Cybersecurity

RESEARCH

Open Access

A circuit area optimization of MK-3 S-box

Yanjun Li[1,2], Weiguo Zhang[3], Yiping Lin[3*], Jian Zou[4] and Jian Liu[1]

Abstract

In MILCOM 2015, Kelly et al. proposed the authentication encryption algorithm MK-3, which applied the 16-bit S-box. This paper aims to implement the 16-bit S-box with less circuit area. First, we classified the irreducible polynomials over $\mathbb{F}_{2^n}$ into three kinds. Then we compared the logic gates required for multiplication over the finite field constructed by the three types of irreducible polynomials. According to the comparison result, we constructed the composite fields, $\mathbb{F}_{(2^4)^2}$ and $\mathbb{F}_{(2^8)^2}$. Based on the isomorphism of finite fields, the operations over $\mathbb{F}_{2^{16}}$ can be conducted over $\mathbb{F}_{(2^8)^2}$. Similarly, elements over $\mathbb{F}_{2^8}$ can be mapped to the corresponding elements over $\mathbb{F}_{(2^4)^2}$. Next, the SAT solver was used to optimize the operations over smaller field $\mathbb{F}_{2^4}$. At last, the architecture of the optimized MK-3 S-box was worked out. Compared with the implementation proposed by the original designer, the circuit area of the MK-3 S-box in this paper is reduced by at least 55.9%.

Introduction

In 2015, Wood et al. proposed a 16-bit S-box (Wood et al. 2015), based on which Kelly et al. designed the MK-3 algorithm (Kelly et al. 2015). The MK-3 S-box has excellent cryptographic security criteria and less hardware implementation cost. Its construction idea was firstly proposed in the 8-bit S-box (Daemen and Rijmen 1998) of the Advanced Encryption Standard (AES) (NIST 2001). In order to cut down the hardware resources for calculating multiplicative inverses in AES, Rijmen et al. applied the composite field arithmetic (Paar 1995; Itoh and Tsujii 1988) to map elements of field $\mathbb{F}_{2^8}$ to the composite field $\mathbb{F}_{(2^4)^2}$ based on polynomial basis (Rijmen 2000). In this way, the arithmetic in $\mathbb{F}_{2^8}$ can be reduce to the operation in the smaller subfield $\mathbb{F}_{2^4}$. However, Rijmen (2000) did not offer detailed implementation results. The optimization results of the AES S-box based on polynomial basis were presented by Satoh et al. in ASIACRYPT 2001 (Satoh et al. 2001). Next, normal basis was introduced to optimize the AES S-box by Canright et al. in CHES2005 (Canright 2005). Then, in CHES 2018 and CHES 2019, Arash and Alexander et al. presented the optimization results of the AES S-box based on the redundant normal basis (Reyhani-Masoleh et al. 2018; Maximov and Ekdahl 2019).

In order to refine the realization of S-box, there are two points that need to be focused on: optimizing the linear components and reducing the multiplicative complexity (Boyar and Peralta 2010). The optimization of linear components is just the Shortest Linear Program (SLP). In 2008, Boyar et al. proved that the SLP problem is NP-hard (Boyar et al. 2008), so optimization of the linear components is generally considered using heuristic algorithms. The two classical algorithms for solving SLP, namely Parr's algorithm (Paar 1997) and BP algorithm (Boyar and Peralta 2010; Boyar et al. 2013), are both essentially based on the greedy strategy. Reducing the multiplication complexity means minimizing the AND gates in the non-linear component, where the non-linear components, i.e., the inverters and the multipliers. The optimization of inverters can be traced back to Itoh's work (Itoh and Tsujii 1988). In 2000, Itoh et al. proposed a recursive method

*Correspondence:
Yiping Lin
linyiping2225@163.com
[1] Information Industry Information Security Evaluation Center, The 15th Research Institute of China Electronic Technology Group Corporation, Beijing 100083, China
[2] Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China
[3] Beijing Electronic Science and Technology Institute, No.7 Fufeng Road, Fengtai Distric, Beijing 100083, China
[4] College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China

for calculating the inverse in $\mathbb{F}_{(2^m)^n}$, given a circuit for calculating the inverse in $\mathbb{F}_{2^m}$. And for the optimization of multipliers, a common approach is to search for implementation using SAT solver (Stoffelen 2016). However, under the restriction of the actual calculation condition, the SAT solver can only search for the logic expressions of small-scale multipliers, and it is difficult to work out the desired results for large-scale multipliers.

At present, there are few design schemes for 16-bit S-box because the cryptographic algorithms based on 4-bit S-box or 8-bit S-box are enough to resist various attacks under traditional computational models. But with the emergence of quantum computers, the security of existing algorithms is increasingly threatened. In 2010, Hidenori et al. proposed a quantum 3-round distinguisher of Feistel construction (Kuwakado and Morii 2010) based on Simon's algorithm (Simon 1997), which reduces the time complexity of key-recovery from $O(2^n)$ to $O(n)$. Later, more and more structures were analyzed, such as Even-Mansour cipher (Kuwakado and Morii 2012), CBC-like MACs (Kaplan et al. 2016), AEZ (Shi et al. 2018), AES-COPA (Xu et al. 2021), Feistel constructions (Dong et al. 2020) et al. So, in order to resist quantum attacks, large-scale S-box with better security criteria will become a trend.

In this paper, we first classified irreducible polynomials into three classes and gave the implementation cost of the multiplication operation for each. Then we selected the lowest cost irreducible polynomials to construct the corresponding composite fields. At last, we applied the scheme to refine the MK-3 S-box. Our scheme reduced the circuit size by at least 55.9% compared with the original.

The rest of this paper is organized as follows. Section is the pre-knowledge, and in Sect. we give the optimization scheme for the multiplier over finite fields. Section is about the optimized implementation of the 16-bit S-box. The proposed scheme is compared with the original scheme in Sects. , and provides conclusions and prospects.

Pre-knowledge

MK-3 is an authenticated encryption algorithm based on a simplified duplex sponge structure (Kelly et al. 2015). The algorithm supports two versions of symmetric keys recommended by NIST. Moreover, the algorithm can be customized according to the guidelines provided by the designer (Wood et al. 2015). The customization guidelines include initial state and bijective function F. The inner architecture of the bijective function F is shown in Fig. 1.

