







Article

Enhancement of Non-Permutation Binomial Power Functions to Construct Cryptographically Strong S-Boxes

Herman Isa ^{1,2,3,*} , Syed Alwee Aljunid Syed Junid ^{1,2,*} , Muhammad Reza Z'aba ³ , Rosdisham Endut ^{1,2} ,
Syed Mohammad Ammar ¹  and Norshamsuri Ali ^{1,2} 

¹ Faculty of Electronic Engineering & Technology, Universiti Malaysia Perlis, Arau 02600, Perlis, Malaysia

² Centre of Excellence Advanced Communication Engineering (ACE), Universiti Malaysia Perlis, Arau 02600, Perlis, Malaysia

³ MIMOS Berhad, Kuala Lumpur 57000, Malaysia

* Correspondence: hermanisa@studentmail.unimap.edu.my (H.I.); syedalwee@unimap.edu.my (S.A.A.S.J.)

Abstract: A Substitution box (S-box) is an important component used in symmetric key cryptosystems to satisfy Shannon's property on confusion. As the only nonlinear operation, the S-box must be cryptographically strong to thwart any cryptanalysis tools on cryptosystems. Generally, the S-boxes can be constructed using any of the following approaches: the random search approach, heuristic/evolutionary approach or mathematical approach. However, the current S-box construction has some drawbacks, such as low cryptographic properties for the random search approach and the fact that it is hard to develop mathematical functions that can be used to construct a cryptographically strong S-box. In this paper, we explore the non-permutation function that was generated from the binomial operation of the power function to construct a cryptographically strong S-box. By adopting the method called the *Redundancy Removal Algorithm*, we propose some enhancement in the algorithm such that the desired result can be obtained. The analytical results of our experiment indicate that all criteria such as bijective, nonlinearity, differential uniformity, algebraic degree and linear approximation are found to hold in the obtained S-boxes. Our proposed S-box also surpassed several bijective S-boxes available in the literature in terms of cryptographic properties.

Keywords: s-box; cryptographically strong s-box; binomial power function; non-permutation function; redundancy removal algorithm

MSC: 11T71; 94A60; 68P25



Citation: Isa, H.; Syed Junid, S.A.A.; Z'aba, M.R.; Endut, R.; Ammar, S.M.; Ali, N. Enhancement of Non-Permutation Binomial Power Functions to Construct Cryptographically Strong S-Boxes. *Mathematics* **2023**, *11*, 446. <https://doi.org/10.3390/math11020446>

Academic Editors: Liehuang Zhu, Meng Li and Zijian Zhang

Received: 9 October 2022

Revised: 25 November 2022

Accepted: 30 November 2022

Published: 14 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

At a fundamental level, Claude Shannon's properties of confusion and diffusion [1] often made for bench marking the security of a symmetric encryption. Confusion is defined as the complexity of the relationship between secret-key and cipher-text while diffusion is defined as the degree of influence of single input bit changed in the resulting ciphertext. To realize the confusion and diffusion of a symmetric encryption, several layers of substitution and permutation operation, called Substitution-Permutation Networks (SPNs), are applied. The substitution layer is a nonlinear operation used to improve the overall confusion of the encryption. On the contrary, the permutation layer is a linear operation that increases the measure of diffusion.

In symmetric encryption, the substitution layer is typically referred to as Substitution box (S-box). An S-box plays a crucial role against various cryptanalysis. Therefore, an S-box needs to be cryptographically strong by at least having high nonlinearity, low differential uniformity, a high algebraic degree and complex algebraic expression to resist cryptanalysis, such as differential attack first introduced by Biham and Shamir [2], linear attack described by Matsui [3], higher order differential attacks that were introduced by Lai [4] and Knudsen [5], interpolation attacks introduced by Jakobsen and Knudsen [6]

and, lastly, algebraic attack introduced by Courtois and Pieprzyk [7]. As a result, research to improve the cryptographic properties of an S-box and its implementation efficiency can be found in the literature. To date, most published constructions of an S-box are based on the bijective functions of elements of four or eight bits.

1.1. Problem Statement

Over the last 20 years, there has been great progress in the field of nonlinear power functions which serve as an S-box. The most successful and widely used technique is the one adopted by the Advanced Encryption Standard (AES) [8] that makes use of the multiplicative inverse function over a finite field \mathbb{F}_{2^8} . Often, the cryptographic properties exhibited by AES's S-box become a benchmark for other researchers to construct their S-boxes. This is because AES S-box is conjectured as having optimal cryptographic properties [9] with respect to resistance against linear, differential and algebraic attacks.

Nonlinearity, which is one of the cryptographic properties of S-box, remains a great challenge for future research. Among the main considerations in constructing a new S-box is the following question: *Are there more optimal cryptographic properties than the one achieved by AES's S-box?* In other words, this question exclusively relates to the nonlinearity property achieved by an AES S-box. Although there were many new constructions proposed in the last 20 years, the cryptographic properties achieved by AES S-box remain *unbeatable*. Therefore, we aim to further explore the construction of an S-box.

Our main interest is the construction of bijective S-boxes when n is even, (i.e., $n = 8$). Most of our research relates to the non-permutation of binomial power functions over \mathbb{F}_{2^8} . The design criteria of an S-box are affected by resistance against the main attacks on block ciphers [10], as mentioned earlier. All these attacks contribute to the countermeasures of cryptographically strong S-boxes.

1.2. Contribution

Generally, S-box construction can be categorized into three generic methods which are random search, heuristic method and mathematical function approaches. There also exist S-box constructions that combine all or several of these generic methods to produce their desired result. In this paper, we are more interested in exploring and enhancing the S-box construction from a non-permutation power function as previously proposed by Mamadolimov et al. [11] and subsequently improved by Isa et al. [12]. Their method starts with the binomial operation of power functions over a finite field to select the initial S-box. The main criteria in selecting the initial S-box are that the S-box must exhibit optimal cryptographic properties, except balancedness. Then, using a heuristic algorithm known as the *Redundancy Removal Algorithm*, the final result with the permutation function is produced.

We improve the method by examining the properties exhibits from binomial power functions in more detail. This includes the main cryptographic properties (i.e., NL, DU and AD) and its redundant elements, since composition of two power functions are prone to produce a non-permutation power function [13]. From there, several other steps such as elements exchange and elements rotation are conducted to produce final S-box with desired cryptographically strong properties. The final S-box that we obtained exhibits better cryptographic properties compared to S-box with the same construction proposed by Mamadolimov et al. [11] and Isa et al. [12].

1.3. Organisation

The rest of the paper is organized as follows. In the second section, we discuss the security requirements for an S-box and the available S-box design methods in the literature. In the third section, we present some preliminaries and our proposed S-box construction. Following this, in the fourth section, we provide a discussion and comparative analysis between our findings and the S-boxes available in the literature. We conclude our paper in last section.

2. Security Requirement and S-Box Design Methods

2.1. S-Box Properties

An S-box plays a crucial role in resisting cryptanalysis. A cryptographically strong S-box is classified based on its exhibited cryptographic properties. Moreover, an S-box must exhibit high nonlinearity, low differential uniformity, a high algebraic degree and complex algebraic expression to resist cryptanalysis, such as linear [3], differential [2], algebraic [7] and interpolation [6] attacks. Therefore, a lot of research is still in place to investigate and strengthen the cryptographic properties of each S-box proposed.

