IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 14, 2021, accepted January 10, 2022, date of publication January 18, 2022, date of current version January 31, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3144458

New Genetic Operators for Developing S-Boxes With Low Boomerang Uniformity

MAN KANG AND MINGSHENG WANG
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Man Kang (kangman@iie.ac.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61772516 and Grant 61772517.

ABSTRACT The boomerang uniformity measures the resistance of block ciphers to boomerang attacks and has become an essential criterion of the substitution box (S-box). However, the S-boxes created by the Feistel structure have a poor property of boomerang uniformity. The genetic algorithm is introduced to improve the properties of the S-boxes created by the Feistel structure. New genetic operators are designed for the genetic algorithm to improve its searchability. The new genetic algorithm generates some 8×8 bijective S-boxes with differential uniformity 6, nonlinearity 108, and boomerang uniformity 10, which has dramatically improved the properties of the S-boxes created by the Feistel structure. Furthermore, the new genetic algorithm also improves the properties of the S-box population created by the Feistel structure as a whole. We compare the S-boxes generated by the new genetic algorithm with those generated by the traditional one. The comparison results show that the S-boxes generated by the new genetic algorithm have better properties than the S-boxes generated by the traditional genetic algorithm, demonstrating the new genetic algorithm's effectiveness and superiority in developing S-boxes.

INDEX TERMS Boomerang uniformity, S-box, genetic algorithm, genetic operator.

I. INTRODUCTION

The boomerang attack [1] is a variant of the differential attack. For ciphers that the probabilities of the differential characteristics decrease exponentially with respect to the growth of rounds, the boomerang attack can concatenate two short characteristics to form a longer characteristic with a better probability. In boomerang attack, two short parts E_0 and E_1 make up a larger characteristic E. Assume that p is the probability of the differential characteristic (α, β) for E_0, and q is the probability of the differential characteristic (γ, δ) for E_1. Then the probability of the boomerang distinguisher is

$$Pr\left[E^{-1}\left(E(x) \oplus \delta\right) \oplus E^{-1}\left(E\left(x \oplus \alpha\right) \oplus \delta\right) = \alpha\right] = p^2q^2.$$

The boomerang attack is an effective cryptanalysis tool, which has been successfully applied to famous block ciphers such as AES, IDEA and SHACAL1 [2]–[5].

Boomerang connectivity table (BCT) [6] provides a unified representation for boomerang-style attacks, which has become a new tool of substitution boxes (S-boxes) for more accurately evaluating the probability of generating a right quartet in boomerang-style attacks. The boomerang uniformity [7] is the maximum value in BCT among all nonzero input differences and output differences that measures the resistance of an S-box to a boomerang attack.

S-boxes are crucial nonlinear building blocks providing confusion in modern block ciphers. The emergence of cryptographic attacks has led to the development of criteria for resisting such attacks. Existing attacks require S-boxes to meet cryptographic properties, including bijectivity, low differential uniformity [8], and high nonlinearity [9]. With the development of boomerang attacks, boomerang uniformity has become a new essential criterion for the S-box, which has attracted the interest of researchers.

Boura and Canteaut [7] completely characterized the BCT of all differentially 4-uniform permutations of 4 bits and then studied these objects for inverse functions and quadratic permutations. Their work provided the first examples of differentially 4-uniform S-boxes optimal against boomerang attacks for an even number of variables. The boomerang uniformities of some specific permutations were studied in [10]

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Tedesco.

and a class of 4-uniform BCT permutations over $\mathbb{F}_{2^n}$ were obtained. Mesnager *et al.* [11] focused their research on the boomerang uniformity of quadratic permutations in even dimensions. A new class of optimal S-boxes was found by generalizing previous results on quadratic permutations with optimal BCT. Calderini and Villa [12] further studied the boomerang uniformity of some non-quadratic differentially 4-uniform functions. Wang *et al.* [13] studied the boomerang uniformity of all normalized permutation polynomials of degree up to six over the arbitrary finite field $\mathbb{F}_q$ by using the resultant elimination method. Li *et al.* [14] presented infinite families of permutations of $\mathbb{F}_{2^{2n}}$ for a positive odd integer n, which have the best-known nonlinearity and boomerang uniformity 4.

In addition to mathematical methods, intelligent methods have also been used to create S-boxes in recent years. Reinforcement learning was used to train a method expressed in the Markov decision process to an agent to generate S-boxes that can effectively resist the side-channel attack [15]. Heuristic evolution strategy improved the initial S-Boxes created by a modular operation [16]. The S-box construction time was reduced by constrainedly maximizing the nonlinearity of the S-boxes created by a random-restart hill-climbing algorithm [17]. The S-boxes based on chaos were designed in [18]. The combination of the chaos method and intelligent algorithm was also used to generate S-boxes. An artificial bee colony algorithm was used to optimize the S-boxes generated by chaotic sequence [19]. A β-hill climbing search was applied to improve the S-boxes based on chaotic map [20]. As an intelligent algorithm simulating the evolution of nature, the genetic algorithm provides a practical solution to the combinatorial optimization problem that is difficult to deal with by traditional methods and provides a new idea and means for the complex problems in cryptography.

In recent years, genetic algorithms have been increasingly used to generate S-boxes with good performances. The traditional genetic algorithm was used to generate S-boxes with good values of the confusion coefficient in terms of improving their side-channel resistance [21]. A method based on chaos and the genetic algorithm was proposed by [22] for designing an S-box. The full use of the traits of chaotic map and evolution process makes it possible to obtain a stronger S-box. A genetic algorithm working in a reversed way was proposed by [23], which can rapidly and repeatedly generate a large number of strong bijective S-boxes. Several genetic algorithms and problem sizes were explored by [24] to find functions having differential uniformity equal to 6. In addition, simulated annealing and genetic algorithm were used to optimize the design of symmetric-key primitives in [24].