Substitution Layer(S) It is the only nonlinear component in the F function. It is consisted of 32 16-bit S-boxes, and the 16-bit S-box is inspired by the construction of the S-box in AES (Kelly et al. 2015). First, multiplicative inversion over finite field $\mathbb{F}_{2^{16}}$ (based on irreducible polynomials with degree 16) is performed on the input bits, then for the output make affine transformation over finite field $\mathbb{F}_2$. The irreducible polynomial, namely

$$p(x) = x^{16} + x^5 + x^3 + x + 1 \tag{1}$$

is chosen for the inversion operation, and the affine transformation for the S-box is as follows:

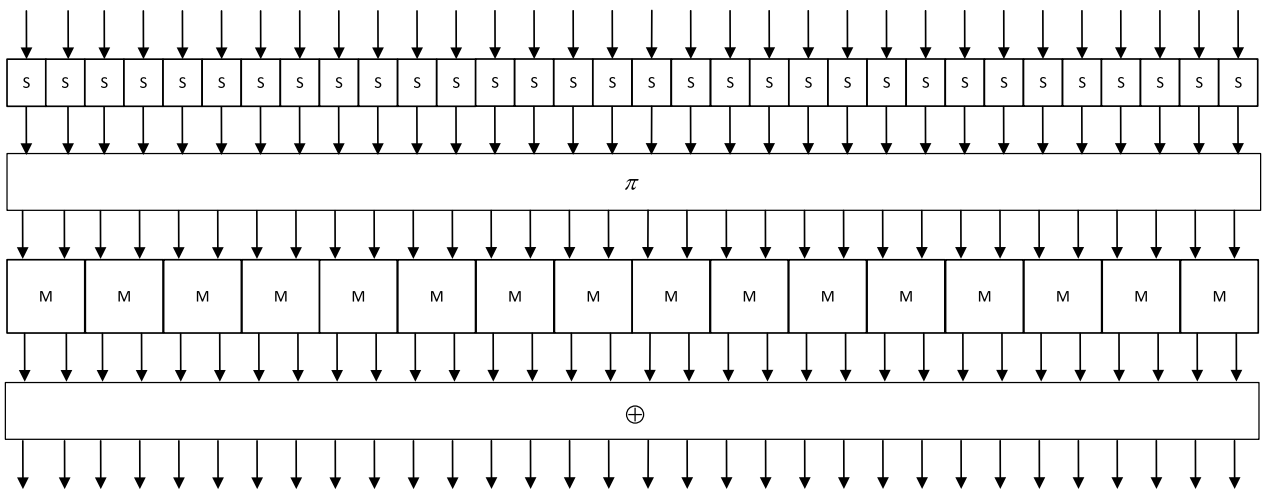

Fig. 1 internal structure of F function

$$S(x) = \begin{Bmatrix} 0&0&1&0&0&0&0&1&0&0&1&1&1&1&1&0 \\ 1&1&0&0&0&0&0&1&0&1&1&0&1&0&1&0 \\ 1&1&0&0&1&0&1&1&0&1&0&1&0&0&1&1 \\ 1&1&1&0&0&0&1&0&0&1&1&0&0&0&0&0 \\ 1&1&0&0&0&1&1&0&0&1&1&1&1&0&1&1 \\ 0&1&0&0&0&0&1&1&0&1&1&1&1&1&0&1 \\ 0&0&1&0&1&0&1&0&1&1&0&0&1&1&0&0 \\ 1&0&1&1&1&0&1&1&0&0&0&1&0&1&1&1 \\ 0&1&0&0&0&0&0&0&1&0&0&1&1&1&0&1 \\ 1&0&1&1&0&0&0&1&0&0&1&0&1&0&0&0 \\ 1&0&1&0&0&1&1&1&0&0&1&1&0&1&0&0 \\ 1&0&1&1&1&0&1&1&1&1&0&1&1&0&0&1 \\ 1&0&1&0&0&1&0&1&1&0&0&1&0&0&0&1 \\ 0&1&0&0&0&1&1&1&1&0&0&0&0&0&0&1 \\ 1&0&0&0&1&1&0&1&0&1&1&1&1&0&0&0 \\ 1&1&0&1&0&1&1&0&1&0&0&1&1&0&0&0 \end{Bmatrix} \cdot \begin{Bmatrix} x_{15} \\ x_{14} \\ x_{13} \\ x_{12} \\ x_{11} \\ x_{10} \\ x_9 \\ x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \\ x_0 \end{Bmatrix}^{-1} \oplus \begin{Bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{Bmatrix} \tag{2}$$

The S-box proposed in Wood et al. (2015) requires 1238 XOR gates and 144 AND gates for hardware implementation, which is more efficient than using look-up table since less hardware cost is needed. However, it still does not achieve the desired effect. In this paper, the isomorphism of finite fields is exploited to get a further reduction in the hardware cost of this 16-bit S-box.

The relevant algebraic theories required for this paper are introduced below.

Theorem 1 *Every element of the finite field $\mathbb{F}_{2^n}$ can be represented as a first degree polynomial $bx + c$ and multiplication is performed modulo an irreducible polynomial with degree two, denoted as $x^2 + \alpha x + \beta$, where $b, c, \alpha, \beta \in \mathbb{F}_{2^{(n/2)}}$(n is even) (Satoh et al. 2001).*

Theorem 2 *Each field $\mathbb{F}_2$ can be extended to a finite field $\mathbb{F}_{2^n}$ with an irreducible polynomial of degree n* (*Canright* 2005).

The multiplication of arbitrary polynomials modulo $x^2 + \alpha x + \beta$ is as follows:

$$(bx + c)(dx + e)/x^2 + \alpha x + \beta = (be + cd + bd\alpha)x + (bd\beta + ce)$$
$$b, c, d, e, \alpha, \beta \in \mathbb{F}_{2^{(n/2)}} \tag{3}$$

The multiplicative inverse of arbitrary polynomial $bx + c$ modulo $x^2 + \alpha x + \beta$ is as follows:

$$(bx + c)^{-1} = bhx + (c + b\alpha)h;\ h = (b^2\beta + bc\alpha + c^2)^{-1}$$
$$b, c, \alpha, \beta, h \in \mathbb{F}_{2^{(n/2)}} \tag{4}$$

The optimization scheme proposed in Satoh et al. (2001) is composed of the following four steps:

1. Select parameters: First, choose $T1(x)$ with degree n and $T2(x)$ with degree $(n/2)$ over $\mathbb{F}_2$, with which extend $\mathbb{F}_2$ to get the two finite fields, $\mathbb{F}_{2^n}$ and $\mathbb{F}_{2^{(n/2)}}$. Next, choose $T3(x)$ with degree two over $\mathbb{F}_{2^{(n/2)}}$ to extend $\mathbb{F}_{2^{(n/2)}}$ and get the composite field $\mathbb{F}_{(2^{(n/2)})^2}$, namely the quadratic extension of $\mathbb{F}_{2^{(n/2)}}$. The above construction is detailed in Table 1.
2. Find the coefficient matrix for isomorphism transformation: According to the theory of abstract algebra, There definitely exists a transformation relation between two isomorphic fields. The mapping relations between isomorphic fields can be linear or nonlinear. In terms of reducing the computational cost, searching for linear relations is helpful to our work. In the following, the linear relations will be described as the multiplication with a coefficient matrix. After defining $T1(x)$ and the parameters in Table 1, we search for the matrices that represent the isomorphisms between field $\mathbb{F}_{2^n}$ and composite field $\mathbb{F}_{(2^{(n/2)})^2}$ (Satoh et al. 2001). By using the matrix operation, the element in the field $\mathbb{F}_{2^n}$ can be mapped to that in $\mathbb{F}_{(2^{(n/2)})^2}$.