There is also much research conducted in designing and constructing an S-box. Typically, the measurement of *cryptographically strong* S-box properties depends on assumptions. To avoid more confusion in determining a cryptographically strong S-box, Carlet takes responsibility by facilitating and simplifying the desired properties in designing *good* S-boxes [14]. In particular, the desired properties suggested by Carlet [14] and later followed by Piret et al. [15], are bench-marked against AES’s S-box. This is because AES’s S-box is already conjectured as having the optimal cryptographic properties [9].

Therefore, based on the suggestions made in [14,15], we set the desired values for each of the S-box properties to be considered cryptographically strong, as summarized in Table 1. The following subsections explain each property in detail.

Table 1. Desired Value for Cryptographically Strong S-boxes.

Properties	Desired Value
High Nonlinearity	$100 < NL \leq 120$
Low Differential Uniformity	$2 \leq DU \leq 6$
High Algebraic Degree	$4 \leq AD \leq 7$
Low Linear Approximation	$8 \leq LA < 28$
Algebraic Complexity	Complex
Fixed & Opposite Fixed Points	None/Low
Balanced Output	Permutation

Let \mathbb{F} be a finite field with 2 elements, while \mathbb{F}_{2^n} is a finite field with 2^n elements. An $n \times n$ S-box is a Boolean map:

$$F : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n} = (f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))$$

where $f_1(x_1, \dots, x_n)$ to $f_n(x_1, \dots, x_n)$ are called as component functions of F .

2.1.1. Nonlinearity (NL)

Let $c = (c_1, c_2, \dots, c_n)$ be a nonzero element in \mathbb{F}_{2^n} and $c \cdot F = c_1 f_1 + c_2 f_2 + \dots + c_n f_n$ be a linear combination of the coordinate Boolean functions (f_1, f_2, \dots, f_n) of F . The NL of an S-box, F , is the Hamming distance between the set of all affine functions over \mathbb{F}_{2^n} and the set of all non-constant linear combinations of component functions of F , as defined below:

$$NL(F) = \min_{c \in \mathbb{F}_{2^n}, c \neq 0} NL(c \cdot F)$$

Claude [14] suggested that the value of NL should be as close as possible to the best-known NL (i.e., $NL \approx 112$) to thwart linear cryptanalysis [3]. In our study, we set the minimum threshold for NL as 100 (i.e., $NL > 100$).

2.1.2. Differential Uniformity (DU)

By excluding the trivial entry case (i.e., $a = b = 0$) from the difference distribution table, the largest value present in the table can determine the value of DU. The value of DU is defined as follows:

$$DU(F) = \max_{a,b \in \mathbb{F}_2^n, a \neq 0} |\{x \in \mathbb{F}_2^n : F(x+a) + F(x) = b\}|$$

A smaller value of DU is preferable [14]. In our case, we determine that the value of DU must lie in the range of $2 \leq DU \leq 6$ to resist differential cryptanalysis [2].

2.1.3. Algebraic Degree (AD)

$deg(f)$ is denoted from the number of variables in the largest monomial for the component function f of an S-box. Therefore, the AD of an S-box is determined by the maximum degree of all component functions, as follows:

$$AD(F) = \max\{deg(f_1), deg(f_2), \dots, deg(f_n)\}$$

The value of the AD of an S-box must be high [14] in order to resist higher order differential cryptanalysis [5]. In our study, we set $AD \geq 4$.

2.1.4. Linear Approximation (LA)

The LA of an S-box is defined as follows:

$$LA(F) = \max_{\alpha, \beta \in \mathbb{F}_2^n, \alpha \neq 0} \{\#\{X \in \mathbb{F}_2^n \mid \alpha \cdot X = \beta \cdot f(X)\} - 2^{n-1}\}$$

where $\alpha \in \mathbb{F}_2^n, \beta \in \{\mathbb{F}_2^n \setminus 0\}$ and $\alpha \cdot X, \beta \cdot f(X)$ are evaluated over \mathbb{F}_2 . The operation $\alpha \cdot X$ denotes the inner product of vectors α and input X of the S-box. $\beta \cdot f(X)$ denotes the product of vectors β and output $f(X)$ of the S-box.

By omitting the trivial case $\alpha = \beta = 0$, the LA can be determined through the maximum entry in linear approximation table of the S-box. The lower value of LA is preferable in resistance against linear cryptanalysis [3]. Usually, the value of LA must be less than 28 (i.e., $LA < 28$).

2.1.5. Algebraic Complexity (AIC)

The Lagrange interpolation polynomial is the polynomial $P(x)$ of degree $\leq (n - 1)$ that passes through n points $(x_1, y_1 = f(x_1)), (x_2, y_2 = f(x_2)), \dots, (x_n, y_n = f(x_n))$ and is given by:

$$P(x) = \sum_{j=1}^n P_j(x),$$

where

$$P_j(x) = y_j \prod_{k=1, k \neq j}^n \frac{x - x_k}{x_j - x_k}$$

The number of terms in $P_j(x)$ determines whether an S-box has a simple or complex algebraic expression, thereby classifying whether the S-box has a low or high algebraic complexity. A complex AIC is needed to resist an interpolation attack [6] and other concerning algebraic attacks [7].

2.1.6. Fixed Points (Fp) and Opposite Fixed Points (OFp)

For an S-box, a fixed point is defined as $f(x) = x$. This mean if an input x is given, the output is also x . An opposite fixed point is defined as $f(x) = \bar{x}$, where \bar{x} denotes the bit-wise complement of x . The number of Fp and OFp should be kept as low as possible to avoid any leakage in statistical cryptanalysis.

2.1.7. Balancedness

An n -variable Boolean function f is said to be balanced if $wt(f) = 2^{n-1}$, where $wt(f)$ is the Hamming weight of f for the n -variable. An S-box is called balanced if every value in \mathbb{F}_2^n occurs once. This implies that the function is bijective (also known as the permutation function).

2.2. S-Box Design Methods

In this section, we review several 8×8 S-box constructions available in literature and classify them into three generic methods which are random generation, evolutionary or heuristic, and a mathematical function. In general, the classification of each S-box proposal is based on its construction. As an example, random methods that use a pseudo-random generator to generate a key and/or select a key randomly from its construction elements are categorized as a random generation approach. Likewise, using mathematical functions as the core of S-box construction will be categorized as a mathematical function approach. The S-box constructions that use neither a random generator nor mathematical function can be categorized as heuristic approaches. In addition, we also classify S-box constructions that use more than one generic method as a *Combination Method*.

The random generation approach is the simplest technique to construct an S-box; however, its disadvantage is that most of the final S-boxes generated do not achieve our desired value for cryptographically strong S-boxes, as stated in Table 1. Conversely, the mathematical function approach is the best technique with which to achieve the desired S-box properties; however, it is difficult to find this function. For the heuristic approach, the latest S-box construction proposals show an improvement compared to the early proposal in terms of its S-box properties. Therefore, nowadays, researchers focus their attention on exploring heuristic approaches to generate cryptographically strong S-boxes since the implementation of this approach is convenient in both hardware and software.