A. OUR CONTRIBUTIONS

S-boxes constructed by the Feistel structure have the advantage of low hardware implementation cost [26]; however, they have high boomerang uniformities. In this paper, a new genetic algorithm is introduced to improve the properties of the S-boxes created by the Feistel structure. A new crossover operator and a new mutation operator are proposed to improve the performance of the genetic algorithm. The new genetic algorithm generates 8×8 bijective S-boxes with low differential uniformity, high nonlinearity, and low boomerang uniformity. It is the first time that a meta-heuristic algorithm has been used to search for S-boxes with low boomerang uniformity. Benefiting from the full use of the advantages of gene exchange and gene mutation, the new genetic algorithm in this paper dramatically improves the properties of the S-boxes created by the Feistel structure. The new genetic algorithm generates the best S-box with differential uniformity 6, nonlinearity 108, and boomerang uniformity 10, whereas the Feistel structure creates the best S-box with differential uniformity 16, nonlinearity 96, and boomerang uniformity 52. Furthermore, the new genetic algorithm improves the properties of the population created by the Feistel structure. In addition, we compare the S-boxes generated by the new genetic algorithm and the S-boxes generated by the traditional genetic algorithm. The comparison results show that the S-boxes generated by our new genetic algorithm have better properties than those generated by the traditional genetic algorithm. The experimental results show the effectiveness and superiority of our new genetic algorithm.

B. OUTLINE

This paper is organized as follows. Section 2 gives some preliminaries on necessary concepts. Section 3 describes our new genetic algorithm and the traditional genetic algorithm. Section 4 illustrates the experimental parameters and gives the results of this paper. Then the results are compared and analyzed. Finally, Section 5 concludes this paper.

II. PRELIMINARIES

A bijective $n \times n$ S-box is a permutation on $\mathbb{F}_2^n$. Mathematically, S-box is a vectorial Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, which can be defined as a vector $F = (f_1, f_2, \ldots, f_n)$. The Boolean function $f_i : \mathbb{F}_2^n \to \mathbb{F}_2$, $i \in \{1, 2, \ldots, n\}$ is called the coordinate function of F. The component functions of an $n \times n$-function F are all the linear combinations of the coordinate functions with non all-zero coefficients.

Definition 1 (Differential Uniformity [8]): Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an $n \times n$ vectorial Boolean function. The derivative of F with regard to vector $a \in \mathbb{F}_2^n$ is $b = F(x \oplus a) \oplus F(x)$. The difference distribution table (DDT) of F is

$$DDT_F(a, b) = \#\left\{x \in \mathbb{F}_2^n | F(x) \oplus F(x \oplus a) = b\right\}.$$

The symbol # here represents the number of elements in the set. Differentially δ_F-uniform is the maximum value of $DDT_F(a, b)$ for every non-zero $a \in \mathbb{F}_2^n$ and every $b \in \mathbb{F}_2^n$, i.e.,

$$\delta_F = \max_{a,b \in \mathbb{F}_2^n; a \neq \mathbf{0}} DDT_F(a, b).$$

Definition 2 (Nonlinearity and Linearity [9]): Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an $n \times n$ vectorial Boolean function. The nonlinearity of an S-box F is defined as the minimum

Hamming distance between all non-zero component functions of F and all n-variable affine Boolean functions, which can be represented by the Walsh spectrum,

$$\mathcal{N}_F = 2^{n-1} - \frac{1}{2} \max_{a,b\in\mathbb{F}_2^n; b\neq \mathbf{0}} |\mathcal{W}_F(a,b)|.$$

The Walsh spectrum of an $n \times n$ F with respect to two vectors $a, b \in \mathbb{F}_2^n$ is

$$\mathcal{W}_F(a,b) = \sum_{x\in\mathbb{F}_2^n} (-1)^{b\cdot F(x)\oplus a\cdot x},$$

where $b \cdot F$ for all $b \in \mathbb{F}_2^n$ and $b \neq \mathbf{0}$ are called component functions and symbol $\cdot$ is an inner product over $\mathbb{F}_2$. The linear approximation table (LAT) of F is

$$LAT_F(a,b) = \frac{1}{2}\mathcal{W}_F(a,b).$$

The linearity of an S-box F is defined as

$$\mathcal{L}_F = \max_{a,b\in\mathbb{F}_2^n; b\neq \mathbf{0}} |\mathcal{W}_F(a,b)| = \max_{a,b\in\mathbb{F}_2^n; b\neq \mathbf{0}} 2\,|LAT_F(a,b)|.$$

Definition 3 (Boomerang Uniformity [7]): Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an $n \times n$ invertible vectorial Boolean function. For input difference $a \in \mathbb{F}_2^n$ and output difference $b \in \mathbb{F}_2^n$, the entries of the boomerang connectivity table (BCT) are defined as

$$BCT_F(a,b) = \#\left\{x \in \mathbb{F}_2^n | F^{-1}(F(x)+b) + F^{-1}(F(x+a)+b) = a\right\},$$

where F^{-1} denotes the compositional inverse of F. The boomerang uniformity of F is defined as

$$\beta_F = \max_{a,b\in\mathbb{F}_2^n; a,b\neq \mathbf{0}} BCT_F(a,b).$$

III. GENETIC ALGORITHMS

Genetic algorithm [27] is a computational model that simulates the evolution process of nature, which has been successfully applied to various optimization problems. Many researchers have also applied genetic algorithms to design block cipher primitives in recent years.

The genetic algorithm principle is based on Darwinian natural selection and Mendelian genetics. The selection method allows high-quality individuals to be more likely to survive, thereby improving the quality of individuals in the population. Mendelian genetics provides a theoretical basis for the population to produce new individuals. The crossover operator recombines the genes of the two-parent individuals to generate two new individuals, which is the primary way to generate new individuals. The mutation operator generates new individuals by changing the genes at specific loci. As the primary way of generating new individuals, genetic operators have a significant impact on the performance of the genetic algorithm. Traditional genetic operators are universal, but they can not guarantee to generate better new individuals. We design new genetic operators for the genetic algorithm to produce better individuals in the process of evolution.

