A Compact S-Box Design for SMS4 Block Cipher

Imran Abbasi and Mehreen Afzal
College of Telecommunication(MCS)
National University of Sciences and Technology,
Islamabad, Pakistan
imranabbasi@ mcs.edu.pk.com, mehreenafzal@mcs.edu.pk

Abstract. This paper proposes a compact design of SMS4 S-box using combinational logic which is suitable for the implementation in area constraint environments like smart cards. The inversion algorithm of the proposed S-box is based on composite field $GF(((2^2)^2)^2)$ using normal basis at all levels. In our approach, we examined all possible normal basis combinations having trace equal to one at each subfield level. There are 16 such possible combinations with normal basis and we have compared the S-box designs based on each case in terms of logic gates it uses for implementation. The isomorphism mapping and inverse mapping bit matrices are fully optimized using greedy algorithm. We prove that our best case reduces the complexity upon the SMS4 S-box design with existing inversion algorithm based on polynomial basis by 15% XOR and 42% AND gates.

Keywords: Composite field arithmetic, SMS4, Normal Basis, S-box

1 Introduction

SMS4 is the mandatory block cipher standard for securing Wireless Local Area Network (WLAN) devices in China. The Office of State Commercial Cipher Administration of China (OSCCA) released the cipher description in January, 2006 [8] and the English version of the document is published by Diffie and Ledin [9]. SMS4 is used in WLAN Authentication and Privacy Infrastructure (WAPI) standard in order to provide data confidentiality. The Chinese WLAN industry widely uses WAPI, and it is supported by many international corporations like SONY in the relevant products.

The efficiency of SMS4 hardware implementation in terms of power consumption, area and throughput mainly depends upon the implementation of its S-box. It is the most computationally intensive operational structure of SMS4 as it comprises of non-linear multiplicative inversion. The designers of the SMS4 had chosen its S-box design similar to Rijndael which employs inversion base mapping [14]. Implementing a circuit to find the multiplicative inverse in the $GF(2^8)$ using Extended Euclidean algorithm or Fermat theorem is very complex and costly. Several architectures of $GF(2^8)$ inverter have been proposed by researchers over the period of time for area efficient implementation of S-boxes that comprises of inversion in their algebraic expressions. An efficient way to implement S-box is to use combinational logic because it requires small area for implementation. V. Rijmen [3] proposed the first

hardware implementation of AES S-box using composite field representation. The proposed design suggested the use of Optimal Normal Basis for efficient inversion in $GF(2^8)$. J. Wolkerstorfer [1] and A.Rudra [5] implemented the AES S-box by representing $GF(2^8)$ as a quadratic extension of the $GF(2^4)$ using polynomial basis. In this approach a byte in $GF(2^8)$ is first decomposed into linear polynomial with coefficients in $GF(2^4)$ and different arithmetic operations in $GF(2^4)$ are computed using combinational logic. The inversion in hardware is then implemented with the simple logic gates by further decomposing $GF(2^4)$ into $GF(2^2)$ operations. Satoh [6] and Mentens [7] further optimized the hardware implementation of AES S-box by applying a composite field with multiple extensions of smaller degrees. The tower field $GF(2^8)\rightarrow GF(((2^2)^2)^2)$ is constructed with repeated degree 2 extensions using polynomial basis. Canright in [2] analyzed all possible combinations of normal and polynomial basis at subfield levels of $GF(((2^2)^2)^2)$ and proved that use of normal bases at all levels of composite field decomposition further reduces the area of the AES S-box implementation. X. Bai [4] proposed a $GF(2^8)$ inversion algorithm for SMS4 S-box based on slight modification of design in [1].

In this paper, a new combinational structure of SMS4 S-box with the inversion algorithm in tower field representation $GF(2^8)\rightarrow GF(((2^2)^2)^2)$ based on normal basis, has been proposed. We have analyzed all possible combinations of normal basis at each level with trace one from the field generated by irreducible primitive polynomial of SMS4 cipher. The comparison of our resulting best case architecture with the S-box design based on proposed $GF(2^8)$ inverter of [4] is also given.

The organization of the rest of paper is as follows. In subsequent section, structure of SMS4 block cipher is briefly described with the focus on its S-box. In section 3, the design of S-box using the composite field representation with normal basis is explicated. Section 4 gives the comparison of combinatorial S-box designs of SMS4 with different normal basis combinations at subfield level. In section 5, a comparative analysis is given between our proposed design of S-box with the one based on the inversion algorithm presented in [4]. Conclusions and work in progress are stated in section 6.

2 The SMS4

SMS4 block cipher is based on the iterative fiestel structure with input, output, and key size of 128 bits each. The data input is divided into four 32 bit words. The algorithm comprises of 32 rounds, and in each round one word is modified by adding it to other three words with a keyed function. Encryption and decryption processes have the similar structure and only the key schedule is reversed. For the detailed description of cipher one may refer to [9]. The official depiction of SMS4 S-box is given as a lookup table (LUT) with 256 entries. The S-box is commonly implemented with the ROM lookup table where the pre-computed values are stored. However, significant hardware resources are required if lookup table is implemented with 16×16 entries. SMS4 S-box is bijective and it substitutes byte input for byte output using arithmetic computations over $GF(2^8)$. A method suitable for hardware implementation of S-box is to first perform affine transformation on $GF(2)$, then carry out inversion in

$GF(2^8)$, followed by second affine transformation over $GF(2)$ [13,14]. The S-box algebraic structure is given as the following expression [13].

$$S(x) = A_2(A_1.x + C_1)^{-1} + C_2 . \quad (1)$$

The row vectors are C_1 = 0xCB = $(11001011)_2$ and C_2 = 0xD3 = $(11010011)_2$. The cyclic matrices A_1 and A_2 in the algebraic expression are as below:

$$A_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (2)$$

The irreducible primitive polynomial in $GF(2^8)$ is

$$f(x) = (x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1) . \quad (3)$$

3 SMS4 S-box Design in Composite Field

In this section we describe the proposed SMS4 combinatorial structure based on composite field $GF(((2^2)^2)^2)$ in normal basis with the logical equations for inversion, multiplications, squaring and addition. SMS4 S-box design in composite field arithmetic is more efficient than using ROM/RAM for lookup tables (LUT) in area constrained environments [4]. All finite fields of same cardinality are isomorphic but their arithmetic efficiency depends significantly on the choice of basis that is used for the field element representation. For the hardware implementation, normal basis has significant advantage over polynomial basis as mathematical operations in normal basis representation generally comprises of rotation, shifting and XORing [11, 12].

3.1 $GF(2^8)$ Inversion Algorithm using Normal Basis

For input byte x to SMS4 S-box, inverse is computed for the expression (A_1.x + C_1). The complexity of basis conversion is dependent on the selected irreducible polynomial and if the polynomial is adequately chosen, the basis conversion is simple [7]. Following are the irreducible polynomials and their corresponding normal basis representation.

$GF(2^2)$	$: z^2 + z + 1$	$\rightarrow (z + Z)(z + Z^2)$	Normal basis (Z^2, Z)
$GF((2^2)^2)$	$: y^2 + Ty + N$	$\rightarrow (y + Y)(y + Y^4)$	Normal basis (Y^4, Y)
$GF(((2^2)^2)^2)$	$: x^2 + \tau x + n$	$\rightarrow (x + X)(x + X^{16})$	Normal basis (X^{16}, X)

(4)

Where $T = Y^4 + Y$ is the trace and $N = Y^4 .Y$ is the norm in $GF(2^4)/GF(2^2)$, $\tau = X^{16} + X$ is the trace and $n = X^{16} .X$ is the norm in $GF(2^8)/GF(2^4)$. To minimize the operations and simplify inversion circuit in composite field we consider only those basis combinations which have $\tau = T = 1$. The nested structure of $GF(2^8)$ inverter comprises of different subfield operations. In the following sections logical structures for inversion, multiplication and scaling in composite field are given.