To facilitate the classification of S-box construction, we divide the S-box design methods into two subsections. The first subsection reviews the S-box constructions which are implemented in the block ciphers proposal, while the second subsection reviews other S-box constructions available in literature.

2.2.1. S-Boxes in Block Ciphers

Mathematical Function

The block ciphers SQUARE [16], BKSQ [17] and Rijndael [18,19] were developed by the same designer in the late 1990s. All S-boxes in these three block ciphers were constructed using similar transformations which take the multiplicative inverse over finite field \mathbb{F}_{2^8} , as suggested in [20]; then, an affine transformation over \mathbb{F}_2 is applied to the output bits. Rijndael block cipher was then promoted as the Advanced Encryption Standard (AES) by the National Institute of Standards and Technology (NIST) in 2001 [8]. Following the same construction and transformations are the S-boxes used in block cipher Camellia [21], Mercy [22], ARIA [23], SMS4 [24] and BC2 [25]. The S-box in HyRAL [26] block cipher was also constructed based on multiplicative inverse over \mathbb{F}_{2^8} using the irreducible polynomial of $z^8 + z^4 + z^3 + z + 1$ with two transformations performed beforehand, namely affine transformation (i.e., $y = x \oplus 64$) and Gray code transformation (i.e., $z = y \oplus (y \ll 1)$). Therefore, HyRAL S-box is classified as a mathematical function approach.

Another block cipher proposed in 2010 is called PP-1 [27]. The S-box construction is classified as a mathematical function approach since they use a multiplicative inverse procedure over finite field, similar to AES but with a randomly chosen primitive polynomial (i.e., $z^7 + z^2 + z + 1$). In the studies of Fuller et al. [28] for removing linear redundancy in S-boxes, they select several random pairs of S-box elements and rearrange four corresponding S-box entries in such a way that the S-box produced is an involution S-box.

Heuristic

CS-Cipher [29] and CRYPTON [30] block cipher use a 3-round Feistel cipher to generate 8×8 S-box. In general, their S-box is constructed with concatenation of elements generated in \mathbb{F}_{2^4} and pre-determined parameters to produce 8×8 S-box. Using the same construction is the S-box used in PICARO [15] block cipher. However, the S-box generated in PICARO is a non-bijective S-box. The ZORRO [31] block cipher proposed an improvement to PICARO's S-box construction. They instantiate a 4-round Feistel network with a monomial x^3 generated in \mathbb{F}_{2^4} . Then, they add an 8-bit linear transformation at the

end of each round and lastly, an affine transformation to remove the fixed point. Unlike PICARO, the final S-box of ZORRO is bijective but the construction classification technique is the same, i.e., a heuristic approach.

S-box used in Skipjack [32] and FOX [33] block ciphers are also classified as a heuristic approach. Although the real S-box construction in Skipjack is unknown, the research of Biryukov and Perrin [34] on the reverse-engineering of S-boxes, concludes that the Skipjack's S-box construction is not the result of random generation. The method of S-box construction used in the FOX block cipher consists of a three-round Lai–Massey scheme to avoid a purely algebraic construction. The three rounds of the Lai–Massey scheme take three different small S-boxes as the round function F .

Random Generation

CHAIN [35] block ciphers generate their S-box using random generation while ANUBIS [36] block ciphers generate theirs using a pseudo-random number generator with consideration of the involution S-box as a result. An involution S-box is where the S-box has an inverse onto itself. Therefore, the same S-box can be used in encryption and decryption processes. The S-box used in KHAZAD [37] block ciphers is an improvement of ANUBIS's S-box that uses linear shuffling. The same linear shuffling for S-box construction is also adopted in the ICEBERG [38] block cipher.

KAMKAR [39] block ciphers use a key-dependent S-box. KAMKAR uses two different generators of a quasi-random sequence of numbers to alternately trigger based on the scanned cipher key. Then, the generators are used to generate two random numbers and derive two integers between 0 and 255 to be used for swapping the location from the initial S-box. In KAMKAR, the initial S-box is fixed with a linear power function (i.e., x over \mathbb{F}_{2^8}). The iteration continues until all bytes are scanned and swapped.

Combination Method

The Hierocrypt [40] block cipher uses two different S-boxes, namely a high-level S-box and lower-level S-box. For high-level S-box, the construction is based on bit permutation, followed by a power polynomial of x^{247} over \mathbb{F}_{2^8} using the irreducible polynomial of $z^8 + z^6 + z^5 + z + 1$. Lastly, an affine transformation over \mathbb{F}_2 is applied on the output. The construction for a lower-level S-box is not described; therefore, we conclude that they used the heuristic approach for lower-level S-box. Using the same approach, the CLEFIA [41] block cipher employs two different S-boxes constructed using inverse function over \mathbb{F}_{2^8} and using four 4-bit random S-boxes for the second construction. Therefore, Hierocrypt's and CLEFIA's S-box construction are combination methods because they use the mathematical function approach for the first S-box and heuristic approach for the second S-box.

The KALYNA [42] block cipher adopted the S-box construction proposed by Kazymyrov et al. [43], which was constructed based on gradient descent. This S-box construction is basically an enhancement of the improved hill climbing method proposed by Gao et al. [44] in 2010. Generally, they generate an initial solution, S , which is a bijective S-box with a minimum value of DU based on function F . The function F is basically determined as an inverse function over finite field \mathbb{F}_{2^8} . Then, they randomly swap a number of values in S to generate S_t as the final S-box. Therefore, we classify this construction as a combination method since they combine the mathematical function approach with the random generation approach.

2.2.2. Other S-Boxes Proposal

Aside from the S-boxes used in block ciphers, there exist a large number of S-box construction proposals in the literature. We have reviewed, analyzed and categorized all the S-box proposals that we have collected into random generation, heuristic, mathematical function and combination methods.

Random Generation

There are several S-box constructions that generate dynamic S-boxes which depend on a key. Among others are the proposal in [45–48]. In particular, their construction begins with an initial S-box, then a string of keys is generated to permute the content of the initial S-box. Thus, a new S-box is constructed. Various methods are used to generate the keys such as random generation, a pseudo-random number generator (PRNG) or a round key obtained from the RC4 algorithm [49].

Mamadolimov et al. [50] also proposed an S-box construction based on random generation. The initial S-box is represented as eight component functions. The first three component functions are deterministically constructed, such that the highest nonlinearity for each Boolean function is found. Then, the other five component functions are generated randomly to complete the S-box.

However, the S-box construction using random generation has received less attention among researchers. This might be because of the unfavorable cryptographic properties exhibited by this approach.