Algorithm 1 depicts the framework of our genetic algorithm. In Algorithm 1, the size of population P is N. Individuals in the initial population are created by an unbalanced Feistel structure. r_p is a randomly generated probability. The parents in the population perform crossover according to probability p_c. p_m is the mutation probability. In our work, the termination condition of the genetic algorithm is that the maximum number MAX of generations is reached. $\mathcal{C}_F$ is the fitness function that calculates the fitness value f_p for the individual. Next, the components of the genetic algorithm will be introduced in detail.

Algorithm 1 The Framework of Our Genetic Algorithm

```
for each p ∈ P do
    p ← Unbalanced Feistel structure;
    f_p ← C_F(p);
end for
g ← 0;//Number of iterations
while g < MAX do
    g++;
    //Tournament selection operator;
    for i ∈ [0, N/2) do
        k individuals are randomly selected;
        Two individuals with the lowest fitness values are
        copied into the new population;
    end for
    //The process of crossover;
    for i ∈ [0, N/2) do
        if r_p < p_c then
            (p, q) ← randomly select two individuals from
            the population;
            (p, q) ← Crossover operator (p, q);
            f_p ← C_F(p);
            f_q ← C_F(q);
        end if
    end for
    //The process of mutation;
    for i ∈ [0, N/2) do
        if r_p < p_m then
            p ← The i-th individual in P;
            p ← Mutation operator (p);
            f_p ← C_F(p);
        end if
    end for
end while
```

Permutation Encoding:

The form of permutation encoding is intuitively more suitable for representing S-boxes. In this representation, the bijectivity property is automatically satisfied. An $n \times n$ S-box is represented as an array of 2^n integer numbers with elements in range $[0, 2^n - 1]$. Each value occurs exactly once in an array and represents one entry for the S-box lookup table.

Initial Population:

Individuals in the initial population are created by an unbalanced Feistel structure. We extend the method in [26] to

generate 8×8 S-boxes. Let f be a seven-variable nonlinear Boolean function, and $x_i \in \mathbb{F}_2^n$, $1 \le i \le 7$ is the variable. One round conversion of the unbalanced Feistel structure is

$$t(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_0 \oplus f(x_1, x_2, x_3, x_4, x_5, x_6, x_7)),$$

where

$$f(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = x_{r1} \cdot x_{r2} \oplus x_{r3} \cdot x_{r4} \oplus x_{r5} \cdot x_{r6}.$$

$r1, r2, r3, r4, r5, r6, r7$ are random integers in $[1, 7]$. Then, an S-box on $\mathbb{F}_2^8$ can be obtained through 8 rounds of conversion

$$F(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) = t^8(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7),$$

where $t^j = tt^{j-1}$, $2 \le j \le 8$, $t^1 = t$.

Fitness Function:

The fitness function design is related to the criteria for evaluating the S-box. The properties of the S-box concerned about in this paper mainly include differential uniformity, nonlinearity, and boomerang uniformity. Our fitness function is

$$\mathcal{C}_F = \delta_F + \mathcal{L}_F + \beta_F.$$

It is easy to see that the first term δ_F and the third term β_F are differential uniformity and boomerang uniformity, respectively. Both of these two terms in the S-box are as low as possible. However, the higher the nonlinearity, the better. For consistency, the second term in the fitness function is linearity $\mathcal{L}_F$. Algorithm 2 gives the calculation process of the fitness function.

Algorithm 2 Fitness Function $\mathcal{C}_F$

Input: Individual p

Output: $\delta_p + \mathcal{L}_p + \beta_p$

1: $(DDT_p, LAT_p, BCT_p) \leftarrow$ Calculate the DDT_p, LAT_p, BCT_p of p;

2: $\delta_p \leftarrow \max_{a,b \in \mathbb{F}_2^n; a \neq \mathbf{0}} DDT_p(a, b)$;

3: $\mathcal{L}_p \leftarrow \max_{a,b \in \mathbb{F}_2^n; b \neq \mathbf{0}} 2\left|LAT_p(a, b)\right|$;

4: $\beta_p \leftarrow \max_{a,b \in \mathbb{F}_2^n; a,b \neq \mathbf{0}} BCT_p(a, b)$;

5: **return** $\delta_p + \mathcal{L}_p + \beta_p$;

Selection Operator:

The k-tournament selection [28] is suitable for target minimization problem. First, k individuals are randomly selected from the population P. Then the two individuals with the smallest fitness values are copied into the new population. Repeat this process until the size of the new population reaches N.

A. NEW GENETIC OPERATORS

1) NEW CROSSOVER OPERATOR

In order to improve the performance of the genetic algorithm, we design a new crossover operator for the genetic algorithm. The fitness function considers three properties of an S-box: differential uniformity, linearity, and boomerang uniformity. The smaller their values, the better. In each iteration, the new crossover operator takes advantage of gene exchange to reduce the values of three properties.

The new crossover operator is described in Algorithm 3. First, randomly select two individuals p and q from the population as the two parents. Let $p = p_0, \cdots, p_{2^n-1}$ and $q = q_0, \cdots, q_{2^n-1}$. The crossover processes performed on p and q are similar. We take individual p as an example. Find the input-output differential pair (a, b) that satisfies $DDT_p(a, b) = \delta_p$ in the differential distribution table. For each pair (a, b), find $p_i, i \in [0, 2^n - 1]$ that increases $DDT_p(a, b)$ of p, and exchange p_i and p_j to obtain a new individual p', where $p_j = q_i$. If $\delta_{p'} \le \delta_p$, $\mathcal{L}_{p'} \le \mathcal{L}_p$, $\beta_{p'} \le \beta_p$, $DDT_{p'}(a, b) \le DDT_p(a, b)$ and no new value is added to δ_p after exchange, replace p with p'. If $DDT_{p'}(a, b) < DDT_p(a, b)$, find the next input-output differential pair (a, b) satisfying $DDT_{p'}(a, b) = \delta_{p'}$ and repeat the process. If $DDT_{p'}(a, b) = DDT_p(a, b)$, find the elements adding $DDT_{p'}(a, b)$ in p' and perform the same operation to reduce $DDT_{p'}(a, b)$. The process of reducing the boomerang uniformity is similar to that of reducing the differential uniformity. When reducing the linearity, it should be considered in two cases: $\mathcal{L}_p = \max_{a,b \in \mathbb{F}_2^n; b \neq \mathbf{0}} \mathcal{W}_p(a, b)$ and $\mathcal{L}_p = \max_{a,b \in \mathbb{F}_2^n; b \neq \mathbf{0}} -\mathcal{W}_p(a, b)$. When $\mathcal{L}_p = \max_{a,b \in \mathbb{F}_2^n; b \neq \mathbf{0}} \mathcal{W}_p(a, b)$, the purpose of gene exchange is to reduce the number of $b \cdot p_x = a \cdot x$; when $\mathcal{L}_p = \max_{a,b \in \mathbb{F}_2^n; b \neq \mathbf{0}} -\mathcal{W}_p(a, b)$, the purpose of gene exchange is to reduce the number of $b \cdot p_x \neq a \cdot x$.