Inversion in $GF(2^8)$, $GF(2^4)$ and $GF(2^2)$. Let the pair $(a_h, a_l) \in GF(2^4)$ represents a $\in GF(2^8)$ in terms of Normal basis (X^{16}, X). If $b \in GF(2^8)$ is inverse of a, then product of a and b is 1.

$$\begin{aligned} a &= a_h X^{16} + a_l X \\ b &= b_h X^{16} + b_l X \\ a \times b &= (a_h X^{16} + aX)(b_h X^{16} + bX) = 1 . \end{aligned} \tag{5}$$

Substituting $X + X^{16} = 1$, $(X^{16})^2 = X^{16} + n$ and $(X)^2 = X + n$ and solving for b_h and b_l.

$$\begin{aligned} b_h &= [(a_h \otimes a_l) \oplus ((a_h \oplus a_l)^2 \otimes n)]^{-1} \otimes a_l . \\ b_l &= [(a_h \otimes a_l) \oplus ((a_h \oplus a_l)^2 \otimes n)]^{-1} \otimes a_h . \end{aligned} \tag{6}$$

Where $\otimes$ is multiplication and $\oplus$ is addition in $GF(2^4)$. If $\Omega = [(a_h \otimes a_l) \oplus ((a_h \oplus a_l)^2 \otimes n)]^{-1}$, then inversion in $GF(2^8)$ is expressed by following relation.

$$b = a^{-1} = (\Omega \otimes a_l) X^{16} + (\Omega \otimes a_h) X . \tag{7}$$

The logical structure of $GF(2^8)$ inverter is shown in figure 1. Similarly, if $c \in GF(2^4)$ and it has an inverse $d \in GF(2^4)$ using normal basis (Y^4, Y), then $c = c_h Y^4 + c_l Y$, $c_h, c_l \in GF(2^2)$ and $d = d_h Y^4 + d_l Y$ $d_h, d_l \in GF(2^2)$. If $\otimes$ is multiplication and $\oplus$ is bitwise addition in $GF(2^2)$ and $\Phi = [(c_h \otimes c_l) \oplus ((c_h \oplus c_l)^2 \otimes N)]^{-1}$, then equation for $GF(2^4)$ inversion is given as below:

$$d = c^{-1} = (\Phi \otimes c_l) Y^4 + (\Phi \otimes c_h) Y . \tag{8}$$

The $GF(2^4)$ inverter is depicted in figure 2. The inversion in $GF(2^2)$ is same as squaring and implemented without gates by swapping of bits. If $e \in GF(2^2)$ is represented in normal basis (Z^2, Z) as $e = e_h Z^2 + e_l Z$, $e_h, e_l \in GF(2)$ and f is the inverse of e in $GF(2^2)$ then inversion in $GF(2^2)$ is:

$$f = e^{-1} = (e_l) Z^2 + (e_h) Z . \tag{9}$$

Multiplication in $GF(2^4)$ and $GF(2^2)$. The structures of multipliers in $GF(2^4)$ and $GF(2^2)$ in normal basis are derived as below.

$$(c_h Y^4 + c_l Y)(d_h Y^4 + d_l Y) = c_h d_h (Y^4)^2 + c_h d_l Y^4 Y + c_l d_h Y^4 Y + c_l d_l Y^2 \tag{10}$$

Substituting $Y + Y^4 = 1$, $(Y^4)^2 = Y^4 + N$ and $(Y)^2 = Y + N$.

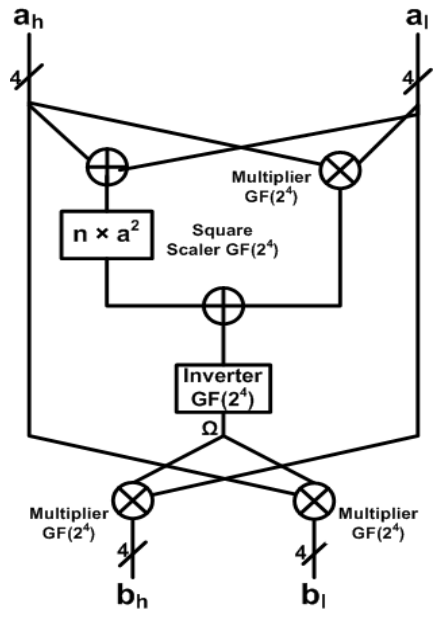

Fig. 1. $GF(2^8)$ Inverter

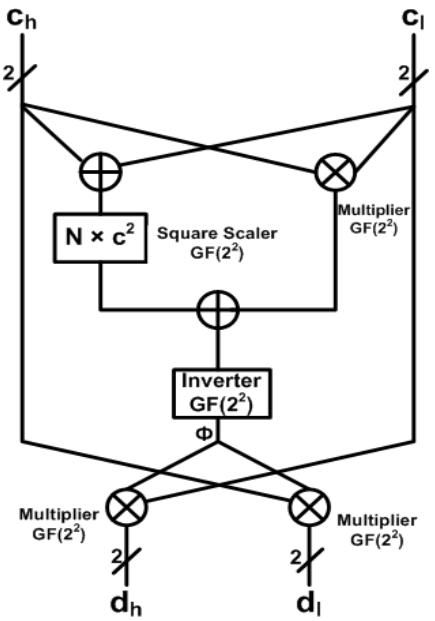

Fig. 2. $GF(2^4)$ Inverter

$$(c_h Y^4 + c_l Y)(d_h Y^4 + d_l Y) = (c_h d_h \oplus \epsilon) Y^4 + (c_l d_l \oplus \epsilon) Y . \quad (11)$$

Where $\oplus$ is bit wise addition, $\otimes$ is multiplication in $GF(2^2)$ and $\epsilon = (c_h \oplus c_l) \otimes (d_h \oplus d_l) \otimes N$. Similarly $GF(2^2)$ multiplier in normal basis is represented as:

$$(e_h Z^2 + e_l Z)(f_h Z^2 + f_l Z) = (e_h f_h \oplus \Lambda) Z^2 + (e_l f_l \oplus \Lambda) Z . \quad (12)$$

$\oplus$ represents the bit addition, $\otimes$ is AND operation and $\Lambda = (e_h \oplus e_l) \otimes (f_h \oplus f_l)$. The above mentioned structures are illustrated in figure 3 and figure 4 respectively.

Scaling and Squaring in $GF(2^4)$ and $GF(2^2)$. In $GF(2^8)$ and $GF(2^4)$ inverters there are constant multiplication operations ($n \times a^2$) and ($N \times c^2$) and in $GF(2^4)$ multiplier there is constant multiplication term ($N \times c$). The combination of squaring and scaling operation results in further optimization [2]. The computation of these terms depends on the values of n in $GF(2^4)$ and N in $GF(2^2)$ for the chosen normal basis. $N \in GF(2^2)$ and N is not equal to zero or one, therefore N and N+1 are the roots of $z^2 + z + 1$. So depending on the choice of basis, scalars for N and N^2 implies to scalars for z or z^2. The two bit factor ($N \times c$) is given in two ways.

$$\begin{aligned} Z \times (e_h Z^2 + e_l Z) &= (e_h \oplus e_l) Z^2 + e_h Z . \\ Z^2 \times (e_h Z^2 + e_l Z) &= e_l Z^2 + (e_h \oplus e_l) Z . \end{aligned} \quad (13)$$

Similarly the square scaling two bit factor ($N \times c^2$) is represented in following two ways depending upon choice of conjugate basis pair.