Heuristic

The heuristic approach has received more attention from researchers, especially on chaos-based S-box construction. This might be because chaos is a stochastic process in nonlinear dynamic systems, thus satisfying the need for nonlinearity in block ciphers. Among the popular chaotic maps used are the Chebyshev map [51,52], Baker map [53,54], Lorenz systems [55,56], logistic map [57,58], neural network [59,60], piecewise linear chaotic map [61,62], tent map [63,64] and Van Der Pol oscillator [65]. In general, chaos-based S-box construction starts with the selection of an initial value in selected chaotic map. Then, they iterate the chaotic map and use their defined technique to obtain an integer that lies between 0 and 255. This procedure is repeated to generate another integer until all 256 elements of the 8×8 S-box are fulfilled.

Furthermore, there are several S-box proposals that we categorized as heuristic approaches, such as the S-box construction using hill climbing [43,66], Latin square [67], analytical approach [68], genetic algorithm [69,70], simulated annealing [71,72], cellular automata [73,74], ant colony optimization [75], artificial immune system [76] and bee waggle dance [77]. Nature-inspired systems, such as genetic algorithms, work in reverse, i.e., [70], gradient descent [43] (i.e., modified hill climbing), artificial immune system [76] and bee waggle dance [77] show encouraging developments in S-box construction using the heuristic approach.

Mathematical Function

The main criterion for categorizing S-box proposals as mathematical functions is that designers use the same transformation as AES [8]. This includes either revising the transformation [78–80] itself by using different irreducible polynomials or by adding another affine transformations [81–84] in their construction.

Moreover, some S-box proposals using mathematical functions other than AES were found in the literature, such as linear fractional transformation or projective general linear group [85–87], trace-representation polynomial function [88], gray code encode [89], semi-fields pseudo-extensions [90] and using finite field power functions [91]. All these S-box constructions provide encouraging results and can be categorized among cryptographically strong S-boxes.

Combination Method

Fuller et al. proposed 2-step tweaking [28] and 4-step tweaking [92] in order to avoid linear redundancy and preserve the involution properties of an S-box, respectively. We categorized their proposed techniques as a combination method since they used methods such as divide and swap in the tweaking of the initial S-box that was generated based on a multiplicative inverse function over \mathbb{F}_{2^8} . Mamadolimov et al. [11] and Isa et al. [12,93]

also proposed an S-box construction using a combination method. In this proposal, they constructed an initial S-box based on the non-permutation power function followed by a heuristic approach, which is an algorithm to make the S-box bijective as a final result.

3. S-Box Construction

3.1. Preliminaries

Table 2 is an example of a non-permutation power function generated over finite field in \mathbb{F}_{2^4} . In this case, we use x^3 . We can extract the following information: (1) D_{EL} , (2) R_{EL} and (3) N_{EL} from Table 2.

Table 2. x^3 over \mathbb{F}_{2^4} .

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x^3	0	1	8	F	C	A	1	1	A	F	F	C	8	A	8	C

D_{EL} is a notation for *Desired Element*. A desired element only occurs once in a permutation function, such that a bijective S-box is achieved. From Table 2, only one element is found in the D_{EL} category, which is the element 0.

R_{EL} is a notation for *Repeated Elements*. The repeated elements were obtained from generated functions that occur more than once. From the analysis of Table 2, there are five elements that fall into this category, which are the elements 1, 8, A, C and F. All five repeated elements are found to be repeated three times.

N_{EL} denotes the *Non-existent Elements* which means that the elements are ‘missing’ from the generated function. This happens because the element that should exist has been replaced by a repeated element. From Table 2, we can determine that there are a total of 10 elements that do not exist in x^3 over \mathbb{F}_{2^4} , which are 2, 3, 4, 5, 6, 7, 9, B, D and E.

After these three group of elements are obtained, a *Redundancy Analysis Table* will be constructed. The *Redundancy Analysis Table* is a representation of bits error between all elements in R_{EL} compared to the elements in N_{EL} . Table 3 shows the redundancy analysis table of x^3 over the finite field \mathbb{F}_{2^4} . For example, there are 3 bits that needs to be changed to replace element 1 from R_{EL} with element 6 from N_{EL} , as illustrated in Figure 1. Likewise, the element F from R_{EL} has three options which only involve one bit change. The elements listed from N_{EL} are 7, B and E.

Table 3. Redundancy Analysis Table for x^3 over \mathbb{F}_{2^4} .

$R_{EL} \backslash N_{EL}$	2	3	4	5	6	7	9	B	D	E
1	2	1	2	1	3	2	1	2	3	2
8	2	3	2	3	3	4	1	2	1	2
A	1	2	3	4	2	3	2	1	2	3
C	3	4	1	2	2	3	2	3	0	1
F	3	2	3	2	2	1	2	1	2	1

3.2. Our Proposed Construction

Let x^d denote a power function in \mathbb{F}_{2^8} using the irreducible polynomial $x^8 + x^4 + x^3 + x^2 + 1$, where $d = \{1, 2, \dots, 2^8 - 2\}$ and $x \in \mathbb{F}_{2^8}$. All power functions (i.e., x^d) can be classified into linearly non-equivalent functions using the squaring method [94], as shown in Table 4. The first column of Table 4 represents the powers d that are non-equivalent to each other. The second column lists all the equivalent power functions for each power d . For instance, the power function x^{253} is equivalent to x^{127} , as is the case with x^{191} . Four more columns in Table 4 denote the values of NL, DU, AD and LA produced by the underlying power function. The N_{EL} column lists the number of non-existent elements, while the R_{EL} column lists the number of repeated elements in each of the involved power functions. The last column denotes whether the power function is a permutation or otherwise, based on the number given in columns N_{EL} and R_{EL} (i.e., label ‘Y’ is given when $N_{EL} = R_{EL} = 0$, otherwise label ‘N’ will be given).

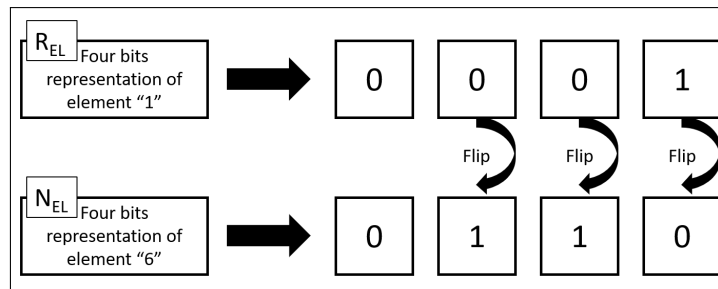


Figure 1. Bits Error Representation.