2) NEW MUTATION OPERATOR

This paper also designs a new mutation operator. Randomly select a position $c_1 \in [0, 2^n - 1]$. Exchange the gene at position c_1 in individual p with genes at other positions in p in turn to generate new individuals. For the new individual p, if one or more of $\delta_{p'} < \delta_p$, $\mathcal{L}_{p'} < \mathcal{L}_p$ and $\beta_{p'} < \beta_p$ are satisfied, replace the original individual p with the new individual p'; otherwise, retain the original individual p and delete the new individual p'. The mutation process is described in Algorithm 4.

B. TRADITIONAL GENETIC OPERATORS

1) TRADITIONAL CROSSOVER OPERATOR

The partially mapped crossover (PMX crossover) [29] is the traditional crossover operator we use. Randomly select two individuals p and q from the population as the two parents. Let $p = p_0, \cdots, p_{2^n-1}$ and $q = q_0, \cdots, q_{2^n-1}$. Randomly select two positions (c_1, c_2), $c_1, c_2 \in [0, 2^n - 1]$, and exchange the gene fragments of the two parents between c_1 and c_2. Check the elements in the uncrossed gene segment of the

Algorithm 3 New Crossover Operator

Input: Parent individuals p and q
Output: Offspring individuals p and q

$p'' \leftarrow p$;
$p \leftarrow$ Exchange (p, q);
$q \leftarrow$ Exchange (q, p'');
return p and q;
The procedure of Exchange (p, q):
$p \leftarrow$ ReduceProcess(p, q,DDT);
$p \leftarrow$ ReduceProcess(p, q,LAT);
$p \leftarrow$ ReduceProcess(p, q,-LAT);
$p \leftarrow$ ReduceProcess(p, q,BCT);
return p;
The procedure of ReduceProcess (p, q,T):
if T=DDT **then**
  uniformity $(a, b) \leftarrow \mathrm{DDT}_p(a, b) = \delta_p$;
  condition $(p, q, i) \leftarrow p_i \oplus p_{i\oplus a} = b$;
  increase $(x, y) \leftarrow \mathrm{DDT}_{p'}(x, y) = \delta_p$ and $\mathrm{DDT}_p(x, y) < \delta_p$;
else if T=LAT **then**
  uniformity $(a, b) \leftarrow \mathrm{LAT}_p(a, b) = \frac{1}{2}\mathcal{L}_p$;
  condition $(p, q, i) \leftarrow b \cdot p_i = a \cdot i$;
  increase $(x, y) \leftarrow \mathrm{LAT}_{p'}(x, y) = \frac{1}{2}\mathcal{L}_p$ and $\mathrm{LAT}_p(x, y) < \frac{1}{2}\mathcal{L}_p$;
else if T=-LAT **then**
  T=LAT;
  uniformity $(a, b) \leftarrow \mathrm{LAT}_p(a, b) = -\frac{1}{2}\mathcal{L}_p$;
  condition $(p, q, i) \leftarrow b \cdot p_i \neq a \cdot i$;
  increase $(x, y) \leftarrow |\mathrm{LAT}_{p'}(x, y)| = \frac{1}{2}\mathcal{L}_p$ and $|\mathrm{LAT}_p(x, y)| < \frac{1}{2}\mathcal{L}_p$;
else if T=BCT **then**
  uniformity $(a, b) \leftarrow \mathrm{BCT}_p(a, b) = \beta_p$;
  condition $(p, q, i) \leftarrow p^{-1}(p_i \oplus b) \oplus p^{-1}(p_{i\oplus a} \oplus b) = a$;
  increase $(x, y) \leftarrow \mathrm{BCT}_{p'}(x, y) = \beta_p$ and $\mathrm{BCT}_p(x, y) < \beta_p$;
end if
for uniformity (a, b), $a, b \in [0, 2^n - 1]$ **do**
  $p' \leftarrow$ Exchange p_i and p_j, where $p_i = q_j$;
  for condition (a, b, i), $i \in [0, 2^n - 1]$ **do**
    if $\delta_{p'} \leq \delta_p, \mathcal{L}_{p'} \leq \mathcal{L}_p, \beta_{p'} \leq \beta_p, \mathrm{T}_{p'}(a, b) \leq \mathrm{T}_p(a, b)$ and $\nexists x, y \in [0, 2^n - 1]$, increase (x, y) **then**
      $p \leftarrow p'$;
      If $\mathrm{T}_p(a, b)$ is reduced, jump out of this loop;
    end if
  end for
end for
return p;

first parent p. If an element is the same as the element at position $j, j \in [c_1, c_2]$, replace it with the element p_j in q. Repeat this process until p becomes a permutation with no repeating elements. Then perform the same operation on the second parent q. This process is described in Algorithm 5.