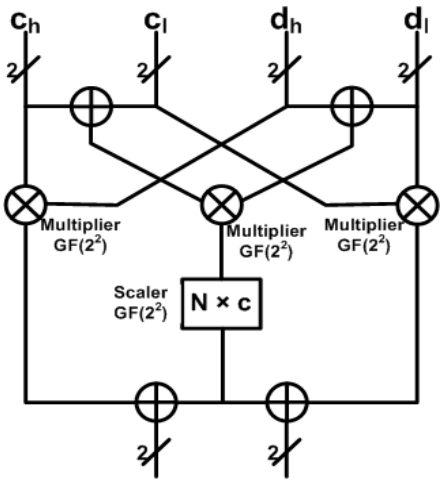

Fig. 3. *GF*(2^4) Multiplier

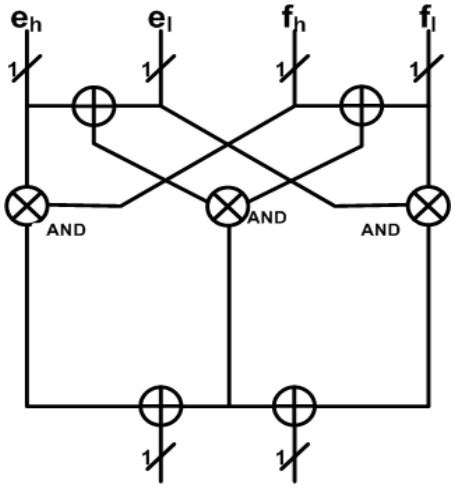

Fig. 4. *GF*(2^2) Multiplier

$$\begin{aligned} Z \times (e_h Z^2 + e_l Z)^2 &= (e_h \oplus e_l) Z^2 + e_l Z \,. \\ Z^2 \times (e_h Z^2 + e_l Z)^2 &= e_h Z^2 + (e_h \oplus e_l) Z \,. \end{aligned} \tag{14}$$

The scaling operation (n x a^2) is a four bit factor in *GF*(2^8) inverter and its computation in *GF*(2^2) depends on the normal basis types and the relation between norm n and N as in [2]. For computations in *GF*(2^4), tables in appendix 'B' are used.

3.2 Generating Isomorphic and Inverse Mapping Functions

The standard SMS4 form is defined by 8 bit vector as coefficients of powers of x which is root of irreducible primitive polynomial in (3). Multiplicative inversion in composite field is computed after a byte in *GF*(2^8) is mapped to its composite field representation using isomorphism function δ [6]. After the multiplicative inverse is computed in the composite field, the 8 bit result is mapped back to standard equivalent representation in *GF*(2^8) using inverse isomorphic function δ^{-1}. The isomorphic and it inverse mapping is one to one and onto mapping and is represented as 8×8 matrix [10]. If byte s is in standard polynomial basis then it can be represented as a quadratic extension as $s = a_h X^{16} + a_l X$, $a_h, a_l \in GF(2^4)$, where each 4 bit coefficient is represented as $c = c_h Y^4 + c_l Y$, $c_h, c_l \in GF(2^2)$, each of which is then further represented as pair of bits $e = e_h Z^2 + e_l Z$ in $GF(2^2)/GF(2)$. If the new byte is given as $t_7 t_6 t_5 t_4 t_3 t_2 t_1 t_0$ then we have the following expression [2].

$$\begin{aligned} & s_7 S^7 + s_6 S^6 + s_5 S^5 + s_4 S^4 + s_3 S^3 + s_2 S^2 + s_1 S^1 + s_0 S^0 \\ &= \{(t_7 Z^2 + t_6 Z) Y^4 + (t_5 Z^2 + t_4 Z) Y\} X^{16} \\ &+ \{(t_3 Z^2 + t_2 Z) Y^4 + (t_1 Z^2 + t_0 Z) Y\} X \,. \\ &= t_7 Z^2 Y^4 X^{16} + t_6 Z Y^4 X^{16} + t_5 Z^2 Y X^{16} + t_4 Z Y X^{16} + t_3 Z^2 Y^4 X + t_2 Z Y^4 X + \\ &\quad t_1 Z^2 Y X + t_0 Z Y X \,. \end{aligned} \tag{15}$$

The values of X, Y and Z are substituted from the conjugate basis chosen and these 8 hexadecimal values with coefficient t_i represents the columns of 8 × 8 reverse base transformation matrix δ^{-1}. The inverse matrix δ is used for changing standard basis to corresponding composite field representation [2]. The inverse mapping matrix δ^{-1} is combined with affine transformation matrix A_2 for further optimization as in [6]. The block diagram of SMS4 S-box is given in the figure below.

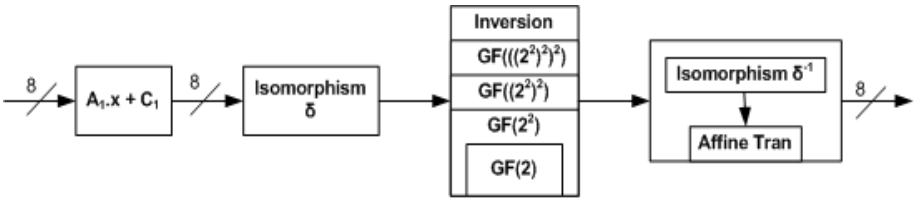

Fig. 5. SMS4 S-box Block Diagram

4 Results

For the possible choices of norms in $GF(2^4)$ and $GF(2^2)$ along with the normal basis at each subfield level satisfying τ = T = 1, we have 16 possible cases as shown in appendix 'A'. SMS4 S-box design based on each case is fully tested and simulated. The most compact case is the one which gives the least number of XOR gates for implementation. It can be observed from the results in table 1 that choosing different normal basis combination results in small difference in number of XOR gates. These small differences exist due to different mapping matrices and slight differences in the inverter architectures. The matrices operations for mapping, inverse mapping and affine transformation are fully optimized using greedy algorithm [10]. The greedy algorithm operates iteratively on the mentioned matrices determining the occurrences of all possible repeating pairs in the output. The repeating pairs are pre-computed to reduce the number of XOR gates. Our best case S-box design (case 5, table 1) saves 35 XOR gates by application of greedy algorithm.