Table 4. Classification of linearly non-equivalent power function, x^d based on maximum NL in \mathbb{F}_{2^8} .

d	$\{d \times 2 \pmod{2^8 - 1}\}$	NL	DU	AD	LA	NEL	REL	PERM
127	254, 253, 251, 247, 239, 223, 191	112	4	7	16	0	0	Y
111	222, 246, 189, 123, 237, 219, 183	112	4	6	16	170	85	N
21	42, 84, 168, 162, 138, 81, 69	112	4	3	16	170	85	N
39	78, 156, 114, 228, 57, 201, 147	112	2	4	16	170	85	N
3	6, 12, 24, 48, 96, 192, 129	112	2	2	16	170	85	N
9	18, 36, 72, 144, 66, 132, 33	112	2	2	16	170	85	N
31	62, 124, 248, 241, 227, 199, 143	112	16	5	16	0	0	Y
91	182, 218, 214, 109, 181, 107, 173	112	16	5	16	0	0	Y
63	126, 252, 249, 243, 231, 207, 159	104	6	6	24	170	85	N
47	94, 188, 242, 121, 229, 203, 151	104	16	5	24	0	0	Y
19	38, 76, 152, 98, 196, 49, 137	104	16	3	24	0	0	Y
95	190, 250, 125, 245, 235, 215, 175	96	4	6	16	204	51	N
5	10, 20, 40, 80, 160, 130, 65	96	4	2	32	204	51	N
7	14, 28, 56, 112, 224, 193, 131	96	6	3	32	0	0	Y
37	74, 148, 82, 164, 146, 41, 73	96	6	3	32	0	0	Y
25	50, 100, 200, 70, 140, 145, 35	96	6	3	32	204	51	N
29	58, 116, 232, 142, 209, 163, 71	96	10	4	24	0	0	Y
11	22, 44, 88, 176, 194, 97, 133	96	10	3	24	0	0	Y
59	118, 236, 206, 217, 179, 103, 157	96	12	5	32	0	0	Y
55	110, 220, 230, 185, 115, 205, 155	96	12	5	32	204	51	N
13	26, 52, 104, 208, 134, 161, 67	96	12	3	32	0	0	Y
61	122, 244, 158, 233, 211, 167, 79	96	16	5	32	0	0	Y
23	46, 92, 184, 226, 113, 197, 139	96	16	4	32	0	0	Y
53	106, 212, 166, 154, 169, 83, 77	96	16	4	32	0	0	Y
27	54, 108, 216, 198, 177, 99, 141	80	26	4	48	170	85	N
87	174, 186, 234, 93, 117, 213, 171	80	30	5	48	170	85	N
43	86, 172, 178, 202, 89, 101, 149	80	30	4	48	0	0	Y
15	30, 60, 120, 240, 225, 195, 135	76	2	4	12	238	17	N
45	90, 180, 210, 150, 105, 165, 75	76	2	4	12	238	17	N
17	34, 68, 136	0	16	2	8	240	15	N
119	238, 221, 187	0	22	6	8	240	15	N
51	102, 204, 153	0	24	4	10	250	5	N
85	170	0	60	4	6	252	3	N
1	2, 4, 8, 16, 32, 64, 128	0	256	1	128	0	0	Y

Our construction starts with the generation of binomial power functions. F_1 and F_2 which are two different power functions selected over finite field \mathbb{F}_{2^8} is added to produce a new function F (i.e., $F = F_1 + F_2$). In total, there are $C_2^{2^8-2} = 32,131$ possible combination of binomial power functions produced from this operation and none of them are permutation function. We analyzed its repeated elements (R_{EL}) and non-existent elements (N_{EL}), then summarized the findings in Table 5.

Table 5. Redundancy Analysis of Binomial Power Functions

	N_{EL}	R_{EL}	Total		N_{EL}	R_{EL}	Total		N_{EL}	R_{EL}	Total		N_{EL}	R_{EL}	Total		N_{EL}	R_{EL}	Total
1	15	1	1024	27	85	35	256	53	85	55	256	79	77	69	128	105	96	80	256
2	17	1	128	28	221	35	64	54	80	56	256	80	85	69	384	106	100	80	256
3	51	1	256	29	75	37	256	55	90	58	512	81	97	69	128	107	85	81	896
4	85	1	128	30	217	39	64	56	89	59	1024	82	187	69	192	108	125	81	256
5	16	2	128	31	45	41	128	57	88	60	256	83	78	70	128	109	175	81	512
6	254	2	1	32	81	41	512	58	92	60	384	84	92	70	768	110	104	82	256
7	253	3	2	33	85	41	256	59	75	61	256	85	94	70	128	111	105	83	256
8	252	4	4	34	214	42	64	60	99	61	768	86	96	70	128	112	97	85	256
9	251	5	4	35	85	43	512	61	195	61	64	87	93	71	256	113	105	85	128
10	40	6	128	36	213	43	256	62	84	62	128	88	185	71	64	114	171	85	256
11	250	6	4	37	50	46	128	63	85	63	256	89	84	72	128	115	90	86	128
12	248	8	12	38	82	46	256	64	89	63	768	90	92	72	256	116	102	86	128
13	247	9	8	39	90	46	512	65	193	63	64	91	75	73	640	117	120	86	128
14	246	10	8	40	210	46	64	66	84	64	128	92	93	73	256	118	170	86	64
15	245	11	32	41	209	47	64	67	88	64	384	93	97	73	256	119	96	88	256
16	244	12	8	42	75	49	128	68	96	64	256	94	99	73	640	120	104	88	256
17	243	13	32	43	99	49	256	69	192	64	128	95	123	73	256	121	102	90	256
18	80	16	128	44	207	49	256	70	89	65	640	96	183	73	320	122	160	96	64
19	240	16	96	45	206	50	128	71	191	65	128	97	92	74	256	123	99	97	128
20	239	17	112	46	85	51	256	72	94	66	256	98	93	75	128	124	100	98	128
21	68	18	128	47	125	51	128	73	85	67	256	99	181	75	320	125	108	100	128
22	238	18	16	48	205	51	224	74	93	67	256	100	84	76	256	126	144	112	64
23	64	22	256	49	72	52	256	75	97	67	256	101	108	76	256	127	113	113	128
24	51	25	512	50	84	52	640	76	189	67	64	102	179	77	1024	128	120	120	128
25	230	26	16	51	204	52	64	77	84	68	384	103	178	78	64	129	136	120	192
26	69	31	512	52	77	53	256	78	92	68	768	104	177	79	128	130	128	128	576
Overall Total																			32,131

Table 5 lists the number of binomial power functions according to (N_{EL}, R_{EL}) pairs. As an example, there are a total of 1024 binomial power functions that produce 15 non-existent elements over only one repeated element. In addition, there are a total of 576 generated binomial power functions that produce the same number of N_{EL} and R_{EL} , which are equal to 128. Other than that, we can categorize the generated binomial power functions into 130 groups based on (N_{EL}, R_{EL}) pairs.

Overall, our proposed algorithm in constructing cryptographically strong S-boxes is illustrated in Figure 2, which is inspired by the S-box construction of Isa et al. [12]. Our initial S-box is selected from the groups of binomial power functions that produce only one (1) repeated element. This includes (N_{EL}, R_{EL}) pairs of (15, 1), (17, 1), (51, 1) and (85, 1) from Table 5.

In total, there are 1540 candidates that can be used for our initial S-box. However, this number of candidates can be minimized to 200 using the squaring method [94] because of its linearly non-equivalent properties, as shown in Table 6. The complete list of linearly non-equivalent binomial power functions that produces one repeated element from these four groups is included in Appendix A. In this study, we select and limit the selection of the initial S-box to only one repeated element to preserve the cryptographic properties of the newly generated function such that it is not compromised significantly.