Algorithm 4 New Mutation Operator

Input: Parent individual p
Output: Offspring individual p

$c_1 \leftarrow$ Randomly generate a position;
for $i \in [0, 2^n - 1]$ **do**
  if $i \neq c_1$ **then**
    $p' \leftarrow$ Exchange p_i and p_{c_1};
    $f_{p'} \leftarrow \mathcal{C}_F(p')$;
    if $f_{p'} < f_p$ or $f_{p'} = f_p, \delta_{p'} < \delta_p$ or $f_{p'} = f_p, \delta_{p'} < \delta_p, \mathcal{L}_{p'} < \mathcal{L}_p$ **then**
      $p \leftarrow p'$;
    end if
  end if
end for
return p;

Algorithm 5 Traditional Crossover Operator

Input: Parent individuals p and q
Output: Offspring individuals p and q

$(c_1, c_2) \leftarrow$ Randomly generate two positions;
exchange(p; q; c_1; c_2)= $\begin{pmatrix} p_0, \cdots, p_{c_1}, \cdots, q_{c_2}, \cdots, p_{2^n-1} \\ q_0, \cdots, p_{c_1}, \cdots, p_{c_2}, \cdots, q_{2^n-1} \end{pmatrix}$;
for $i \in [0, c_1)$ **do**
  if $p_i = q_j, j \in [c_1, c_2]$ **then**
    $p_i \leftarrow p_j$ e.g. $p = \{p_0, \cdots, p_j, \cdots, q_{c_1}, \cdots, q_{c_2}, \cdots, p_{2^n-1}\}$;
  end if
  if $q_i = p_j, j \in [c_1, c_2]$ **then**
    $q_i \leftarrow q_j$ e.g. $q = \{q_0, \cdots, q_j, \cdots, p_{c_1}, \cdots, p_{c_2}, \cdots, q_{2^n-1}\}$;
  end if
end for
for $i \in (c_2, 2^n - 1]$ **do**
  if $p_i = q_j, j \in [c_1, c_2]$ **then**
    $p_i \leftarrow p_j$ e.g. $p = \{p_0, \cdots, q_{c_1}, \cdots, q_{c_2}, \cdots, p_j, \cdots, p_{2^n-1}\}$;
  end if
  if $q_i = p_j, j \in [c_1, c_2]$ **then**
    $q_i \leftarrow q_j$ e.g. $q = \{q_0, \cdots, p_{c_1}, \cdots, p_{c_2}, \cdots, q_j, \cdots, q_{2^n-1}\}$;
  end if
end for
return p and q;

2) TRADITIONAL MUTATION OPERATOR

The inversion mutation [30] is the traditional mutation operator we use. First, two positions (c_1, c_2), $c_1, c_2 \in [0, 2^n - 1]$ are randomly selected for the individual p to be mutated, where $p = p_0, \cdots, p_{c_1}, \cdots, p_{c_2}, \cdots, p_{2^n-1}$. Then, the genes between two positions (c_1, c_2) in individual p are arranged in an inverted order to obtain a new $p = p_0, \cdots, p_{c_2}, \cdots, p_{c_1}, \cdots, p_{2^n-1}$.

C. CONVERGENCE ANALYSIS OF NEW GENETIC ALGORITHM

The convergence of the new genetic algorithm can be analyzed by the Markov chain, which proves the rationality of this method theoretically.

Definition 4 (Markov Chain): Let $\boldsymbol{I} = \{i_0, \ldots, i_n\}$ be the values for stochastic process $\{X(n), n \geq 0\}$. For any $i_0, \ldots, i_n$, if $P\{X(0) = i_0, \ldots, X(n) = i_n\} > 0$, then

$$P\{X(n+1) = i_{n+1} | X(0) = i_0, \ldots, X(n) = i_n\} = P\{X(n+1) = i_{n+1} | X(n) = i_n\}. \quad (1)$$

$\{X(n), n \geq 0\}$ is defined as Markov chain.

Definition 5 (Transition Probability): Let $P_{i,j}(m, n) = P\{X(n) = j | X(m) = i, n > m\}$ be the transition probability. $P_{i,j}(m, n)$ satisfies the properties: $P_{i,j}(m, n) \geq 0$ and $\sum_{j \in \boldsymbol{I}} P_{i,j}(m, n) = 1$.

Definition 6 (Homogeneous Markov Chain): For Markov chain, if

$$P_{i,j}(m, m+1) = P\{X(m+1) = j | X(m) = i\} = P_{i,j}, \quad (2)$$

where $i, j \in I$, i.e. the transition probability from state i to state j is independent of the time starting point m, then such Markov chain is called homogeneous Markov chain.

Definition 7 (Stochastic Matrix): For homogeneous Markov chain, $P_{i,j}$ is the one-step transition probability. A matrix $\mathbf{P} = \{P_{i,j}\}$ composed of all $P_{i,j}$, $(i, j \in I)$ is called a stochastic matrix.

The different operations of the genetic algorithm are performed independently, and the new population has nothing to do with the previous generations of the parent population. Therefore, the genetic algorithm can be described as a homogeneous Markov chain. Before the convergence analysis, we need to observe the design of the new genetic operators. It can be seen from the design of the new genetic operators that the genic changes are nonlinear, and the properties of the newly generated individuals are not inferior to those of the parent individuals, which is very important for the convergence analysis.

Theorem 1: The probability that the new genetic algorithm converges to the optimal solution is 1.

Proof: The possible states $\boldsymbol{I}$ of the population are divided into the state $\boldsymbol{I_o}$ and the state $\boldsymbol{I_n}$, where $\boldsymbol{I_o}$ contains the optimal solution, and $\boldsymbol{I_n}$ does not include the optimal solution. $\boldsymbol{I} = \boldsymbol{I_o} \cup \boldsymbol{I_n}$ and $\boldsymbol{I_o} \cap \boldsymbol{I_n} = \emptyset$.

The transition probability of the selection operator from state i to state j is $s_{i,j}$. Similarly, $c_{i,j}$ and $m_{i,j}$ are the transformation probabilities of crossover operator and mutation operator, respectively. $\boldsymbol{S} = \{s_{i,j}\}$, $\boldsymbol{C} = \{c_{i,j}\}$, and $\boldsymbol{M} = \{m_{i,j}\}$ are the corresponding stochastic matrices. Then the population state transition $\boldsymbol{R} = \boldsymbol{SCM} = r_{i,j}$ of genetic algorithm is easily proved to be positive definite. Let $P_j(t), t = 0, 1, \ldots\ldots$ be the probability that the population is state j at time t. Genetic algorithm can be described as a homogeneous Markov chain. Therefore, the stable probability distribution of $P_j(t)$ is independent of its initial probability distribution.

