

REFERENCES

[1] K. Nyberg, “Differentially uniform mappings for cryptography,” in Advances in Cryptology – EUROCRYPT 93, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 1994, vol. 765, pp. 55–64.

[2] J. Daemen and V. Rijmen, The design of Rijndael: AES–the Advanced Encryption Standard, ser. Information security and cryptography. Springer, 2002.

[3] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, “Camellia : A 128-bit block cipher suitable for multiple platforms - design and analysis,” in Selected Areas in Cryptography, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2001, vol. 2012, pp. 39–56.

[4] S. Corporation, “The 128-bit blockcipher CLEFIA – algorithm specification (revision 1.0),” 2007, URL: http://www.sony.net/Products/cryptography/clefia/download/data/clefia-spec-1.0.pdf [retrieved: July, 2014].

[5] W. Diffie and G. Ledin, “SMS4 encryption algorithm for wireless networks,” Cryptology ePrint Archive, Report 2008/329, 2008, URL: http://eprint.iacr.org/2008/329.pdf [retrieved: July, 2014].

[6] B. Sunar, E. Savas, and C. Koc, “Constructing composite field representations for efficient conversion,” IEEE Transactions on Computers, vol. 52, no. 11, nov. 2003, pp. 1391 – 1398.

[7] A. Rudra, P. K. Dubey, C. S. Jutla, V. Kumar, J. R. Rao, and P. Rohatgi, “Efficient Rijndael encryption implementation with composite field arithmetic,” in Cryptographic Hardware and Embedded Systems – CHES 2001, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2001, vol. 2162, pp. 171–184.

[8] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, “A compact Rijndael hardware architecture with S-Box optimization,” in Advances in Cryptology – ASIACRYPT 2001, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2001, vol. 2248, pp. 239–254.

[9] N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, “A systematic evaluation of compact hardware implementations for the Rijndael S-Box,” in Topics in Cryptology - CT-RSA 2005, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3376, pp. 323–333.

[10] D. Canright, “A very compact S-Box for AES,” in Cryptographic Hardware and Embedded Systems - CHES 2005, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3659, pp. 441–455.

[11] X. Zhang and K. K. Parhi, “On the optimum constructions of composite field for the AES algorithm,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 53, no. 10, oct. 2006, pp. 1153–1157.

[12] S. Nikova, V. Rijmen, and M. Schläffer, “Using normal bases for compact hardware implementations of the AES S-Box,” in Security and Cryptography for Networks, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5229, pp. 236–245.

[13] Y. Nogami, K. Nekado, T. Toyota, N. Hongo, and Y. Morikawa, “Mixed bases for efficient inversion in F$(((2^2)^2)^2)$ and conversion matrices of subbytes of AES,” in Cryptographic Hardware and Embedded Systems, CHES 2010, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6225, pp. 234–247.

[14] B. Hannes, C. Andreas, and H. Max, “On computing multiplicative inverses in GF(2^m),” IEEE Transactions on Computers, vol. 42, no. 8, aug 1993, pp. 1010 –1015.

[15] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen, “A side-channel analysis resistant description of the AES S-Box,” in Fast Software Encryption, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3557, pp. 199–228.

[16] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in Advances in Cryptology – CRYPTO 99, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 1999, vol. 1666, pp. 789–789.

[17] O. Özen, K. Varıcı, C. Tezcan, and Ç. Kocair, “Lightweight block ciphers revisited: Cryptanalysis of reduced round present and hight,” in Information Security and Privacy, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5594, pp. 90–107.

[18] C. Carlet, “Boolean functions for cryptography and error correcting codes,” in Boolean Models and Methods in Mathematics, Computer Science, and Engineering, Y. Crama and P. L. Hammer, Eds. Cambridge University Press, 2010, pp. 257–397, URL: http://www.math.univ-paris13.fr/∼carlet/chap-fcts-Bool-corr.pdf [(a preliminary version) retrieved: July, 2014].

[19] T. Jakobsen and L. R. Knudsen, “The interpolation attack on block ciphers,” in Fast Software Encryption, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 1997, vol. 1267, pp. 28–40.

[20] N. T. Courtois and J. Pieprzyk, “Cryptanalysis of block ciphers with overdefined systems of equations,” Cryptology ePrint Archive, Report 2002/044, 2002, URL: http://eprint.iacr.org/2002/044.pdf [retrieved: July, 2014].

[21] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, “Improved cryptanalysis of Rijndael,” in Fast Software Encryption, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2001, vol. 1978, pp. 136–141.

[22] H. Gilbert and M. Minier, “A collision attack on 7 rounds of Rijndael,” in AES Candidate Conference’00, 2000, pp. 230–241.

[23] A. Biryukov and D. Khovratovich, “Related-key cryptanalysis of the full aes-192 and aes-256,” in Advances in Cryptology - ASIACRYPT 2009, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5912, pp. 1–18.

[24] S. Murphy and M. J. Robshaw, “Essential algebraic structure within the AES,” in Advances in Cryptology – CRYPTO 2002, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2002, vol. 2442, pp. 1–16.

[25] S. Oh, C. H. Kim, J. Lim, and D. H. Cheon, “Remarks on security of the AES and the XSL technique,” Electronics Letters, vol. 39, no. 1, Jan 2003, pp. 36–38.