The $GF(2^8)$ inverter in normal basis comprises of one $GF(2^4)$ inverter, three $GF(2^4)$ multipliers, one square scaling and two additions in $GF(2^4)$ as shown in figure 1. One $GF(2^4)$ inversion is computed using three multipliers, one inversion, one square scaling and two additions in $GF(2^2)$ as depicted in figure 2, where one $GF(2^4)$ multiplier comprises of three multipliers, four additions and a scaling operation in $GF(2^2)$ as in figure 3. Thus total number of logic gates computed in hierarchical structure of inverter for our best case S-box is 91 XOR and 36 AND. The structures of multipliers in figure 3 and figure 4 depicts that it requires summation of high and low halves of each input factor. If the same factor is shared by two different multipliers then share factor can save one subfield addition [2]. Thus, a four bit common factor in one $GF(2^4)$ multiplier can save five XOR gates and a two bit common factor in $GF(2^2)$ multiplier can save one XOR gate. In $GF(2^8)$ inverter in figure 1, all three $GF(2^4)$ multipliers have share factors i.e. Ω, ah, a_l are all shared between respective two $GF(2^4)$ multipliers thus saving 15 XOR gates. Similarly in $GF(2^4)$ normal inverter we have Φ, c_h, c_l shared between respective two $GF(2^4)$ multipliers thus

saving 3 XOR gates. In total 15+3 = 18 XOR gates can be saved by the share factors in $GF(2^8)$ and $GF(2^4)$ normal inverters in hardware implementation. Thus total number of gates required for case 5 SMS4 S-box are 73 XOR and 36 AND gates.

Table. 1. All Cases of SMS4 S-box design using Normal basis in $GF(((2)^2)^2)^2$

No	Conjugate Ordered Pair Basis			Logic Gates S-box	
	(X^{16}, X)	**(Y^4,Y)**	**(Z^2, Z)**	**XOR**	**AND**
1	(0x98, 0x99)	(0x51, 0x50)	(0x5C, 0x5D)	137	36
2	(0x98, 0x99)	(0x0C, 0x0D)	(0x5C, 0x5D)	135	36
3	(0xBF, 0xBE)	(0x51, 0x50)	(0x5C, 0x5D)	135	36
4	(0xBF, 0xBE)	(0x0C, 0x0D)	(0x5C, 0x5D)	139	36
5	(0x94, 0x95)	(0x51, 0x50)	(0x5C, 0x5D)	134	36
6	(0x94, 0x95)	(0x0C, 0x0D)	(0x5C, 0x5D)	136	36
7	(0xEF, 0xEE)	(0x51, 0x50)	(0x5C, 0x5D)	138	36
8	(0xEF, 0xEE)	(0x0C, 0x0D)	(0x5C, 0x5D)	136	36
9	(0xC5, 0xC4)	(0x51, 0x50)	(0x5C, 0x5D)	136	36
10	(0xC5, 0xC4)	(0x0C, 0x0D)	(0x5C, 0x5D)	136	36
11	(0xE3, 0xE2)	(0x51, 0x50)	(0x5C, 0x5D)	139	36
12	(0xE3, 0xE2)	(0x0C, 0x0D)	(0x5C, 0x5D)	136	36
13	(0xC9, 0xC8)	(0x51, 0x50)	(0x5C, 0x5D)	138	36
14	(0xC9, 0xC8)	(0x0C, 0x0D)	(0x5C, 0x5D)	139	36
15	(0xB3, 0xB2)	(0x51, 0x50)	(0x5C, 0x5D)	137	36
16	(0xB3, 0xB2)	(0x0C, 0x0D)	(0x5C, 0x5D)	137	36

5 Comparative Analysis

Our most compact SMS4 S-box comprises of 134 XOR and 36 AND gates with conjugate pair basis (0x94, 0x95), (0x51, 0x50) and (0x5C, 0x5D) respectively. We provide comparison of our most compact case 5 S-box design with the one based on $GF(2^8)$ inversion algorithm proposed in [4] that uses polynomial basis. The operations in the subfield and the number of XOR and AND logic gates required to design SMS4 S-box based on [4] is given in table 3. The matrices computations are optimized using greedy algorithm as in [1].

Table. 2. Logic gates for our Best case SMS4 S-box

Mathematical Operation	**XOR**	**AND**
Affine Trans 1 (A_1. x + C_1)	29	-
Map $GF(2^8) \rightarrow GF((2^2)^2)^2$	15	-
Map inv + Affine Trans 2	17	-
$GF(2^8)$ Inversion	73	36
Total	134	36

Table. 3. Logic Gates for SMS4 S-box based on Polynomial Basis Inverter of [4]

Mathematical Operation	Instances	XOR	AND
Affine Trans 1 ($x.A_1 + C_1$)	1	29	-
Map $GF(2^8) \rightarrow GF(2^4)^2$	1	12	-
Map inv $GF(2^4)^2 \rightarrow GF(2^8)$	1	10	-
Map $GF(2^4) \rightarrow GF(2^2)^2$	1	3	-
Map inv $GF(2^2)^2 \rightarrow GF(2^4)$	1	2	-
Affine Trans 2 ($y.A_2 + C_2$).	1	29	-
$GF(2^4)$ Multiplier	3	45	48
$GF(2^4)$ Squaring	1	2	-
$GF(2^4)$ Scaling	1	1	-
$GF(2^4)$ Addition	2	8	-
$GF(2^2)$ Multiplier	3	9	15
$GF(2^2)$ Squaring	1	1	-
$GF(2^2)$ Scaling	1	1	-
$GF(2^2)$ Addition	2	4	-
$GF(2^2)$ Inverter	1	1	-
	Total	157	63

6 Conclusion and Future Work

In this paper we have proposed an improved design for SMS4 S-box based on the combinational logic with a low gate count. The proposed algorithm for computing SMS4 S-box function is based on composite field $GF(((2^2)^2)^2)$ and we have simulated all the possible cases of subfield combination depending upon the choice of normal basis, from which we have determined the best case. All the transformation matrices are optimized using greedy algorithm. We have proved that our best case S-box design results in much lower gate count and reduces the complexity by 15% XOR gates and 42% AND gates over the S-box based on the inversion algorithm of [4]. Our compact architecture of SMS4 S-box can save a significant amount of chip area in the hardware implementation of SMS4 in ASICs and it can be used for area constrained and demanding throughput SMS4 integrated circuits for applications ranging from smart cards to high speed processing units. The future work will concentrate on the ASIC implementation of the S-box, where our design can be further improved using the logic gate optimizations depending on specific CMOS standard library.

References

[1] Wolkerstorfer, J., Oswald, E., Lamberger, M.: An ASIC Implementation of the AES Sboxes. In: CT-RSA, LNCS, vol. 2271, pp. 67–78. Springer, Heidelberg (2002)

[2] Canright, D.: A Very Compact Rijndael S-box.Technical Report NPS-MA-04- 001,Naval Postgraduate School (September 2004)
http://web.nps.navy.mil/~dcanrig/pub/NPS-MA-05-001.pdf
[3] Rijmen, V.:Efficient Implementation of the Rijndael S-box (2000)
www.iaik.tugraz.at/RESEARCH/krypto/AES/old/~rijmen/rijndael/sbox.pdf
[4] Bai, X., Xu, Y., Guo, L.: Securing SMS4 Cipher against Differential Power Analysis and its VLSI implementation. In: ICCS (2008)
[5] Rudra, A., Dubey P., Jutla, C., Kumar, V., Rao, J., Rohatgi, P.: Efficient Rijndael encryption implementation with composite field arithmetic. In: CHES 2001, LNCS, pp. 171–184, Heidelberg (2001)
[6] Satoh, A., Morioka, S., Takano, K., Munetoh, S.: A compact Rijndael hardware architecture with S-box optimization. In: ASIACRYPT 2001, LNCS, vol. 2248, pp. 239–254. Springer Heidelberg (2001).
[7] Mentens, N., Batina, L., Preneel, B., Verbauwhede, I.: A systematic evaluation of compact hardware implementations for the Rijndael S-box. In: CT-RSA, LNCS, vol. 3376, pp 323-333. Springer, Heidelberg (2005)
[8] Office of State Commercial Cipher Administration of China. SMS4 cipher for WLAN products. http://www.oscca.gov.cn/UpFile/200621016423197990.pdf, 2006
[9] Diffie, W., Ledin, G.: SMS4 encryption algorithm for wireless networks. Cryptology ePrint Archive, Report 2008/329, 2008. http://eprint.iacr.org/
[10] Paar, C.:Efficient VLSI architectures for bit parallel computation in Galois fields. Ph.D thesis, Institute for Experimental Mathematics, University of Essen, (1994)
[11] Lidl, R., Niederreiter, H.: Introduction to Finite Fields and their Applications. Cambridge University Press, New York, USA (1986)
[12] Deschamps, J., Sutter, G., Imana, J.: Hardware Implementation of Finite Field Arithmetic. McGraw-Hill Professional ISBN: 978-0-07-154582-2 (2009)
[13] Erickson, J., Ding, J., Christensen, C.: Algebraic cryptanalysis of SMS4: Grobner basis attack and SAT attack compared. In : ICISC (2009)
[14] Liu, F., Ji, W., Hu, L., Ding, J., Shuwang, L., Pyshkin, A., R.P. Weinmann, R.: Analysis of the SMS4 Block Cipher. In: ACISP, LNCS, vol. 4586, pp. 158–170. Springer, Heidelberg (2007)