Table 1 $\mathbb{F}_{(2^{(n/2)})^2}$ construction parameters

Composite field	Composite field	Irreducible polynomial
$\mathbb{F}_{2^{(n/2)}}$	$\mathbb{F}_2[x]/T2(x)$	$T2(x)$
$\mathbb{F}_{(2^{(n/2)})^2}$	$\mathbb{F}_{2^{(n/2)}}[x]/T3(x)$	$T3(x) = x^2 + \alpha x + \beta$ $\alpha, \beta \in \mathbb{F}_{2^{(n/2)}}$

3. Compute the multiplicative inverse over the composite fields: After step 2, we can get the corresponding element over the composite field $\mathbb{F}_{(2^{(n/2)})^2}$. According to the formula (4), we can obtian the corresponding multiplicative inverse over the composite field.
4. Remap the element of composite field $\mathbb{F}_{(2^{(n/2)})^2}$ to field $\mathbb{F}_{2^n}$: Isomorphism inverse matrix is the inverse of coefficient matrix mentioned in step 2. For the output of step 3, we transform it by isomorphism inverse matrix to finally obtain the multiplicative inverse in the field $\mathbb{F}_{2^n}$.

The Optimization of Multiplier

Multiplication over finite fields is a costly operation, and its efficient hardware implementation has always been a research hotspot. In this section, we will classify multiplication operations into three classes according to the different kinds of irreducible polynomials and provide their hardware costs over the corresponding $\mathbb{F}_{2^n}$.

Type 1: The first type of irreducible polynomial denoted as $p_1(x)$ is proposed by Paar et al., and the multiplication over $\mathbb{F}_{2^n}$ extended based on such irreducible polynomials has the smallest circuit size (Paar 1996). This special irreducible polynomial $p_1(x)$ has only 3 terms, which are described as follows:

$$p_1(x) = x^n + x + 1 \tag{5}$$

Parr et al. specified that when $n = 2, 3, 4, 6, 7, 9, 10, 11, 15$ (when n is one of these numbers, the polynomials could be guaranteed to be irreducible), n^2 AND gates and $n^2 - 1$ XOR gates are needed for the hardware implementation of multiplication over $\mathbb{F}_{2^n}$. For example, when $p_1(x) = x^2 + x + 1$, the multiplication operation over $GF(2^2)$ can be described as follows:

$$\begin{aligned}
&input : a(a_{n-1}, a_{n-2}, \ldots, a_0), b(b_{n-1}, b_{n-2}, \ldots, b_0)\\
&output : c(c_{2n-2}, c_{2n-3}, \ldots, c_n, \ldots, c_0)\\
&(a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_0) \cdot (b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_0) = c\\
&c = (c_{2n-2}x^{2n-2} + c_{2n-3}x^{2n-3} + +c_0)\\
&c_{2n-2} = a_{n-1} \cdot b_{n-1}\\
&c_{2n-3} = (a_{n-1} \cdot b_{n-2}) \oplus (a_{n-2} \cdot b_{n-1})\\
&\cdots\cdots\\
&\cdots\cdots\\
&c_0 = a_0 \cdot b_0
\end{aligned}$$

From the above calculation, we can see that the construction of vector **c** requires $(n-1)^2$ XOR gates. If the modulo operation is required, vector **c** needs to be further constructed as vector **d**, and vector **d** needs $2n-2$ XOR gates:

$$\begin{aligned}
&d_{n-1} = c_{n-1} \oplus c_n;\\
&d_{n-2} = c_{n-2} \oplus c_n;\\
&d_{n-3} = c_{n-3} \oplus c_{2n-2} \oplus c_n;\\
&d_{n-4} = c_{n-4} \oplus c_{2n-3} \oplus c_n;\\
&\cdots\cdots;\\
&d_0 = c_0 \oplus c_{n+1} \oplus c_n;
\end{aligned}$$

$$\begin{aligned}
&input : a(a_1, a_0), b(b_1, b_0)\\
&output : c((a_1 \cdot b_1) \oplus (a_0 \cdot b_1) \oplus (a_1 \cdot b_0), (a_1 \cdot b_1) \oplus (a_0 \cdot b_0))
\end{aligned}$$

The operation above needs 4 AND gates and 3 XOR gates.

Type 2: $p_2(x)$ can be called all-terms irreducible polynomial, just as follows:

$$p_2(x) = x^n + x^{n-1} + \cdots + x + 1 \tag{6}$$

For multiplication modulo such polynomials, the number of AND gates is n^2, and the number of XOR gates is $n^2 - 1$. Actually, the number of AND gates required for all the multiplication operation over $\mathbb{F}_{2^n}$ is n^2, since two n-bit binary numbers are input and one bit is selected from each of the two sets of data for the multiplication operation. Details about the number of AND gates are as follows:

$$\begin{aligned}
&input : a(a_{n-1}, a_{n-2}, \ldots, a_0), b(b_{n-1}, b_{n-2}, \ldots, b_0)\\
&multiplication : \sum_{i=0}^{n-1}(a_i \cdot \sum_{j=0}^{n-1} b_j)
\end{aligned}$$

For the XOR gates, we first consider the ordinary polynomial multiplication (excluding modulo operation):

Table 2 Construction parameters of the composite fields $\mathbb{F}_{(2^8)^2}$

Composite field	The structure of the composite field	Irreducible polynomial
$\mathbb{F}_{2^8}$	$\mathbb{F}_2[x]/T2(x)$	$x^8+x^7+x^4+x^3+x^2+x+1$
$\mathbb{F}_{(2^8)^2}$	$\mathbb{F}_{2^8}[x]/T3(x)$	$x^2+00000001x+00000010$

Table 3 Construction parameters of the composite fields $\mathbb{F}_{(2^8)^2}$

Composite field	The structure of the composite field
$\mathbb{F}_{2^4}$	$\mathbb{F}_2[x]/x^4+x^3+x^2+x+1$
$\mathbb{F}_{(2^4)^2}$	$\mathbb{F}_{2^4}[x]/x^2+0001x+0010$

So the number of XOR gates required for multiplication modulo all-terms irreducible polynomials is $n^2-1=(n-1)^2+2n-2$. By now, we get the number of logic gates required for the second type of irreducible polynomial multiplication. We give following a relevant example whose multiplication operation is based on $p_2(x)=x^4+x^3+x^2+x+1$, which needs 16 AND gates and 15 XOR gates.