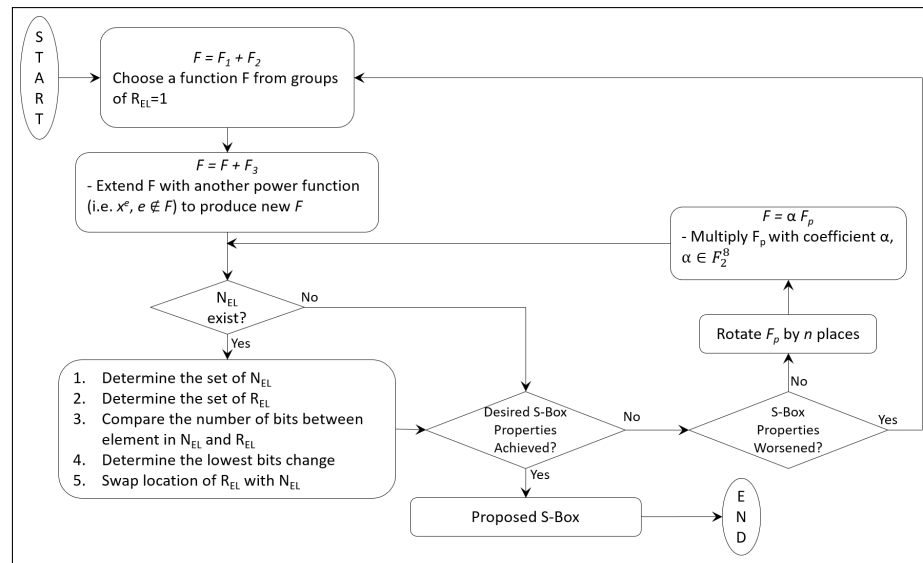


Figure 2. S-box Optimization Algorithm.

Table 6. Linearly Non-Equivalent Functions for $R_{EL} = 1$.

	N_{EL}	R_{EL}	Distinct Total	Linearly Non-Equivalent Total
1	15	1	1024	136
2	17	1	128	16
3	51	1	256	32
4	85	1	128	16
	Overall Total		1540	200

The construction continues with the addition of an S-box candidate with another power function (i.e., $F = F + F_3$), where the selection of a new power function must not include the previously selected power function. After a new function F is generated, we extract the information of D_{EL} , R_{EL} and N_{EL} on the function, as described in Section 3.1. If N_{EL} contains no element, that means the function is already bijective. Then, we measure its cryptographic properties and store it as the proposed S-box if it exhibits the desired strong properties.

Otherwise, if N_{EL} contains any element, we construct a redundancy analysis table by determining all the elements in N_{EL} and R_{EL} groups of F . Then, the bit errors rate between all elements in N_{EL} and all elements in R_{EL} are produced to complete the redundancy analysis table. The bit error rate is the number of bit errors per unit time. In this study, the bit errors range from a minimum of one (1) bit, up to the maximum of eight (8) bits since we construct an 8×8 S-box.

From the redundancy analysis table, we identify one current ‘smallest’ bit error. If there are several choices obtained from the table, the selection will be made randomly. Then, we determine the actual element of N_{EL} and R_{EL} that correspond with the selected smallest bit error. Next, the exact location of the selected R_{EL} is identified in function F and lastly we swap the element with the previously selected N_{EL} (i.e., $F(R_{EL}) = N_{EL}$). Similarly, the selection will be made randomly if more than one option of the identified R_{EL} is located. By selecting the smallest bit errors, we expect that the cryptographic properties of the newly generated function will not be compromised significantly.

After the swapping process, we measure the cryptographic properties of this newly generated function. We store it as the proposed S-box if it exhibits the desired strong properties. Otherwise, the function will be denoted as F_p and rotated by n places. This rotation step that we proposed is operated based on circular rotation either column-wise (i.e., rotation of component function) or row-wise (i.e., rotation of elements in the function). The purpose of this step is to increase the algebraic expression (i.e., no. of terms (#term)) of the function F_p while preserving its cryptographic properties. Current construction, which

exhibits the algebraic expression with simple terms, might expose ciphers using the S-box to an interpolation attack [6]. From our observation, column-wise rotation does not impact the cryptographic properties of the function. However, different to row-wise rotation, the cryptographic properties might change significantly. Our analysis found that, for row-wise rotations of 64, 128 and 192 places, the cryptographic properties of the S-box can be preserved.

Then, another round of iteration will be conducted in our algorithm. The next iteration contains a multiplication step of F_p with a coefficient α where α ranges from 0 to 255; it then continues again with a redundancy analysis table. The stopping criteria that we fix in our iterations are based on the final S-box produced that meets our requirement or based on the cryptographic properties of F_p that are found to worsen (i.e., the cryptographic properties exhibited are lower than the values stated in Table 1). In both cases, another set of $F = F_1 + F_2$ will be selected for our next experiments.

Table 7 shows the optimal S-box obtained from the proposed method and is represented as hexadecimal. The first column in Table 7 denotes the first four bits, while the first row denotes the remaining four bits of the 8-bit input to the S-box. For instance, the input 71 gives the output CE, (i.e., $F(71) = CE$). For input 71, element 7 is selected from the first column while element 1 is selected from the first row). The cryptographic properties exhibited by this S-box meet all the desired values stated in Table 1.

Table 7. Optimal S-box using Proposed Technique.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8B	C3	6E	5E	9B	E9	03	68	BE	BD	12	27	2C	F4	BB	41
1	31	A3	8A	86	C8	38	94	3C	08	32	3E	B5	56	ED	14	47
2	37	1B	7B	DE	35	99	1F	A0	81	65	8D	5B	DC	E3	79	9C
3	64	E5	90	06	D5	3A	AE	B4	DF	71	6D	16	A9	6C	25	BA
4	CF	9D	30	D3	EC	A5	D8	B0	05	C1	6F	0E	7F	D4	91	62
5	B9	E1	61	C9	D6	C6	43	E2	88	52	0B	87	4E	17	EF	CC
6	60	2D	3F	78	2E	24	45	CB	D1	C4	FC	1C	22	FE	4D	6B
7	B6	CE	07	6A	66	58	F8	0F	70	67	F5	98	74	D0	FD	82
8	3B	FB	B3	9A	EE	75	F2	77	76	34	63	E7	29	10	36	44
9	4A	89	BC	DB	18	2F	DA	FA	19	AB	EB	7A	13	72	50	D2
A	9F	4F	96	F3	7C	CA	7D	C0	09	D7	B1	C5	57	73	48	04
B	A6	5A	11	A7	93	97	1A	0D	26	0A	C7	69	F0	3D	39	8E
C	00	20	A4	95	AC	E6	40	59	23	92	A8	33	49	02	F1	AA
D	15	8F	5F	1E	80	53	4B	42	F6	7E	2B	01	9E	E4	CD	AF
E	5C	2A	55	F7	8C	28	0C	E8	B8	5D	4C	E0	84	21	FF	A1
F	B7	A2	B2	BF	F9	DD	54	83	1D	EA	C2	85	51	AD	46	D9
NL = 108				DU = 4				AD = 7				LA = 20				
(Fp/OFp) = (1/0)				#term = 255				AIC = Complex				PERM = Y				

4. Discussion

Table 8 compares our obtained S-box with the existing 8×8 cryptographically strong S-boxes available in literature, as discussed in the previous section. Only S-boxes that satisfy our prerequisite requirements are selected for comparison. The S-boxes were then arranged based on the optimal cryptographic properties exhibited by each S-box, starting with the highest NL, lowest DU, highest AD and, lastly, lowest LA. From there, we rank the S-boxes based on these four properties. Also included in Table 8 are the number of terms exhibited by an S-box and lastly the generic design method of each proposal.