Appendix A: $GF(2^8)$ Representation for SMS4 S-Box

A.1 The table below gives the decimal, hexadecimal and binary values of the $GF(2^8)$ generated modulo irreducible primitive polynomial $f(x) = x^8+x^7+x^6+x^5+x^4+x^2+1$. Let A be the root of f(x) then the field generated with respective names of elements is as below:

Dec	Hex	Binary	θ^i	Name	Dec	Hex	Binary	θ^i	Name
0	00	00000000	-	0	39	27	00100111	θ^{187}	β^4
1	01	00000001	θ^0	1	40	28	00101000	θ^{16}	A^{16}
2	02	00000010	θ^1	A	41	29	00101001	θ^{104}	G^8
3	03	00000011	θ^{134}	G^{128}	42	2A	00101010	θ^{153}	γ^8
4	04	00000100	θ^2	A^2	43	2B	00101011	θ^{119}	β^8
5	05	00000101	θ^{13}	G	44	2C	00101100	θ^{176}	F^{16}
6	06	00000110	θ^{135}	H^{128}	45	2D	00101101	θ^{223}	q^{32}
7	07	00000111	θ^{76}	J^4	46	2E	00101110	θ^{169}	b^2
8	08	00001000	θ^3	B	47	2F	00101111	θ^{114}	d^{128}
9	09	00001001	θ^{210}	a^{16}	48	30	00110000	θ^{138}	K^{128}
10	0A	00001010	θ^{14}	D^2	49	31	00110001	θ^{250}	n
11	0B	00001011	θ^{174}	g^{16}	50	32	00110010	θ^{241}	m^2
12	**0C**	**00001100**	$\boldsymbol{\theta^{136}}$	$\boldsymbol{\alpha^8}$	51	33	00110011	θ^{160}	C^{32}
13	**0D**	**00001101**	$\boldsymbol{\theta^{34}}$	$\boldsymbol{\alpha^2}$	52	34	00110100	θ^{36}	E^4
14	0E	00001110	θ^{77}	b^{16}	53	35	00110101	θ^{82}	P^{16}
15	0F	00001111	θ^{147}	d^4	54	36	00110110	θ^{90}	a^2
16	10	00010000	θ^4	A^4	55	37	00110111	θ^{96}	B^{32}
17	11	00010001	θ^{26}	G^2	56	38	00111000	θ^{79}	k^{16}
18	12	00010010	θ^{211}	k^4	57	39	00111001	θ^{47}	j^{16}
19	13	00010011	θ^{203}	j^4	58	3A	00111010	θ^{54}	N^2
20	14	00010100	θ^{15}	H	59	3B	00111011	θ^{220}	e^{32}
21	15	00010101	θ^{152}	J^8	60	3C	00111100	θ^{149}	Q^{128}
22	16	00010110	θ^{175}	n^{16}	61	3D	00111101	θ^{50}	M^2
23	17	00010111	θ^{168}	K^8	62	3E	00111110	θ^{10}	C^2
24	18	00011000	θ^{137}	J^{128}	63	3F	00111111	θ^{31}	m^{32}
25	19	00011001	θ^{240}	H^{16}	64	40	01000000	θ^6	B^2
26	1A	00011010	θ^{35}	M^{32}	65	41	01000001	θ^{165}	a^{32}
27	1B	00011011	θ^{89}	Q^8	66	42	01000010	θ^{144}	E^{16}
28	1C	00011100	θ^{78}	d^{16}	67	43	01000011	θ^{73}	P^{64}
29	1D	00011101	θ^{53}	b^{64}	68	44	01000100	θ^{28}	D^4
30	1E	00011110	θ^{148}	P^4	69	45	01000101	θ^{93}	g^{32}
31	1F	00011111	θ^9	E	70	46	01000110	θ^{111}	l^{16}
32	20	00100000	θ^5	C	71	47	01000111	θ^{184}	L^8
33	21	00100001	θ^{143}	m^{16}	72	48	01001000	θ^{213}	g^2
34	22	00100010	θ^{27}	N	73	49	01001001	θ^{193}	D^{64}
35	23	00100011	θ^{110}	e^{16}	74	4A	01001010	θ^{58}	f^{64}
36	24	00100100	θ^{212}	b	75	4B	01001011	θ^{181}	c^2
37	25	00100101	θ^{57}	d^{64}	76	4C	01001100	θ^{205}	e^2
38	26	00100110	θ^{204}	γ^4	77	4D	01001101	θ^{99}	N^{32}