$$
\begin{aligned}
&input: a(a_3,a_2,a_1,a_0), b(b_3,b_2,b_1,b_0)\\
&Multiplication:\\
&(a_3\cdot b_3);(a_3\cdot b_2);(a_3\cdot b_1);(a_3\cdot b_0);(a_2\cdot b_3);(a_2\cdot b_2);(a_2\cdot b_1);(a_2\cdot b_0)\\
&(a_1\cdot b_3);(a_1\cdot b_2);(a_1\cdot b_1);(a_1\cdot b_0);(a_0\cdot b_3);(a_0\cdot b_2);(a_0\cdot b_1);(a_0\cdot b_0)\\
&Middle:\\
&c_6=(a_3\cdot b_3);\ c_5=(a_3\cdot b_2)\oplus(a_2\cdot b_3);\ c_4=(a_3\cdot b_1)\oplus(a_1\cdot b_3)\oplus(a_2\cdot b_2);\\
&c_3=(a_3\cdot b_0)\oplus(a_0\cdot b_3)\oplus(a_2\cdot b_1)\oplus(a_1\cdot b_2);\ c_2=(a_2\cdot b_0)\oplus(a_0\cdot b_2)\oplus(a_1\cdot b_1);\\
&c_1=(a_1\cdot b_0)\oplus(a_0\cdot b_1);\ c_0=(a_0\cdot b_0);\\
&output: d_3=c_3\oplus c_4;\ d_2=c_2\oplus c_4;\ d_1=c_1\oplus c_6\oplus c_4;\ d_0=c_0\oplus c_5\oplus c_4;
\end{aligned}
$$

Type 3: The irreducible polynomials that do not satisfy the definition of type 1 and type 2 are referred to as irreducible polynomials of type 3. Compared with the first two types of irreducible polynomials, multiplication based on the third always consumes more XOR gates.

Above all, this paper chose irreducible polynomials $p_1(x)$ and $p_2(x)$ to construct the composite field $\mathbb{F}_{(2^8)^2}$, The construction optimized the hardware implementation of multiplication operation over $\mathbb{F}_{(2^8)^2}$ which would be applied to the hardware optimization of the 16-bit S-Box. The details were given in "Appendix A".

S-box Optimization

This section is about the optimized circuit implementation of MK-3 S-box. The irreducible polynomial in MK-3 is $T1(x)=x^{16}+x^5+x^3+x+1$ and the corresponding finite field is $\mathbb{F}_{2^{16}}=\mathbb{F}_2[x]/T1(x)$. The composite field of $\mathbb{F}_{2^{16}}$ should be constructed first.

Firstly we choose the first irreducible polynomial $T2(x)$ with degree eight to get $\mathbb{F}_{2^8}$, which is $\mathbb{F}_2/T2(x)$. In addtion, we choose the second irreducible polynomial $T3(x)$ with degree 2, the coefficients of which are in finite field $\mathbb{F}_{2^8}$. $T3(x)$ is the irreducible polynomial with the least costly multiplication implementation. Its first-order coefficient is chosen to be one in order to save a multiplication operation with the constant. Based on $T3(x)$, we extend $\mathbb{F}_{2^8}$ to get the composite field $\mathbb{F}_{2^8}[x]/T3(x)$, denoted as $\mathbb{F}_{(2^8)^2}$. The above parameters are given in Table 2.

After the above parameters are determined, we search for the linear map (coefficient matrix) between the finite field $\mathbb{F}_{2^{16}}$ and the composite field $\mathbb{F}_{(2^8)^2}$ where there are 16 kinds of mapping relationships. Since this S-box is constructed by affine transformation, as in Rijmen (2000), the coefficient matrix of the affine transformation and the coefficient matrix of isomorphism can be merged in order to achieve a reduction in hardware cost. Therefore, the optimization results presented later only include the merged matrix and the inverse of the coefficient matrix of isomorphism.

The following components are needed for optimization of the MK-3 S-box:

Component 1: The compound linear transformation. The composite coefficient matrix is denoted as $R=M\times T$, where T is the coefficient matrix of isomorphism between $\mathbb{F}_{2^{16}}$ and $\mathbb{F}_{(2^8)^2}$ and M is the coefficient matrix of affine transformation.

Component 2: Multiplier. It is for the multiplication over $\mathbb{F}_{2^8}$.

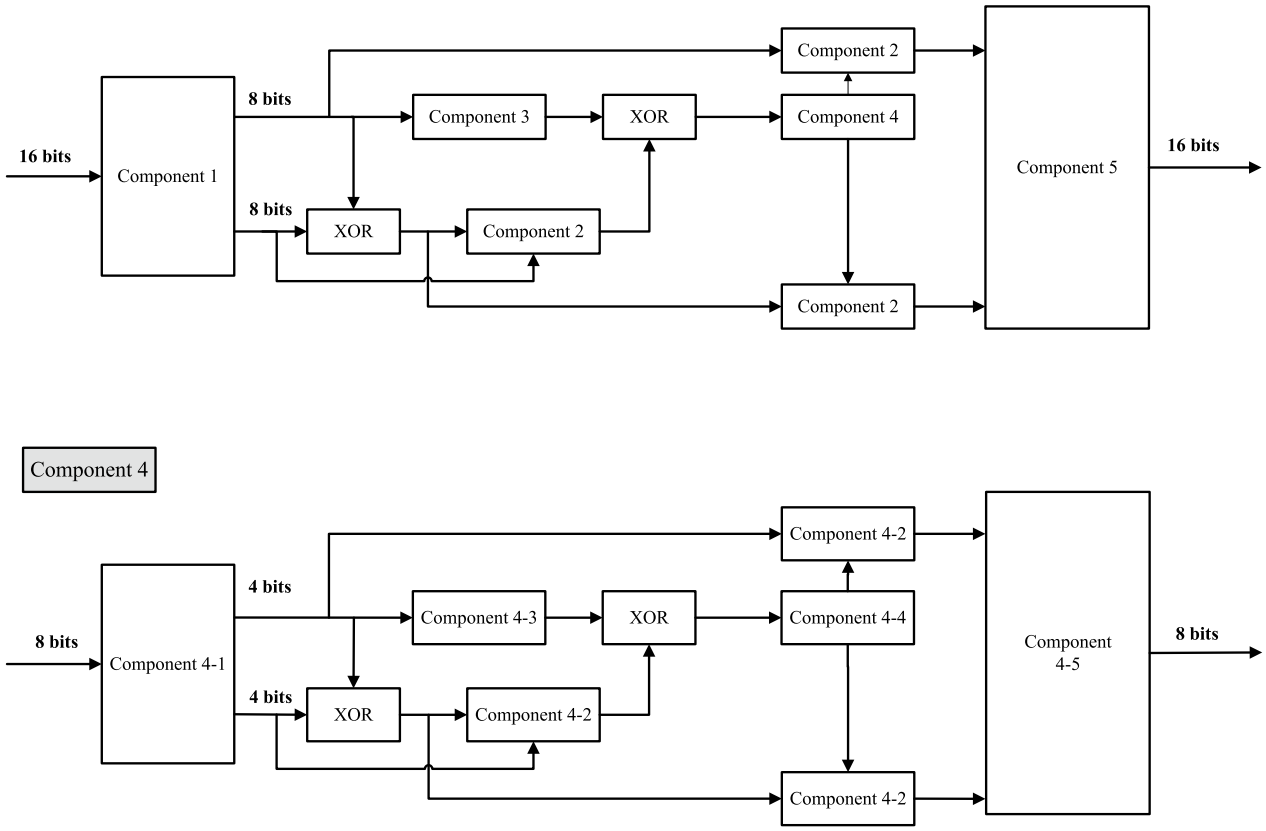

Fig. 2 16-bit S-box design process

Table 4 Area of frequently-used logic gates in three different manufacturing processes