In total, there are 39 S-boxes listed in Table 8 where we classify 16 ranks. The first rank is occupied by the AES S-box which was constructed using multiplicative inversion in \mathbb{F}_{2^8} . There are 13 others S-box construction that are similar to AES's S-box construction; thus, they exhibit the same cryptographic properties as AES, which consists of (112, 4, 7, 16) for its (NL, DU, AD, LA), respectively. So far, this multiplicative inverse technique over finite field \mathbb{F}_{2^8} leads to the best-known cryptographic properties for an 8×8 S-box.

Ranked second is the S-box construction proposed by Li et al. [91]. Their S-box construction exhibits cryptographic properties of (112, 4, 5, 16) for (NL, DU, AD, LA),

respectively. Their S-box proposal is constructed in \mathbb{F}_{2^9} using Quadratic APN functions which are heuristically converted into an 8×8 S-box as the final result.

There are also some authors that proposed several S-boxes such as Yang et al. [80], Kazymyrov et al. [43], Ivanov et al. [70] and Isa et al. in [12,77,93]. The S-box constructions proposed by Yang et al. [80] are ranked third, fourth and fifth and classified as mathematical functions-generated methods, as multiplicative inverse, addition and multiplication are only used in the algorithm. Ranked third and fifth are the S-boxes proposed by Ivanov et al. [70] that use the reversed genetic algorithm on the initial S-box that was generated beforehand using multiplicative inverse. Thus, we classify this algorithm as a combination method that lies between the mathematical function approach and heuristic approach. The third ranked S-box has the same nonlinearity as AES (i.e., $NL = 112$); however, the value for differential uniformity is slightly higher which is $DU = 6$. By the same S-box construction, Ivanov et al. [70] also produced S-boxes that are ranked seventh, ninth and tenth in Table 8.

We classify the S-box construction of Kazymyrov et al. [43] as random generation. This is because the S-box is constructed by randomly swapping two elements in an initial function. This step is repeated until their desired criteria are achieved. Their best proposed S-box is ranked eleventh with cryptographic properties of (104, 8, 7, 24), while the other one is ranked thirteenth.

Our proposed S-box construction is ranked sixth. The final S-box that we obtained exhibits cryptographic properties of (108, 4, 7, 20). The S-box also produces a complex algebraic expression of 255 terms. The best S-box by Isa et al. [93] shares the same rank of sixth, while the other two proposed S-boxes are ranked eighth and ninth with cryptographic properties of (108, 6, 4, 20) and (106, 6, 7, 22) for (NL, DU, AD, LA), respectively. However, their S-box, which can be constructed using simple algebraic expression (i.e., only three terms involved), might be vulnerable to an interpolation attack [6].

There are other S-boxes proposed by Isa et al. in [12,77] using the combination method. Ranked seventh, Isa et al. [77] used an algorithm called *Bee Waggle Dance* on the multiplicative inverse function. This construction produced an S-box with cryptographic properties of (108, 6, 7, 20). Ranked tenth is the construction of Isa et al. [12], who used an algorithm called the *Redundancy Removal Algorithm* on the non-permutation power function that exhibits cryptographic properties of (104, 6, 7, 24). Mamadolimov et al. [11] first introduced the original version of the *Redundancy Removal Algorithm*, which is ranked twelfth and has cryptographic properties of (102, 8, 7, 26). Empirically, our proposed construction, which is an inspired and enhanced version of the *Redundancy Removal Algorithm*, managed to obtain cryptographically strong S-boxes that compare well with and even outperform the originally proposed constructions [11,12].

From our observation, all the S-boxes listed in Table 8 are designed and constructed either through mathematical functions or through combination methods between mathematical functions and heuristic methods. So far, the best S-box that we found using random generation is the S-box proposed by Kazymyrov et al. [43], which is ranked eleventh. Nevertheless, this S-box exhibits $DU = 8$; thus, it fails to fulfill our cryptographically strong criteria. In fact, the S-boxes ranked from eleventh to sixteenth all fail to fulfill our prerequisite requirement for cryptographically strong S-boxes. The S-boxes are included in Table 8 for a comparative analysis.

The research results that are included in the comparative analysis are obtained by applying the security measurement using a mini-workstation with properties of Intel (R) Core (TM) i7-6600U CPU @ 2.60 GHz, 16 GB of RAM, 64-bit Operating System and pre-installed with MATLAB R2022a. All simulations were performed in MATLAB programming language, and no external embedded devices were involved. Since MATLAB is a high-level programming language, we did not conduct a computational complexity analysis of our S-box construction.

Table 8. S-Boxes Ranks.

Rank	S-Box	NL	DU	AD	LA	#Term	Design Method
1	AES [8]	112	4	7	16	9	Mathematical
	Hierocrypt [40]	112	4	7	16	255	
	Camellia [21]	112	4	7	16	254	
	ARIA [23]	112	4	7	16	9	
	CLEFIA [41]	112	4	7	16	254	
	HyRAL [26]	112	4	7	16	253	
	Yang et al. [80]	112	4	7	16	1	
	Cui et al. [82]	112	4	7	16	253	
	Hussain et al. [57]	112	4	7	16	254	
	Dumas and Orfila [90]	112	4	7	16	254	
	Gondal et al. [54]	112	4	7	16	251	
	Khan and Azam [79]	112	4	7	16	252	
Tran et al. [89]	112	4	7	16	254		
Kapalova et al. [84]	112	4	7	16	255		
2	Li et al. [91]	112	4	5	16	217	Mathematical + Heuristic
3	Yang et al. [80]	112	6	7	16	250	Mathematical
	Ivanov et al. [70]	112	6	7	16	253	Mathematical + Heuristic
4	Yang et al. [80]	110	4	7	18	253	Mathematical
5	Yang et al. [80]	110	6	7	18	253	Mathematical
	Ivanov et al. [70]	110	6	7	18	254	Mathematical + Heuristic
6	This Paper	108	4	7	20	255	Mathematical + Heuristic
	Isa et al. [93]	108	4	7	20	3	Mathematical
7	Ivanov et al. [70]	108	6	7	20	253	Mathematical + Heuristic
	Isa et al. [77]	108	6	7	20	253	
8	Isa et al. [93]	108	6	4	20	3	Mathematical
9	Ivanov et al. [70]	106	6	7	22	252	Mathematical + Heuristic
	Fuller et al. [28,92]	106	6	7	22	254	
	Hierocrypt [40]	106	6	7	22	253	
	Isa et al. [93]	106	6	7	22	3	
10	Isa et al. [12]	104	6	7	24	255	Mathematical + Heuristic
	Ivanov et al. [70]	104	6	7	24	255	
11	KALYNA [42]	104	8	7	24	254	Heuristic
	Kazymyrov et al. [43]	104	8	7	24	254	Random
12	Mamadolimov et al. [11]	102	8	7	26	254	Mathematical + Heuristic
13	Kazymyrov et al. [43]	100	8	7	28	254	Random
14	Picek et al. [69]	100	10	7	28	254	Heuristic
15	CLEFIA [41]	100	10	6	28	246	Mathematical
16	Picek et al. [69]	100	12	7	28	253	Heuristic
	SKIPJACK [32]	100	12	7	28	255	