Dec	Hex	Binary	θ^{i}	Name	Dec	Hex	Binary	θ^{i}	Name
78	4E	01001110	θ^{188}	j^{64}	123	7B	01111011	θ^{238}	β
79	4F	01001111	θ^{61}	k^{64}	124	7C	01111100	θ^{11}	F
80	**50**	**01010000**	$\boldsymbol{\theta^{17}}$	$\boldsymbol{\alpha}$	125	7D	01111101	θ^{253}	q^{2}
81	**51**	**01010001**	$\boldsymbol{\theta^{68}}$	$\boldsymbol{\alpha^{4}}$	126	7E	01111110	θ^{32}	A^{32}
82	52	01010010	θ^{105}	a^{8}	127	7F	01111111	θ^{208}	G^{16}
83	53	01010011	θ^{129}	B^{128}	128	80	10000000	θ^{7}	D
84	54	01010100	θ^{154}	b^{32}	129	81	10000001	θ^{87}	g^{8}
85	55	01010101	θ^{39}	d^{8}	130	82	10000010	θ^{166}	b^{8}
86	56	01010110	θ^{120}	H^{8}	131	83	10000011	θ^{201}	d^{2}
87	57	01010111	θ^{196}	J^{64}	132	84	10000100	θ^{145}	M^{16}
88	58	01011000	θ^{177}	N^{16}	133	85	10000101	θ^{172}	Q^{4}
89	59	01011001	θ^{230}	e	134	86	10000110	θ^{74}	P^{2}
90	5A	01011010	θ^{224}	D^{32}	135	87	10000111	θ^{132}	E^{128}
91	5B	01011011	θ^{234}	g	136	88	10001000	θ^{29}	f^{32}
92	**5C**	**01011100**	$\boldsymbol{\theta^{170}}$	$\boldsymbol{\lambda^{2}}$	137	89	10001001	θ^{218}	c
93	**5D**	**01011101**	$\boldsymbol{\theta^{85}}$	$\boldsymbol{\lambda}$	138	8A	10001010	θ^{94}	j^{32}
94	5E	01011110	θ^{115}	e^{128}	139	8B	10001011	θ^{158}	k^{32}
95	5F	01011111	θ^{216}	N^{8}	140	8C	10001100	θ^{112}	D^{16}
96	60	01100000	θ^{139}	L^{128}	141	8D	10001101	θ^{117}	g^{128}
97	61	01100001	θ^{246}	l	142	8E	10001110	θ^{185}	e^{64}
98	62	01100010	θ^{251}	q^{4}	143	8F	10001111	θ^{108}	N^{4}
99	63	01100011	θ^{22}	F^{2}	144	90	10010000	θ^{214}	c^{8}
100	64	01100100	θ^{242}	j	145	91	10010001	θ^{232}	f
101	65	01100101	θ^{244}	k	146	92	10010010	θ^{194}	F^{64}
102	66	01100110	θ^{161}	G^{32}	147	93	10010011	θ^{127}	q^{128}
103	67	01100111	θ^{64}	A^{64}	**148**	**94**	**10010100**	$\boldsymbol{\theta^{59}}$	$\mathbf{h^{64}}$
104	68	01101000	θ^{37}	P	**149**	**95**	**10010101**	$\boldsymbol{\theta^{179}}$	$\mathbf{h^{4}}$
105	69	01101001	θ^{66}	E^{64}	150	96	10010110	θ^{182}	c^{64}
106	6A	01101010	θ^{83}	b^{4}	151	97	10010111	θ^{71}	f^{8}
107	6B	01101011	θ^{228}	d	**152**	**98**	**10011000**	$\boldsymbol{\theta^{206}}$	$\mathbf{h^{16}}$
108	6C	01101100	θ^{91}	c^{32}	**153**	**99**	**10011001**	$\boldsymbol{\theta^{236}}$	**h**
109	6D	01101101	θ^{163}	f^{4}	154	9A	10011010	θ^{100}	M^{4}
110	6E	01101110	θ^{97}	F^{32}	155	9B	10011011	θ^{43}	Q
111	6F	01101111	θ^{191}	q^{64}	156	9C	10011100	θ^{189}	l^{64}
112	70	01110000	θ^{80}	C^{16}	157	9D	10011101	θ^{226}	L^{32}
113	71	01110001	θ^{248}	m	158	9E	10011110	θ^{62}	m^{64}
114	72	01110010	θ^{48}	B^{16}	159	9F	10011111	θ^{20}	C^{4}
115	73	01110011	θ^{45}	a	160	A0	10100000	θ^{18}	E^{2}
116	74	01110100	θ^{55}	e^{8}	161	A1	10100001	θ^{41}	P^{8}
117	75	01110101	θ^{141}	N^{128}	162	A2	10100010	θ^{69}	K^{64}
118	76	01110110	θ^{221}	β^{2}	163	A3	10100011	θ^{125}	n^{128}
119	77	01110111	θ^{102}	γ^{2}	164	A4	10100100	θ^{106}	b^{128}
120	78	01111000	θ^{150}	a^{128}	165	A5	10100101	θ^{156}	d^{32}
121	79	01111001	θ^{24}	B^{8}	166	A6	10100110	θ^{130}	C^{128}
122	7A	01111010	θ^{51}	γ	167	A7	10100111	θ^{199}	m^{8}

Dec	Hex	Binary	θ^i	Name	Dec	Hex	Binary	θ^i	Name
168	A8	10101000	θ^{155}	e^4	212	D4	11010100	θ^{84}	K^4
169	A9	10101001	θ^{198}	N^{64}	213	D5	11010101	θ^{215}	n^8
170	AA	10101010	θ^{40}	C^8	214	D6	11010110	θ^{229}	j^2
171	AB	10101011	θ^{124}	m^{128}	215	D7	11010111	θ^{233}	k^2
172	AC	10101100	θ^{121}	j^{128}	216	D8	11011000	θ^{92}	L^4
173	AD	10101101	θ^{122}	k^{128}	217	D9	11011001	θ^{183}	l^8
174	AE	10101110	θ^{197}	L^{64}	218	DA	11011010	θ^{164}	P^{32}
175	AF	10101111	θ^{123}	l^{128}	219	DB	11011011	θ^{72}	E^8
176	B0	10110000	θ^{178}	Q^{16}	220	DC	11011100	θ^{98}	J^{32}
177	B1	10110001	θ^{70}	M^{64}	221	DD	11011101	θ^{60}	H^4
178	**B2**	**10110010**	$\boldsymbol{\theta^{231}}$	$\mathbf{p^8}$	222	DE	11011110	θ^{192}	B^{64}
179	**B3**	**10110011**	$\boldsymbol{\theta^{126}}$	$\mathbf{p^{128}}$	223	DF	11011111	θ^{180}	a^4
180	B4	10110100	θ^{225}	H^{32}	224	E0	11100000	θ^{81}	K^{16}
181	B5	10110101	θ^{19}	J	225	E1	11100001	θ^{95}	n^{32}
182	B6	10110110	θ^{235}	n^4	**226**	**E2**	**11100010**	$\boldsymbol{\theta^{249}}$	$\mathbf{p^2}$
183	B7	10110111	θ^{42}	K^2	**227**	**E3**	**11100011**	$\boldsymbol{\theta^{159}}$	$\mathbf{p^{32}}$
184	B8	10111000	θ^{171}	g^4	228	E4	11100100	θ^{49}	J^{16}
185	B9	10111001	θ^{131}	D^{128}	229	E5	11100101	θ^{30}	H^2
186	BA	10111010	θ^{86}	Q^2	230	E6	11100110	θ^{46}	L^2
187	BB	10111011	θ^{200}	M^8	231	E7	11100111	θ^{219}	l^4
188	BC	10111100	θ^{116}	f^{128}	232	E8	11101000	θ^{56}	D^8
189	BD	10111101	θ^{107}	c^4	233	E9	11101001	θ^{186}	g^{64}
190	**BE**	**10111110**	$\boldsymbol{\theta^{217}}$	$\mathbf{h^2}$	234	EA	11101010	θ^{142}	f^{16}
191	**BF**	**10111111**	$\boldsymbol{\theta^{157}}$	$\mathbf{h^{32}}$	235	EB	11101011	θ^{109}	c^{128}
192	C0	11000000	θ^{140}	M^{128}	236	EC	11101100	θ^{222}	l^{32}
193	C1	11000001	θ^{101}	Q^{32}	237	ED	11101101	θ^{113}	L^{16}
194	C2	11000010	θ^{247}	q^8	**238**	**EE**	**11101110**	$\boldsymbol{\theta^{103}}$	$\mathbf{h^8}$
195	C3	11000011	θ^{44}	F^4	**239**	**EF**	**11101111**	$\boldsymbol{\theta^{118}}$	$\mathbf{h^{128}}$
196	**C4**	**11000100**	$\boldsymbol{\theta^{252}}$	**p**	240	F0	11110000	θ^{151}	j^8
197	**C5**	**11000101**	$\boldsymbol{\theta^{207}}$	$\mathbf{p^{16}}$	241	F1	11110001	θ^{167}	k^8
198	C6	11000110	θ^{23}	L	242	F2	11110010	θ^{25}	M
199	C7	11000111	θ^{237}	l^2	243	F3	11110011	θ^{202}	Q^{64}
200	**C8**	**11001000**	$\boldsymbol{\theta^{243}}$	$\mathbf{p^4}$	244	F4	11110100	θ^{52}	G^4
201	**C9**	**11001001**	$\boldsymbol{\theta^{63}}$	$\mathbf{p^{64}}$	245	F5	11110101	θ^8	A^8
202	CA	11001010	θ^{245}	n^2	246	F6	11110110	θ^{239}	q^{16}
203	CB	11001011	θ^{21}	K	247	F7	11110111	θ^{88}	F^8
204	CC	11001100	θ^{162}	K^{32}	248	F8	11111000	θ^{12}	B^4
205	CD	11001101	θ^{190}	n^{64}	249	F9	11111001	θ^{75}	a^{64}
206	CE	11001110	θ^{65}	C^{64}	250	FA	11111010	θ^{254}	q
207	CF	11001111	θ^{227}	m^4	251	FB	11111011	θ^{133}	F^{128}
208	D0	11010000	θ^{38}	J^2	252	FC	11111100	θ^{33}	E^{32}
209	D1	11010001	θ^{195}	H^{64}	253	FD	11111101	θ^{146}	P^{128}
210	D2	11010010	θ^{67}	G^{64}	254	FE	11111110	θ^{209}	f^2
211	D3	11010011	θ^{128}	A^{128}	255	FF	11111111	θ^{173}	c^{16}