Logic gate	NOT	AND	NAND	OR	NOR	XOR/XNOR
SMIC 130nm	0.67	1.33	1.00	1.33	1.00	2.33
SMIC 65nm	0.75	1.2	1	1.5	1	2.25
Nangate 45nm	0.67	1.33	1	1.33	1	2

Table 5 Comparison of implementation schemes of the MK-3 S-box

Logic gate	NOT	AND	XOR	XNOR	OR	SMIC130nm	SMIC65nm	Nangate 45nm
Reference (Wood et al. 2015)	0	144	1238	0	0	3076.06	3001.50	2667.52
This paper	0	245	411	9	6	1312.43	1321.50	1173.83

Component 3: Square-scale operation. It is for the combined operation of squaring then scaling by a constant over $\mathbb{F}_{2^8}$.

Component 4: Multiplicative inverter. It is for calculating the inverse of multiplication over $\mathbb{F}_{2^8}$. Since it is an inversion operation in finite fields, we can also convert the element in $\mathbb{F}_{2^8}$ to its composite fields representation in $\mathbb{F}_{(2^4)^2}$.

Component 5: The isomorphism transformation from $\mathbb{F}_{2^{16}}$ to $\mathbb{F}_{(2^8)^2}$. The coefficient matrix of it is denoted as T^{-1}. For component 4, the parameters for the conversion from finite fields to composite fields are shown in Table 3.

$$T: \begin{pmatrix} 1&0&0&1&1&0&0&1&1&1&0&0&0&1&1&0\\ 0&1&0&0&1&1&1&1&0&0&1&0&1&1&0&0\\ 1&1&0&1&1&0&0&0&0&0&1&1&0&1&0&0\\ 0&0&0&0&1&1&0&0&1&0&1&0&1&0&0&0\\ 1&1&0&0&0&1&0&1&1&0&1&0&1&0&1&0\\ 1&1&0&1&1&1&1&1&0&1&1&1&1&0&1&0\\ 1&0&1&1&0&0&0&0&1&0&1&0&0&1&1&0\\ 1&1&1&1&1&0&0&1&1&0&0&0&1&0&1&0\\ 1&0&0&1&0&1&1&0&0&1&0&1&0&1&0&0\\ 1&0&0&1&1&0&0&1&1&0&1&1&0&0&0&0\\ 1&0&0&1&0&1&1&1&1&0&0&1&1&0&1&0\\ 0&1&0&0&1&1&0&0&0&1&0&0&1&0&0&0\\ 1&1&0&1&0&0&1&0&1&0&1&1&1&0&0&0\\ 1&1&0&0&1&0&0&0&1&1&0&0&0&1&0&0\\ 0&0&0&0&0&1&1&0&0&0&0&1&0&1&0&0\\ 1&0&0&1&1&0&1&1&1&0&0&0&1&1&1&1 \end{pmatrix} \quad T^{-1}: \begin{pmatrix} 0&1&1&1&0&1&1&1&0&0&1&0&1&1&0&0\\ 1&0&1&1&0&0&1&1&0&1&1&1&0&0&1&0\\ 1&0&0&1&1&1&0&1&0&0&0&1&1&1&0&0\\ 1&1&1&1&1&0&1&1&0&0&0&1&1&0&0&0\\ 0&0&0&1&0&1&0&0&0&0&1&1&0&0&0&0\\ 0&0&1&1&0&0&1&1&0&1&0&0&0&0&0&0\\ 0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0\\ 1&1&0&0&1&1&1&1&1&1&1&0&1&1&1&0\\ 0&1&1&0&0&1&1&1&1&0&1&0&0&0&0&0\\ 1&0&0&0&1&1&0&0&1&0&1&1&0&1&1&0\\ 0&1&0&0&1&0&0&0&0&1&1&1&0&1&0&0\\ 0&1&1&1&1&0&0&0&0&1&1&1&1&1&1&0\\ 0&0&0&1&1&0&0&0&1&0&1&0&0&1&0&0\\ 0&0&1&1&1&0&1&1&0&1&1&1&1&1&0&0\\ 0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0\\ 0&1&1&0&0&1&0&0&0&1&0&1&0&0&1&1 \end{pmatrix}$$

The component 4 can be divided into another 5 parts as follows.

Component 4-1: The coefficient matrix of isomorphism from $\mathbb{F}_{2^8}$ to $\mathbb{F}_{(2^4)^2}$ denoted as δ.

Component 4-2: Multiplication over $\mathbb{F}_{2^4}$.

Component 4-3: Square-scale operation. There are square operation and scaling by the constant 2 over $\mathbb{F}_{2^4}$.

Component 4-4: Inverse of multiplication over $\mathbb{F}_{2^4}$. This section uses SAT solver to search for results.

Component 4-5: The isomorphism transformation from $\mathbb{F}_{(2^4)^2}$ to $\mathbb{F}_{2^8}$. The coefficient matrix is denoted as δ^{-1}.

$$\delta = \begin{pmatrix} 0&0&1&0&0&0&1&0\\ 0&1&0&1&0&0&0&0\\ 1&0&0&0&1&1&0&0\\ 1&0&1&1&0&1&1&0\\ 1&0&0&0&1&1&1&0\\ 0&0&1&0&1&1&1&0\\ 1&0&0&1&1&1&1&0\\ 0&0&0&0&1&1&0&1 \end{pmatrix} \quad \delta^{-1} = \begin{pmatrix} 1&0&1&0&0&1&0&0\\ 0&1&0&0&1&0&1&0\\ 1&0&1&0&1&0&0&0\\ 0&0&0&0&1&0&1&0\\ 1&0&1&1&1&0&1&0\\ 0&0&1&1&1&1&1&0\\ 0&0&1&0&1&0&0&0\\ 1&0&0&0&0&1&0&1 \end{pmatrix}$$