Both the new crossover operator and mutation operator retain individuals with better properties, ensuring that the transition probability from $\boldsymbol{I_o}$ to $\boldsymbol{I_n}$ is equal to 0 and the probability from any state to $\boldsymbol{I_o}$ is greater than 0, i.e. $r_{i,j} = 0(\forall i \in \boldsymbol{I_o}, \forall j \notin \boldsymbol{I_o})$ and $r_{i,j} > 0(\forall i \in \boldsymbol{I}, \forall j \in \boldsymbol{I_o})$. Then for $\forall i \in \boldsymbol{I}, \forall j \notin \boldsymbol{I_o}, r_{i,j}^t = 0, (t \to \infty)$ i.e. $P_j(\infty) = 0, (j \notin \boldsymbol{I_o})$. The probability that the population converges to the state $\boldsymbol{I_n}$ is 0, i.e. the probability that the new genetic algorithm converges to the optimal solution is 1. ■

IV. EXPERIMENTAL SETUP AND RESULTS

This paper uses the new genetic algorithm and the traditional genetic algorithm to search for 8 × 8 S-boxes with low differential uniformity, high nonlinearity, and low boomerang uniformity.

A. EXPERIMENTAL SETUP

For the traditional genetic algorithm and the new genetic algorithm, we run 30 experiments, respectively. Except for the different genetic operators, the other parameters of the two genetic algorithms are the same. The parameter values are determined based on experience and experimental feedback. The population size N is 256. The tournament size k is set to 3. Different crossover probabilities and mutation probabilities have no significant impact on the search of traditional genetic algorithm. The higher the crossover probability and the mutation probability for our new genetic algorithm, the better. Therefore, we set these two parameters to relatively large values. Crossover probability $p_c = 0.9$, and mutation probability $p_m = 0.1$. The maximum number of iterations is determined by observing the output of experimental results, and MAX=400.

B. EXPERIMENTAL RESULTS AND ANALYSIS

In our work, in addition to the differential uniformity and nonlinearity, we also consider the boomerang uniformity. Table 3 describes the distributions of these three properties in the initial population and the final population. The data 16#13 in Table 3 means that the number of differential uniformity $\delta_p = 16$ in the initial population is 13.

As can be seen from Table 3, for the initial population created by the unbalanced Feistel structure, the differential uniformity, nonlinearity, and boomerang uniformity are concentrated at 64, 64, and 256, respectively. The initial population's best differential uniformity, nonlinearity, and boomerang uniformity are 16, 96 and 52, respectively. The best S-box created by the unbalanced Feistel structure is given in Table 1. It can be seen that the properties of the initial population created by the unbalanced Feistel structure are not ideal, especially the boomerang uniformity. At the end of the iteration, the differential uniformity, nonlinearity, and boomerang uniformity of the population obtained by our new genetic algorithm are concentrated at 12, 94, and 20, respectively. At this time, the best values of differential uniformity, nonlinearity, and boomerang uniformity in the population are 6, 108, and 10. At the end of the population, there

TABLE 1. The best S-box created by the unbalanced Feistel structure.

0	1	2	3	4	39	6	33	8	27	78	87	12	60	66	124
16	49	54	17	156	159	174	173	24	42	121	71	132	150	249	225
32	35	98	104	109	69	34	7	41	56	47	62	100	93	102	91
48	19	84	120	243	250	143	142	57	9	26	44	232	242	216	195
64	73	70	79	196	235	209	248	218	193	139	153	68	126	14	58
82	122	112	95	94	85	125	117	201	246	187	138	205	212	183	165
96	106	38	37	168	140	241	219	255	231	238	244	45	31	43	29
114	89	18	55	52	53	88	83	236	208	220	229	160	176	151	135
128	129	147	144	141	171	158	190	152	136	207	214	149	163	203	240
145	180	162	131	20	23	50	51	137	188	253	199	28	13	116	108
164	167	245	252	224	204	191	157	189	175	170	186	251	197	234	210
181	146	198	237	127	119	22	21	172	155	154	169	118	110	80	75
192	200	213	223	76	103	74	101	72	81	10	25	222	227	134	182
211	254	230	206	215	221	226	233	90	99	63	11	86	77	59	40
228	239	179	178	36	5	111	67	105	115	107	113	177	133	166	148
247	217	130	161	185	184	194	202	123	65	92	97	61	46	30	15

TABLE 2. The best S-box generated by the new genetic algorithm.

154	150	45	38	27	130	171	1	56	159	249	16	96	49	53	23
164	100	122	155	151	251	176	143	136	239	111	28	47	144	62	98
234	190	138	33	133	157	120	25	17	218	72	223	224	148	114	18
252	149	66	188	35	233	182	146	34	229	240	191	167	107	137	201
205	46	231	183	253	225	37	185	119	180	123	195	132	245	57	217
61	140	213	26	156	244	87	209	200	238	242	32	129	60	177	199
198	142	127	3	153	118	230	147	36	93	65	15	227	67	241	235
169	103	71	85	192	101	106	236	102	160	0	214	113	42	81	82
83	174	175	80	70	163	110	134	75	78	69	9	39	210	105	116
14	186	226	189	8	90	84	162	250	196	79	88	41	12	89	74
43	172	254	19	95	104	181	152	246	24	194	73	30	55	2	178
220	48	207	63	193	165	168	161	117	94	166	237	109	108	86	135
219	248	255	115	10	4	52	197	121	247	131	212	203	44	124	54
170	215	126	179	21	11	50	58	68	91	20	5	77	40	64	187
97	221	6	59	22	222	31	125	216	158	7	51	141	128	206	99
139	243	232	228	184	208	211	92	13	76	173	112	29	202	145	204

TABLE 3. The property distribution of S-boxes in the initial population and the final populations.