A.2 The minimal polynomials over GF(2) and their respective conjugate roots in terms of θ^i are presented in the following table.

Name	Minimal Polynomial	Conjugate Roots (θ^i)
1	$x + 1$	θ^{0}
λ	$x^2 + x + 1$	θ^{85}, θ^{170}
α	$x^4 + x + 1$	θ^{17}, θ^{34}, θ^{68}, θ^{136}
β	$x^4 + x^3 + 1$	θ^{238}, θ^{221}, θ^{187}, θ^{119}
γ	$x^4 + x^3 + x^2 + x + 1$	θ^{51}, θ^{102}, θ^{204}, θ^{153}
A	$x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1$	θ^{1}, θ^{2}, θ^{4}, θ^{8}, θ^{16}, θ^{32}, θ^{64}, θ^{128}
B	$x^8 + x^7 + x^5 + x^4 + x^3 + x^2 + 1$	θ^{3}, θ^{6}, θ^{12}, θ^{24}, θ^{48}, θ^{96}, θ^{192}, θ^{129}
C	$x^8 + x^4 + x^3 + x + 1$	θ^{5}, θ^{10}, θ^{20}, θ^{40}, θ^{80}, θ^{160}, θ^{65}, θ^{130}
D	$x^8 + x^6 + x^5 + x^4 + 1$	θ^{7}, θ^{14}, θ^{28}, θ^{56}, θ^{112}, θ^{224}, θ^{193}, θ^{131}
E	$x^8 + x^5 + x^4 + x^3 + x^2 + x + 1$	θ^{9}, θ^{18}, θ^{36}, θ^{72}, θ^{144}, θ^{33}, θ^{66}, θ^{132}
F	$x^8 + x^6 + x^3 + x^2 + 1$	θ^{11}, θ^{22}, θ^{44}, θ^{88}, θ^{176}, θ^{97}, θ^{194}, θ^{133}
G	$x^8 + x^7 + x^3 + x^2 + 1$	θ^{13}, θ^{26}, θ^{52}, θ^{104}, θ^{208}, θ^{161}, θ^{67}, θ^{134}
H	$x^8 + x^5 + x^4 + x^3 + 1$	θ^{15}, θ^{30}, θ^{60}, θ^{120}, θ^{240}, θ^{225}, θ^{195}, θ^{135}
J	$x^8 + x^5 + x^3 + x^2 + 1$	θ^{19}, θ^{38}, θ^{76}, θ^{152}, θ^{49}, θ^{98}, θ^{196}, θ^{137}
K	$x^8 + x^7 + x^6 + x^4 + x^3 + x^2 + 1$	θ^{21}, θ^{42}, θ^{84}, θ^{168}, θ^{81}, θ^{162}, θ^{69}, θ^{138}
L	$x^8 + x^7 + x^2 + x + 1$	θ^{23}, θ^{46}, θ^{92}, θ^{184}, θ^{113}, θ^{226}, θ^{197}, θ^{139}
M	$x^8 + x^7 + x^4 + x^3 + x^2 + 1$	θ^{25}, θ^{50}, θ^{100}, θ^{200}, θ^{145}, θ^{35}, θ^{70}, θ^{140}
N	$x^8 + x^7 + x^3 + x + 1$	θ^{27}, θ^{54}, θ^{108}, θ^{216}, θ^{177}, θ^{99}, θ^{198}, θ^{141}
P	$x^8 + x^5 + x^3 + x + 1$	θ^{37}, θ^{74}, θ^{148}, θ^{41}, θ^{82}, θ^{164}, θ^{73}, θ^{146}
Q	$x^8 + x^7 + x^6 + x^5 + x^2 + x + 1$	θ^{43}, θ^{86}, θ^{172}, θ^{89}, θ^{178}, θ^{101}, θ^{202}, θ^{149}
a	$x^8 + x^7 + x^6 + x^4 + x^2 + x + 1$	θ^{45}, θ^{90}, θ^{180}, θ^{105}, θ^{210}, θ^{165}, θ^{75}, θ^{150}
b	$x^8 + x^7 + x^6 + x^3 + x^2 + x + 1$	θ^{212}, θ^{169}, θ^{83}, θ^{166}, θ^{77}, θ^{154}, θ^{53}, θ^{106}
c	$x^8 + x^7 + x^5 + x^3 + 1$	θ^{218}, θ^{181}, θ^{107}, θ^{214}, θ^{173}, θ^{91}, θ^{182}, θ^{109}
d	$x^8 + x^7 + x^5 + x + 1$	θ^{228}, θ^{201}, θ^{147}, θ^{39}, θ^{78}, θ^{156}, θ^{57}, θ^{114}
e	$x^8 + x^7 + x^6 + x^5 + x^4 + x + 1$	θ^{230}, θ^{205}, θ^{155}, θ^{55}, θ^{110}, θ^{220}, θ^{185}, θ^{115}
f	$x^8 + x^7 + x^6 + x + 1$	θ^{232}, θ^{209}, θ^{163}, θ^{71}, θ^{142}, θ^{29}, θ^{58}, θ^{116}
g	$x^8 + x^6 + x^5 + x^4 + x^2 + x + 1$	θ^{234}, θ^{213}, θ^{171}, θ^{87}, θ^{174}, θ^{93}, θ^{186}, θ^{117}
h	$x^8 + x^6 + x^5 + x^3 + 1$	θ^{236}, θ^{217}, θ^{179}, θ^{103}, θ^{206}, θ^{157}, θ^{59}, θ^{118}
j	$x^8 + x^6 + x^5 + x + 1$	θ^{242}, θ^{229}, θ^{203}, θ^{151}, θ^{47}, θ^{94}, θ^{188}, θ^{121}
k	$x^8 + x^6 + x^5 + x^2 + 1$	θ^{244}, θ^{233}, θ^{211}, θ^{167}, θ^{79}, θ^{158}, θ^{61}, θ^{122}
l	$x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1$	θ^{246}, θ^{237}, θ^{219}, θ^{183}, θ^{111}, θ^{222}, θ^{189}, θ^{123}
m	$x^8 + x^4 + x^3 + x^2 + 1$	θ^{248}, θ^{241}, θ^{227}, θ^{199}, θ^{143}, θ^{31}, θ^{62}, θ^{124}
n	$x^8 + x^7 + x^5 + x^4 + 1$	θ^{250}, θ^{245}, θ^{235}, θ^{215}, θ^{175}, θ^{95}, θ^{190}, θ^{125}
p	$x^8 + x^6 + x^5 + x^4 + x^3 + x + 1$	θ^{252}, θ^{249}, θ^{243}, θ^{231}, θ^{207}, θ^{159}, θ^{63}, θ^{126}
q	$x^8 + x^6 + x^4 + x^3 + x^2 + x + 1$	θ^{254}, θ^{253}, θ^{251}, θ^{247}, θ^{239}, θ^{223}, θ^{191}, θ^{127}