The components shown in Fig. 2 are described in the following:

$$component1:$$

$$\left\{x:\{0,1\}^{16}\rightarrow y:\{0,1\}^{16}\right\}$$

$$m_0=x_{15}\oplus x_{14};\ m_1=x_6\oplus x_1;\ m_2=m_0\oplus m_1;\ m_3=x_{13}\oplus x_{12};$$
$$m_4=m_1\oplus m_3;\ m_5=m_2\oplus m_3;\ m_6=x_{10}\oplus x_7;\ m_7=m_6\oplus m_0;$$
$$m_8=x_{12}\oplus x_0;\ m_9=x_9\oplus x_2;\ m_{10}=m_2\oplus m_9;\ m_{11}=x_4\oplus x_5;$$
$$m_{12}=x_{15}\oplus x_{11};\ m_{13}=m_3\oplus m_{12};\ m_{14}=x_3\oplus x_1;\ m_{15}=x_4\oplus x_0;$$
$$m_{16}=m_5\oplus x_8;\ m_{17}=m_{11}\oplus x_{10};\ m_{18}=m_{10}\oplus x_{11};\ m_{19}=m_4\oplus x_{15};$$
$$y_{15}=m_0\oplus m_9\oplus x_{10}\oplus x_{13};\ y_{14}=m_6\oplus m_{16}\oplus x_2\oplus x_5;$$
$$y_{13}=m_8\oplus m_9\oplus m_{14};\ y_{12}=m_{19};\ y_{11}=m_{17}\oplus m_{19}\oplus x_0\oplus x_2;$$
$$y_{10}=m_7\oplus m_{15}\oplus x_8;\ y_9=m_{13}\oplus x_5\oplus x_7;\ y_7=m_1\oplus m_6\oplus m_8;$$
$$y_8=m_2\oplus m_8\oplus x_3\oplus x_7\oplus x_{11};\ y_6=m_{11}\oplus m_{16}\oplus x_7;$$
$$y_5=m_7\oplus m_{11}\oplus x_1;\ y_4=m_{18}\oplus x_0\oplus x_8;\ y_3=m_6\oplus m_{15}\oplus x_9\oplus x_{13};$$
$$y_2=m_7\oplus m_8\oplus x_5;\ y_1=m_{18}\oplus x_{13};\ y_0=m_{13}\oplus m_{14}\oplus m_{17}\oplus x_8;$$

$$component2:$$

$$\mathbb{F}_{2^8}/x^8+x^7+x^6+x^5+x^4+x^3+1:\left(\left\{a:\{0,1\}^8\times b:\{0,1\}^8\rightarrow d:\{0,1\}^8\right\}\right)$$

$$\left(a_7x^7+a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x+a_0\right)\cdot$$
$$\left(b_7x^7+b_6x^6+b_5x^5+b_4x^4+b_3x^3+b_2x^2+b_1x+b_0\right)=c$$

$$c=\left(c_{14}x^{14}+c_{13}x^{13}+\cdots\cdots+c_0\right)$$

$$c_{14}=a_7\cdot b_7;$$
$$c_{13}=(a_7\cdot b_6)\oplus(a_6\cdot b_7);$$
$$c_{12}=(a_7\cdot b_5)\oplus(a_5\cdot b_7)\oplus(a_6\cdot b_6);$$
$$c_{11}=(a_7\cdot b_4)\oplus(a_4\cdot b_7)\oplus(a_6\cdot b_5)\oplus(a_5\cdot b_6);$$
$$c_{10}=(a_7\cdot b_3)\oplus(a_3\cdot b_7)\oplus(a_6\cdot b_4)\oplus(a_4\cdot b_6)\oplus(a_5\cdot b_5);$$
$$c_9=(a_7\cdot b_2)\oplus(a_2\cdot b_7)\oplus(a_6\cdot b_3)\oplus(a_3\cdot b_6)\oplus(a_5\cdot b_4)\oplus(a_4\cdot b_5);$$
$$c_8=(a_7\cdot b_1)\oplus(a_1\cdot b_7)\oplus(a_6\cdot b_2)\oplus(a_2\cdot b_6)\oplus(a_5\cdot b_3)\oplus(a_3\cdot b_5)\oplus(a_4\cdot b_4);$$
$$c_7=(a_7\cdot b_0)\oplus(a_0\cdot b_7)\oplus(a_6\cdot b_1)\oplus(a_1\cdot b_6)\oplus(a_5\cdot b_2)\oplus(a_2\cdot b_5)\oplus(a_4\cdot b_3)\oplus(a_3\cdot b_4);$$
$$c_6=(a_6\cdot b_0)\oplus(a_0\cdot b_6)\oplus(a_5\cdot b_1)\oplus(a_1\cdot b_5)\oplus(a_4\cdot b_2)\oplus(a_2\cdot b_4)\oplus(a_3\cdot b_3);$$
$$c_5=(a_5\cdot b_0)\oplus(a_0\cdot b_5)\oplus(a_4\cdot b_1)\oplus(a_1\cdot b_4)\oplus(a_3\cdot b_2)\oplus(a_2\cdot b_3);$$
$$c_4=(a_4\cdot b_0)\oplus(a_0\cdot b_4)\oplus(a_3\cdot b_1)\oplus(a_1\cdot b_3)\oplus(a_2\cdot b_2);$$
$$c_3=(a_3\cdot b_0)\oplus(a_0\cdot b_3)\oplus(a_2\cdot b_1)\oplus(a_1\cdot b_2);$$
$$c_2=(a_2\cdot b_0)\oplus(a_0\cdot b_2)\oplus(a_1\cdot b_1);$$
$$c_1=(a_1\cdot b_0)\oplus(a_0\cdot b_1);$$
$$c_0=(a_0\cdot b_0);$$

$m_0 = c_{12} \oplus c_{10}$; $m_1 = c_{14} \oplus c_8$; $m_2 = c_{13} \oplus c_{10}$; $m_3 = c_{11} \oplus c_9$; $m_4 = m_1 \oplus m_2$;
$d_7 = c_7 \oplus m_4 \oplus c_9$; $d_6 = c_6 \oplus m_0$; $d_5 = c_5 \oplus m_3 \oplus c_{14}$; $d_4 = c_4 \oplus m_4$;
$d_3 = c_3 \oplus m_0 \oplus c_8$; $d_2 = c_2 \oplus m_4 \oplus c_{11}$; $d_1 = c_1 \oplus m_1 \oplus c_{12}$; $d_0 = c_0 \oplus m_1 \oplus m_3 \oplus c_{10}$;
$component3$:

$$\left\{x : \{0,1\}^8 \to y : \{0,1\}^8\right\}$$

$m_0 = x_4 \oplus x_7$; $m_1 = x_3 \oplus x_6$; $m_2 = x_2 \oplus x_5$; $m_3 = x_6 \oplus x_7$; $m_4 = x_5 \oplus x_6$;
$y_7 = m_0 \oplus m_1$; $y_6 = x_7$; $y_5 = m_0 \oplus m_2$; $y_4 = m_3$; $y_3 = x_1$; $y_2 = m_4$; $y_1 = x_0$; $y_0 = m_0 \oplus x_5$;
$component4-1$:

$$\left\{x : \{0,1\}^8 \to y : \{0,1\}^8\right\}$$

$m_0 = x_3 \oplus x_2$; $m_1 = x_5 \oplus x_1$; $m_2 = x_7 \oplus x_1$;
$y_7 = m_1$; $y_6 = x_4 \oplus x_6$; $y_5 = m_0 \oplus x_7$; $y_4 = m_2 \oplus x_2 \oplus x_4 \oplus x_5$; $y_3 = m_2 \oplus m_0$;
$y_2 = m_1 \oplus m_0$; $y_1 = y_3 \oplus x_4$; $y_0 = m_0 \oplus x_0$;
$component4-2$:

$$\left\{a : \{0,1\}^4 \times b : \{0,1\}^4 \to d : \{0,1\}^4\right\}$$

$$\begin{cases} (a_3 \cdot b_3);\ (a_3 \cdot b_2);\ (a_3 \cdot b_1);\ (a_3 \cdot b_0) \\ (a_2 \cdot b_3);\ (a_2 \cdot b_2);\ (a_2 \cdot b_1);\ (a_2 \cdot b_0) \\ (a_1 \cdot b_3);\ (a_1 \cdot b_2);\ (a_1 \cdot b_1);\ (a_1 \cdot b_0) \\ (a_0 \cdot b_3);\ (a_0 \cdot b_2);\ (a_0 \cdot b_1);\ (a_0 \cdot b_0) \end{cases}$$

$c_6 = (a_3 \cdot b_3)$; $c_5 = (a_3 \cdot b_2) \oplus (a_2 \cdot b_3)$; $c_4 = (a_3 \cdot b_1) \oplus (a_1 \cdot b_3) \oplus (a_2 \cdot b_2)$;
$c_3 = (a_3 \cdot b_0) \oplus (a_0 \cdot b_3) \oplus (a_2 \cdot b_1) \oplus (a_1 \cdot b_2)$; $c_2 = (a_2 \cdot b_0) \oplus (a_0 \cdot b_2) \oplus (a_1 \cdot b_1)$;
$c_1 = (a_1 \cdot b_0) \oplus (a_0 \cdot b_1)$; $c_0 = (a_0 \cdot b_0)$;
$d_3 = c_3 \oplus c_4$; $d_2 = c_2 \oplus c_4$; $d_1 = c_1 \oplus c_6 \oplus c_4$; $d_0 = c_0 \oplus c_5 \oplus c_4$;
$component4-3$:

$$\left\{x : \{0,1\}^4 \to y : \{0,1\}^4\right\}$$

$y_3 = x_1$; $y_2 = x_3$; $y_1 = x_0$; $y_0 = x_2$;
$component4-4$:

$$\left\{x : \{0,1\}^4 \to y : \{0,1\}^4\right\}$$

$m_0 = x_3 \oplus x_2$; $m_1 = m_0 + x_1$; $m_2 = x_3 \cdot x_1$; $m_3 = m_1 \cdot x_0$; $m_4 = x_2 \oplus m_3$;
$m_5 = m_3 \cdot m_4$; $m_6 = x_2 \cdot m_1$; $m_7 = m_3 + m_0$; $m_8 = x_1 \oplus m_4$; $m_9 = m_4 + x_3$;
$m_{10} = m_2 + m_5$; $m_{11} = m_3 \oplus m_5$; $m_{12} = m_8 \cdot m_9$; $m_{13} = m_2 + x_0$;
${x_0}^{-1} = m_8 \oplus m_{10}$; ${x_1}^{-1} = m_6 \oplus m_{13}$; ${x_2}^{-1} = m_{11} + m_{12}$; ${x_3}^{-1} = m_7 \oplus m_2$;

$component4-5$:

$$\left\{x : \{0,1\}^8 \to y : \{0,1\}^8\right\}$$

$m_0 = x_3 \oplus x_1$; $m_1 = x_7 \oplus x_2$; $m_2 = x_5 \oplus x_3$; $m_3 = x_5 \oplus x_4$; $m_4 = m_0 \oplus m_3$;
$y_7 = x_5 \oplus m_1$; $y_6 = m_0 \oplus x_6$; $y_5 = m_2 \oplus x_7$; $y_4 = m_0$; $y_3 = x_7 \oplus m_4$;
$y_2 = x_2 \oplus m_4$; $y_1 = m_2$; $y_0 = x_0 \oplus m_1$;

$component5:$

$\left\{x:\{0,1\}^{16}\rightarrow y:\{0,1\}^{16}\right\}$

$m_0=x_9\oplus x_8;\ m_1=x_{13}\oplus x_{12};\ m_2=m_0\oplus m_1;\ m_3=x_5\oplus x_2;$

$m_4=m_2\oplus m_3;\ m_5=x_{11}\oplus x_4;\ m_6=m_3\oplus m_5;\ m_7=x_{14}\oplus x_6;$

$m_8=m_6\oplus m_7;\ m_9=x_{10}\oplus x_7;\ m_{10}=m_0\oplus m_9;\ m_{11}=x_{15}\oplus x_3;$

$m_{12}=m_5\oplus m_{11};\ m_{13}=x_{10}\oplus x_{12};\ m_{14}=x_5\oplus x_4;\ m_{15}=x_{15}\oplus x_1;$

$m_{16}=m_7\oplus x_1;\ m_{17}=m_4\oplus x_3;\ m_{18}=m_2\oplus x_6;\ m_{19}=m_3\oplus x_{11};$

$y_{15}=m_{17}\oplus x_{10}\oplus x_{14};\ y_{14}=m_{14}\oplus m_{15}\oplus m_{18};$

$y_{13}=m_{12}\oplus m_{13}\oplus x_8\oplus x_2;\ y_{12}=m_2\oplus m_{12}\oplus x_{14};\ y_{11}=m_{13}\oplus m_{14};$

$y_{10}=m_{18};\ y_9=m_1\oplus m_7;\ y_8=m_{10}\oplus m_{11}\oplus m_{16}\oplus m_{19};$

$y_7=m_{10}\oplus x_5\oplus x_{13}\oplus x_{14};\ y_6=m_6\oplus m_9\oplus m_{15};\ y_5=m_8;$

$y_4=m_1\oplus m_8\oplus x_1\oplus x_3;\ y_3=m_{19}\oplus x_7\oplus x_{12};\ y_2=m_5\oplus m_{17}\oplus x_6;$

$y_1=m_{10};\ y_0=m_{16}\oplus x_0\oplus x_4\oplus x_{10}\oplus x_{13};$

Comparison

The hardware cost of S-box is not only related to the number of logic gates but also related to the manufacturing process of logic gates. Circuit area of logic gates in different manufacturing processes is listed in Table 4. Compared with the cost of the MK-3 S-box in previous literature (Wood et al. 2015), our results proposed in this paper can reduce the circuit size by at least 55.9%, which is shown in Table 5.

Conclusion

In this paper, we first focused on the hardware implementation of multiplication over finite fields based on three classes of irreducible polynomials. On this basis, the irreducible polynomial with the lowest cost of multiplication implementation was used to construct the composite fields, $\mathbb{F}_{(2^4)^2}$ and $\mathbb{F}_{(2^8)^2}$. The SAT solver was further used for the optimization of the small-scale nonlinear component. At last, the hardware implementation costs of 16 groups of isomorphism mappings were compared to obtain the least costly circuit implementation scheme for the MK-3 S-box.