Distribution	Initial population	Final population
Best δ_p	16#13	6#4
Maximum δ_p	64#91	12#105
Best $\mathcal{N}_p$	96#36	108#2
Maximum $\mathcal{N}_p$	64#120	94#101
Best β_p	52#1	10#2
Maximum β_p	256#186	20#75

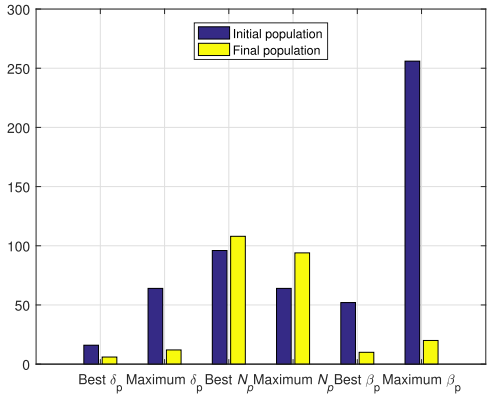

FIGURE 1. Comparison of property distributions between initial population and final population.

are 119 $\delta_p \leq 10$, 47 $\mathcal{N}_p \geq 96$ and 9 $\beta_p \leq 16$. Figure 1 shows the comparison of the distributions of the three properties in the initial population and the final population. As can be seen from Figure 1, on the whole, the new genetic algorithm improves the properties of S-boxes created by the unbalanced Feistel structure.

In Table 4, S-box1-S-box4 are generated by our new genetic algorithm, and S-box5 is generated by the traditional genetic algorithm. It can be seen from Table 4 that the S-box1 and S-box2 generated by our new genetic algorithm have the best cryptographic properties: the lowest differential uniformity 6, the highest nonlinearity 108, and the lowest boomerang uniformity 10. Table 2 shows the lookup table of S-box1.

TABLE 4. S-boxes generated in this paper.

S-box	δ_F	$\mathcal{N}_F$	β_F
S-box1	6	108	10
S-box2	6	108	10
S-box3	6	104	12
S-box4	6	100	14
S-box5	8	98	16

Table 5 compares the cryptographic properties of S-boxes generated in different ways. The values of random S-box are the expected values of differential uniformity, nonlinearity,

and boomerang uniformity given in [33]–[34]. As can be seen from Table 5, the S-box generated by the new genetic algorithm has better properties than the random S-box. Moreover, the S-box generated by the new genetic algorithm is comparable with those generated by other methods.

TABLE 5. The comparison of cryptographic properties between the best S-boxes generated in different ways.

S-box	Method	δ_F	$\mathcal{N}_F$	β_F
S-box1	New genetic algorithm	6	108	10
[21]	Traditional genetic algorithm	12	98	20
[23]	Reversed genetic algorithm	6	112	6
[31]	Tweaking	6	106	14
[32]	Gradient descent method	8	104	16
[15]	Reinforcement Learning	12	98	20
[33], [34]	Random	11.34	92.7	20.2

In summary, the new genetic algorithm has successfully improved the properties of the S-boxes created by the unbalanced Feistel structure. Moreover, the S-boxes generated by the new genetic algorithm have better properties than those generated by the traditional genetic algorithm, demonstrating the new genetic algorithm's effectiveness and superiority in developing S-boxes.

V. CONCLUSION

In this paper, a genetic algorithm is used to improve the properties of the S-boxes created by the Feistel structure. New genetic operators are designed for the genetic algorithm to develop 8×8 S-boxes with low differential uniformity, high nonlinearity, and low boomerang uniformity. It is the first time that a genetic algorithm has been used to improve the boomerang uniformity of the S-box. Experimental results show that the new genetic algorithm successfully improves the properties of the S-boxes created by the Feistel structure. The S-boxes generated by the new genetic algorithm have better properties than those generated by the traditional one, which shows the effectiveness and superiority of the new genetic algorithm. In the future, genetic algorithms can be used to generate S-boxes of different sizes. On the other hand, other new genetic operators can be designed to generate S-boxes with better performances.

REFERENCES

[1] D. Wagner, "The boomerang attack," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 1636. Berlin, Germany: Springer, Mar. 1999, pp. 156–170.

[2] B. Eli, D. Orr, and K. Nathan, "Related-key boomerang and rectangle attacks," in *Advances in Cryptology-EUROCRYPT*. Berlin, Germany: Springer, 2005, pp. 507–525.

[3] H. Seokhie, K. Jongsung, L. Sangjin, and P. Bart, "Related-key rectangle attacks on reduced versions of SHACAL-1 and AES-192," in *Fast Software Encryption*. Berlin, Germany: Springer, 2005, pp. 368–383.

[4] O. Dunkelman, N. Keller, and J. Kim, "Related-key rectangle attack on the full SHACAL-1," in *Selected Areas in Cryptograph* (Lecture Notes in Computer Science), vol. 4356. Montreal, QC, Canada: Springer, 2006, pp. 28–44.

[5] K. Jongsung, H. Seokhie, and P. Bart, "Related-key rectangle attacks on reduced AES-192 and AES-256," in *Fast Software Encryption*. Berlin, Germany: Springer, 2007, pp. 225–241.

[6] C. Carlos, H. Tao, P. Thomas, S. Yu, and S. Ling, "Boomerang connectivity table: A new cryptanalysis tool," in *Advances in Cryptology-EUROCRYPT*. Cham, Switzerland: Springer, 2018, pp. 683–714.

[7] C. Boura and A. Canteaut, "On the boomerang uniformity of cryptographic Sboxes," *IACR Trans. Symmetric Cryptol.*, vol. 2018, no. 3, pp. 290–310, Sep. 2018.

[8] N. Kaisa, "Differentially uniform mappings for cryptography," in *Advances in Cryptology-EUROCRYPT*. Berlin, Germany: Springer, 1994, pp. 55–64.

[9] C. Carlet, "Vectorial Boolean functions for cryptography," in *Boolean Models and Methods in Mathematics*. Cambridge, U.K.: Cambridge Univ. Press, 2010, pp. 398–470.

[10] K. Li, L. Qu, B. Sun, and C. Li, "New results about the boomerang uniformity of permutation polynomials," *IEEE Trans. Inf. Theory*, vol. 65, no. 11, pp. 7542–7553, May 2019.