Appendix B: Tables for $GF(2^4)$ Computations

B.1 The table below gives the decimal, hexadecimal and binary values of the $GF(2^4)$ generated modulo irreducible primitive polynomial $g(x) = x^4 + x + 1$. Let α be the root of $g(x)$ then the field generated with respective names of elements is as below:

Dec	Hex	ANF Ω^i	Bin Ω^i	Ω^i	Name
0	00	0	0000	-	0
1	01	x	0001	Ω^0	1
2	02	x^2	0010	Ω^1	α
3	03	$x + 1$	0011	Ω^4	α^4
4	04	x^2	0100	Ω^2	α^2
5	05	$x^2 + 1$	0101	Ω^8	α^8
6	06	$x^2 + x$	0110	Ω^5	λ
7	07	$x^2 + x + 1$	0111	Ω^{10}	λ^2
8	08	x^3	1000	Ω^3	γ
9	09	$x^3 + 1$	1001	Ω^{14}	β
10	0A	$x^3 + x$	1010	Ω^9	γ^8
11	0B	$x^3 + x + 1$	1011	Ω^7	β^8
12	0C	$x^3 + x^2$	1100	Ω^6	γ^2
13	0D	$x^3 + x^2 + 1$	1101	Ω^{13}	β^2
14	0E	$x^3 + x^2 + x$	1110	Ω^{11}	β^4
15	0F	$x^3 + x^2 + x + 1$	1111	Ω^{12}	γ^4

B.2 The table below gives the minimal polynomials over $GF(2)$ and their respective conjugate roots in terms of Ω^i are presented using irreducible primitive polynomial $g(x) = x^4 + x + 1$.

Name	Minimal Polynomial	Conjugate Roots (θ^i)
1	$x + 1$	Ω^0
λ	$x^2 + x + 1$	Ω^5 , Ω^{10}
α	$x^4 + x + 1$	Ω , Ω^2, Ω^4 , Ω^8
β	$x^4 + x^3 + 1$	Ω^{14}, Ω^{13} , Ω^{11}, Ω^7
γ	$x^4 + x^3 + x^2 + x + 1$	Ω^3 , Ω^6, Ω^{12} , Ω^9

B.3 The addition table in *GF*(16) using the naming convention in table A.1 is given below:

⊕	**0**	**1**	**α**	**α^2**	**γ**	**α^4**	**λ**	**γ^2**	**β^8**	**α^8**	**γ^8**	**λ^2**	**β^4**	**γ^4**	**β^2**	**β**
0	0	1	α	α^2	γ	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2	β
1	1	0	α^4	α^8	β	α	λ^2	β^2	γ^8	α^2	β^8	λ	γ^4	β^4	γ^2	γ
α	α	α^4	0	λ	γ^8	1	α^2	β^4	β	λ^2	γ	α^8	γ^2	β^2	γ^4	β^8
α^2	α^2	α^8	λ	0	γ^2	λ^2	α	γ	γ^4	1	β^4	α^4	γ^8	β^8	β	β^2
γ	γ	β	γ^8	γ^2	0	β^8	β^4	α^2	α^4	β^2	α	γ^4	λ	λ^2	α^8	1
α^4	α^4	α	1	λ^2	β^8	0	α^8	γ^4	γ	λ	β	α^2	β^2	γ^2	β^4	γ^8
λ	λ	λ^2	α^2	α	β^4	α^8	0	γ^8	β^2	α^4	γ^2	1	γ	β	β^8	γ^4
γ^2	γ^2	β^2	β^4	γ	α^2	γ^4	γ^8	0	λ^2	β	λ	β^8	α	α^4	1	α^8
β^8	β^8	γ^8	β	γ^4	α^4	γ	β^2	λ^2	0	β^4	1	γ^2	α^8	α^2	λ	α
α^8	α^8	α^2	λ^2	1	β^2	λ	α^4	β	β^4	0	γ^4	α	β^8	γ^8	γ	γ^2
γ^8	γ^8	β^8	γ	β^4	α	β	γ^2	λ	1	γ^4	0	β^2	α^2	α^8	λ^2	α^4
λ^2	λ^2	λ	α^8	α^4	γ^4	α^2	1	β^8	γ^2	α	β^2	0	β	γ	γ^8	β^4
β^4	β^4	γ^4	γ^2	γ^8	λ	β^2	γ	α	α^8	β^8	α^2	β	0	1	α^4	λ^2
γ^4	γ^4	β^4	β^2	β^8	λ^2	γ^2	β	α^4	α^2	γ^8	α^8	γ	1	0	α	λ
β^2	β^2	γ^2	γ^4	β	α^8	β^4	β^8	1	λ	γ	λ^2	γ^8	α^4	α	0	α^2
β	β	γ	β^8	β^2	1	γ^8	γ^4	α^8	α	γ^2	α^4	β^4	λ^2	λ	α^2	0

B.4 The multiplication table in *GF*(16) is given as below:

⊗	**0**	**1**	**α**	**α^2**	**γ**	**α^4**	**λ**	**γ^2**	**β^8**	**α^8**	**γ^8**	**λ^2**	**β^4**	**γ^4**	**β^2**	**β**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	α	α^2	γ	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2	β
α	0	α	α^2	γ	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2	β	1
α^2	0	α^2	γ	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2	β	1	α
γ	0	γ	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2	β	1	α	α^2
α^4	0	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2	β	1	α	α^2	γ
λ	0	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2	β	1	α	α^2	γ	α^4
γ^2	0	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2	β	1	α	α^2	γ	α^4	λ
β^8	0	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2	β	1	α	α^2	γ	α^4	λ	γ^2
α^8	0	α^8	γ^8	λ^2	β^4	γ^4	β^2	β	1	α	α^2	γ	α^4	λ	γ^2	β^8
γ^8	0	γ^8	λ^2	β^4	γ^4	β^2	β	1	α	α^2	γ	α^4	λ	γ^2	β^8	α^8
λ^2	0	λ^2	β^4	γ^4	β^2	β	1	α	α^2	γ	α^4	λ	γ^2	β^8	α^8	γ^8
β^4	0	β^4	γ^4	β^2	β	1	α	α^2	γ	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2
γ^4	0	γ^4	β^2	β	1	α	α^2	γ	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4
β^2	0	β^2	β	1	α	α^2	γ	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4
β	0	β	1	α	α^2	γ	α^4	λ	γ^2	β^8	α^8	γ^8	λ^2	β^4	γ^4	β^2