Compared with the scheme proposed by the original designer, the results of the implementation proposed in this paper can reduce the hardware resource occupied circuit area by at least 55.9%. In the future, we are committed to discovering a better optimized S-box implementation scheme, considering the circuit depth and circuit area together.

Appendix A Number of gates for multiplication over finite field $\mathbb{F}_{2^8}$

Irreducible polynomial	Number of XOR gates	Number of AND gates	Irreducible polynomial	Number of XOR gates	Number of AND gates
$x^8+x^7+x^5+x^4+1$	71	64	$x^8+x^4+x^3+x+1$	72	64
$x^8+x^6+x^5+x^4+1$	73	64	$x^8+x^7+x^2+x+1$	71	64
$x^8+x^7+x^5+x^3+1$	71	64	$x^8+x^7+x^6+x^5+x^4+x^3+1$	69	64
$x^8+x^6+x^5+x^3+1$	74	64	$x^8+x^7+x^6+x^5+x^4+x^2+1$	69	64
$x^8+x^5+x^4+x^3+1$	73	64	$x^8+x^7+x^6+x^4+x^3+x^2+1$	71	64
$x^8+x^6+x^5+x^2+1$	73	64	$x^8+x^7+x^5+x^4+x^3+x^2+1$	72	64
$x^8+x^7+x^3+x^2+1$	70	64	$x^8+x^7+x^6+x^5+x^4+x+1$	69	64
$x^8+x^6+x^3+x^2+1$	73	64	$x^8+x^6+x^5+x^4+x^3+x+1$	74	64
$x^8+x^5+x^3+x^2+1$	71	64	$x^8+x^7+x^6+x^5+x^2+x+1$	70	64
$x^8+x^4+x^3+x^2+1$	72	64	$x^8+x^7+x^6+x^4+x^2+x+1$	73	64
$x^8+x^7+x^6+x+1$	70	64	$x^8+x^6+x^5+x^4+x^2+x+1$	69	64
$x^8+x^6+x^5+x+1$	73	64	$x^8+x^7+x^6+x^3+x^2+x+1$	72	64

Irreducible polynomial	Number of XOR gates	Number of AND gates	Irreducible polynomial	Number of XOR gates	Number of AND gates
$x^8+x^7+x^5+x+1$	71	64	$x^8+x^7+x^4+x^3+x^2+x+1$	69	64
$x^8+x^7+x^3+x+1$	72	64	$x^8+x^6+x^4+x^3+x^2+x+1$	69	64
$x^8+x^5+x^3+x+1$	71	64	$x^8+x^5+x^4+x^3+x^2+x+1$	74	64

Authors' contributions
The author(s) read and approved the final manuscript.

Funding
This work is supported by the Open Project of Henan Key Laboratory of Network Cryptography Technology (NO. LNCT2021-A09), and the Advanced Discipline Construction Project of Beijing Universities (20210101Z0401).

Availability of data and materials
The datasets generated during analysed during the current study are not publicly available but are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 25 October 2023 Accepted: 15 January 2024
Published online: 03 February 2024

References

Boyar J, Matthews P, Peralta R (2008) On the shortest linear straight-line program for computing linear forms. Math Found Comput Sci 2008:168–179
Boyar Joan, Matthews Philip, Peralta René (2013) Logic minimization techniques with applications to cryptology. J Cryptol 26:280–312
Boyar J, Peralta R (2010) A new combinational logic minimization technique with applications to cryptology. Exp Algorithm, pp 178–189
Canright D (2005) A very compact S-Box for AES. Cryptographic hardware and embedded systems—CHES 2005, pp 441–455
Daemen J, Rijmen V (1998) The block Cipher Rijndael. Smart card research and advanced application conference
Dong X, Dong B, Wang X (2020) Quantum attacks on some feistel block Ciphers. Des Codes Crypt 88:1179–1203
Itoh T, Tsujii S (1988) A fast algorithm for computing multiplicative inverses in GF (2m) using normal bases. Inf Comput 78:171–177
Kaplan M, Leurent G, Leverrier A, Naya-Plasencia M (2016) Breaking symmetric cryptosystems using quantum period finding. Adv Cryptol 2016:207–237
Kelly M, Kaminsky A, Kurdziel MT, Lukowiak M, Radziszowski SP (2015) Customizable sponge-based authenticated encryption using 16-bit s-boxes. MILCOM 2015–2015 IEEE military communications conference, pp 43–48
Kuwakado H, Morii M (2010) Quantum distinguisher between the 3-round Feistel cipher and the Random Permutation. 2010 IEEE international symposium on information theory, pp 2682–2685
Kuwakado H, Morii M (2012) Security on the quantum-type even-Mansour Cipher. 2012 international symposium on information theory and its applications, pp 312–316
Maximov A, Ekdahl P (2019) New circuit minimization techniques for smaller and faster AES SBoxes. IACR Trans Cryptograph Hardware Embedded Syst, pp 91–125
NIST A (2001) Specification of the advanced encryption standard (AES). Federal information processing standards publication **197**
Paar C (1995) Some remarks on efficient inversion in finite fields. In: Proceedings of 1995 IEEE international symposium on information theory vol 58
Paar C (1997) Optimized arithmetic for reed-solomon encoders. Proceedings of IEEE international symposium on information theory, vol 250
Paar C (1996) A new architecture for a parallel finite field multiplier with low complexity based on composite fields. IEEE Trans Comput 45:856–861
Reyhani-Masoleh A, Taha M, Ashmawy D (2018) Smashing the implementation records of AES S-box. IACR transactions on cryptographic hardware and embedded systems, pp 298–336
Rijmen V (2000) Efficient implementation of the Rijndael S-box. Katholieke Universiteit Leuven, Dept. ESAT. Belgium
Satoh A, Morioka S, Takano K, Munetoh S (2001) A compact Rijndael hardware architecture with S-box optimization. Advances in cryptology-ASIACRYPT 2001: Proceedings 7th international conference on the theory and application of cryptology and information security gold coast, Australia, December 9–13, pp 239–254
Shi T, Jin C, Guan J (2018) Collision attacks against AEZ-PRF for authenticated encryption AEZ. China Commun 15:46–53
Simon DR (1997) On the power of quantum computation. SIAM J Comput 26:1474–1483
Stoffelen K (2016) Optimizing S-box implementations for several criteria using SAT solvers. In: Fast software encryption: 23rd international conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers 23, pp 140–160
Wood CA, Radziszowski SP, Lukowiak M (2015) Constructing large S-boxes with area minimized implementations. MILCOM 2015–2015 IEEE military communications conference, pp 49–54
Xu Y, Liu W, Yu W (2021) Quantum forgery attacks on COPA, AES-COPA and marble authenticated encryption algorithms. Quant Inf Process 20:1–21

Publisher's Note