[11] S. Mesnager, C. M. Tang, and M. S. Xiong, "On the boomerang uniformity of (quadratic) permutations over F_{2^n}," *Des., Codes Cryptogr.*, vol. 88, pp. 2233–2246, Jun. 2020, doi: 10.1007/s10623-020-00775-2.

[12] M. Calderini and I. Villa, "On the boomerang uniformity of some permutation polynomials," *Cryptogr. Commun.*, vol. 12, pp. 1161–1178, Jun. 2020, doi: 10.1007/s12095-020-00439-x.

[13] Y. P. Wang, Q. Wang, and W. G. Zhang, "Boomerang uniformity of normalized permutation polynomials of low degree," *Applicable Algebra Eng., Commun. Comput.*, vol. 31, pp. 307–322, Apr. 2020, doi: 10.1007/s00200-020-00431-1.

[14] K. Li, C. Li, T. Helleseth, and L. Qu, "Cryptographically strong permutations from the butterfly structure," *Des. Codes Cryptogr.*, vol. 89, no. 3, pp. 737–761, Feb. 2021, doi: 10.1007/s10623-020-00837-5.

[15] G. Kim, H. Kim, Y. Heo, Y. Jeon, and J. Kim, "Generating cryptographic S-boxes using the reinforcement learning," *IEEE Access*, vol. 9, pp. 83092–83104, 2021, doi: 10.1109/ACCESS.2021.3085861.

[16] A. H. Zahid, A. M. Iliyasu, M. Ahmad, M. M. U. Shaban, M. J. Arshad, H. S. Alhadawi, and A. A. A. El-Latif, "A novel construction of dynamic S-box with high nonlinearity using heuristic evolution," *IEEE Access*, vol. 9, pp. 67797–67812, 2021, doi: 10.1109/ACCESS.2021.3077194.

[17] S. Ibrahim and A. M. Abbas, "A novel optimization method for constructing cryptographically strong dynamic S-boxes," *IEEE Access*, vol. 8, pp. 225004–225017, 2020, doi: 10.1109/ACCESS.2020.3045260.

[18] M. M. Dimitrov, "On the design of chaos-based S-boxes," *IEEE Access*, vol. 8, pp. 117173–117181, 2020, doi: 10.1109/ACCESS.2020.3004526.

[19] M. Long and L. Wang, "S-box design based on discrete chaotic map and improved artificial bee colony algorithm," *IEEE Access*, vol. 9, pp. 86144–86154, 2021, doi: 10.1109/ACCESS.2021.3069965.

[20] A. A. Alzaidi, M. Ahmad, M. N. Doja, E. A. Solami, and M. M. S. Beg, "A new 1D chaotic map and β-hill climbing for generating substitution-boxes," *IEEE Access*, vol. 6, pp. 55405–55418, 2018, doi: 10.1109/ACCESS.2018.2871557.

[21] P. Stjepan, P. Kostas, E. Barış, B. Lejla, and J. Domagoj, "Confused by confusion: Systematic evaluation of DPA resistance of various S-boxes," in *Progress in Cryptology-INDOCRYPT*. Cham, Switzerland: Springer, 2014, pp. 374–390.

[22] Y. Wang, K.-W. Wong, C. Li, and L. Yang, "A novel method to design S-box based on chaotic map and genetic algorithm," *Phys. Lett. A*, vol. 376, nos. 6–7, pp. 827–833, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S03759601 12000291

[23] I. Georgi, N. Nikolay, and N. Svetla, "Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties," *Cryptogr. Commun.*, vol. 8, no. 2, p. 247, 2016, doi: 10.1007/s12095-015-0170-5.

[24] P. Stjepan, K. Karlo, J. Domagoj, and C. Claude, "A search for differentially-6 uniform $(n, n-2)$ functions," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–7, doi: 10.1109/CEC.2018.8477646.

[25] N. Ivica, "How to use metaheuristics for design of symmetric-key primitives," in *Advances in Cryptology-ASIACRYPT*. Cham, Switzerland: Springer, 2017, pp. 369–391.

[26] Y. Q. Li and M. S. Wang, "Constructing S-boxes for lightweight cryptography with Feistel structure," in *Cryptographic Hardware and Embedded Systems-CHES*. Berlin, Germany: Springer, 2014, pp. 127–146.

[27] H. John, "Adaptation in natural and artificial systems," Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. Michigan Press, Ann Arbor, MI, USA, 1975.

[28] W. Thomas, *Global Optimization Algorithm: Theory and Application*. Hefei, China, 2009.

[29] D. E. Goldberg and R. Lingle, Jr., "Alleles, loci, and the traveling salesman problem," in *Proc. 1st Int. Conf. Genetic Algorithms*. Pittsburgh, PA, USA: Lawrence Erlbaum Associates, 1985, pp. 154–159.
[30] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Berlin, Germany: Springer, 2003.
[31] J. Fuller and W. Millan, "Linear redundancy in S-boxes," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 2887. Berlin, Germany: Springer, 2003, pp. 74–86.
[32] O. Kazymyrov, V. Kazymyrova, and R. Oliynykov, "A method for generation of high-nonlinear S-boxes based on gradient descent," *Mat. Vopr. Kriptogr.*, vol. 5, no. 2, pp. 71–78, 2014.
[33] J. Daemen and V. Rijmen, "Probability distributions of correlation and differentials in block ciphers," *J. Math. Cryptol.*, vol. 1, no. 3, pp. 221–242, 2007, doi: 10.1515/JMC.2007.011.
[34] S. Tian, C. Boura, and L. Perrin, "Boomerang uniformity of popular S-box constructions," *Des., Codes Cryptogr.*, vol. 88, no. 9, pp. 1959–1989, Sep. 2020.

MAN KANG received the B.S. degree from Hebei GEO University, Shijiazhuang, China, in 2013, and the M.S. degree from the Beijing University of Chemical Technology, Beijing, China, in 2017. She is currently pursuing the Ph.D. degree with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing. Her research interests include cryptography and information security.

MINGSHENG WANG received the Ph.D. degree from Beijing Normal University, Beijing, China, in 1994. He is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing. His research interests include computational algebra, cryptography, and information security.