5. Conclusions

In this paper, we analyzed the cryptographic properties of binomial power functions. Through our observation of the redundancy table, we discover that all binomial power functions generated over a finite field \mathbb{F}_{2^8} are not bijective. We modify and enhance the method called the *Redundancy Removal Algorithm* to obtain cryptographically strong S-boxes. Our proposed S-box exhibits cryptographic properties of (108, 4, 7, 20) for (NL, DU, AD, LA), respectively; thus all our prerequisite requirements are fulfilled and the performance of the original algorithm proposed by Mamadolimov et al. and Isa et al. is surpassed.

From our comparative analysis, the mathematical function remains the best method to construct a cryptographically strong S-box. The production of a new mathematical function that can challenge multiplicative inversion is an interest of many researchers. However, recent research developments on combination methods of mathematical functions and the heuristic approach show an encouraging trend in the construction of a cryptographically

strong S-box. As for random generation, this method is unfavorable since there are no recent developments driven by this approach.

In future work, our proposed S-box can be applied to a working encryption algorithm to analyze its efficiency and accuracy. To achieve this, we have to convert our script into a low-level programming language and implement it in an embedded device. We are also interested in categorizing our S-box collections based on equivalence classes. To the best of our knowledge, there are three types of equivalence classes in literature which are linear and affine equivalence, extended affine (EA-) equivalent and Carlet–Charpin–Zinoviev (CCZ-) equivalent.

Author Contributions: Conceptualization, S.A.A.S.J. and N.A.; methodology, H.I., S.M.A. and M.R.Z.; software, R.E. and N.A.; validation, H.I.; formal analysis, H.I.; investigation, H.I.; resources, H.I., S.A.A.S.J., R.E. and N.A.; data curation, H.I.; writing—original draft preparation, H.I.; writing—review and editing, M.R.Z., S.A.A.S.J., R.E., S.M.A. and N.A.; supervision, M.R.Z. and S.A.A.S.J. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge support from the Long Term Research Grant Scheme (LRGS) under grant number LRGS/1/2020/UM/01/5/4 (9012-00010) from the Ministry of Higher Education, Malaysia. R and N would also like to acknowledge support from Geran Penyelidikan Pengkomersialan MTUN 2020 9028-00023 provided by the Ministry of Higher Education of Malaysia (MOHE).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

- SPNs Substitution-Permutation Networks
- S-box Substitution box
- NL Nonlinearity
- DU Differential Uniformity
- AD Algebraic Degree
- AES Advanced Encryption Standard
- LA Linear Approximation
- MDPI Multidisciplinary Digital Publishing Institute
- DEL Desired Element
- REL Repeated Elements
- NEL Non-Existent Elements

Appendix A

Appendix A.1

Table A1. Classification of Linearly Non-Equivalent Binomial Power Function with 15 to 1 Output.

A	Linearly Equivalent Binomial Power Function to A
$x^1 + x^{31}$	$x^2 + x^{62}, x^4 + x^{124}, x^8 + x^{248}, x^{16} + x^{241}, x^{32} + x^{227}, x^{64} + x^{199}, x^{128} + x^{143}$
$x^1 + x^{46}$	$x^2 + x^{92}, x^4 + x^{184}, x^8 + x^{113}, x^{16} + x^{226}, x^{32} + x^{197}, x^{64} + x^{139}, x^{128} + x^{23}$
$x^1 + x^{61}$	$x^2 + x^{122}, x^4 + x^{244}, x^8 + x^{233}, x^{16} + x^{211}, x^{32} + x^{167}, x^{64} + x^{79}, x^{128} + x^{158}$
$x^1 + x^{76}$	$x^2 + x^{152}, x^4 + x^{49}, x^8 + x^{98}, x^{16} + x^{196}, x^{32} + x^{137}, x^{64} + x^{19}, x^{128} + x^{38}$
$x^1 + x^{91}$	$x^2 + x^{182}, x^4 + x^{109}, x^8 + x^{218}, x^{16} + x^{181}, x^{32} + x^{107}, x^{64} + x^{214}, x^{128} + x^{173}$
$x^1 + x^{106}$	$x^2 + x^{212}, x^4 + x^{169}, x^8 + x^{83}, x^{16} + x^{166}, x^{32} + x^{77}, x^{64} + x^{154}, x^{128} + x^{53}$
$x^1 + x^{121}$	$x^2 + x^{242}, x^4 + x^{229}, x^8 + x^{203}, x^{16} + x^{151}, x^{32} + x^{47}, x^{64} + x^{94}, x^{128} + x^{188}$
$x^1 + x^{136}$	$x^2 + x^{17}, x^4 + x^{34}, x^8 + x^{68}, x^{16} + x^{136}, x^{32} + x^{17}, x^{64} + x^{34}, x^{128} + x^{68}$
$x^1 + x^{151}$	$x^2 + x^{47}, x^4 + x^{94}, x^8 + x^{188}, x^{16} + x^{121}, x^{32} + x^{242}, x^{64} + x^{229}, x^{128} + x^{203}$
$x^1 + x^{166}$	$x^2 + x^{77}, x^4 + x^{154}, x^8 + x^{53}, x^{16} + x^{106}, x^{32} + x^{212}, x^{64} + x^{169}, x^{128} + x^{83}$
$x^1 + x^{181}$	$x^2 + x^{107}, x^4 + x^{214}, x^8 + x^{173}, x^{16} + x^{91}, x^{32} + x^{182}, x^{64} + x^{109}, x^{128} + x^{218}$

Table A1. Cont.

A	Linearly Equivalent Binomial Power Function to A
$x^1 + x^{196}$	$x^2 + x^{137}, x^4 + x^{19}, x^8 + x^{38}, x^{16} + x^{76}, x^{32} + x^{152}, x^{64} + x^{49}, x^{128} + x^{98}$
$x^1 + x^{211}$	$x^2 + x^{167}, x^4 + x^{79}, x^8 + x^{158}, x^{16} + x^{61}, x^{32} + x^{122}, x^{64} + x^{244}, x^{128} + x^{233}$
$x^1 + x^{226}$	$x^2 + x^{197}, x^4 + x^{139}, x^8 + x^{23}, x^{16} + x^{46}, x^{32} + x^{92}, x^{64} + x^{184}, x^{128} + x^{113}$
$x^1 + x^{241}$	$x^2 + x^{227}, x^4 + x^{199}, x^8 + x^{143}, x^{16} + x^{31}, x^{32} + x^{62}, x^{64} + x^{124}, x^{128} + x^{248}$
$x^7 + x^{22}$	$x^{14} + x^{44}, x^{28} + x^{88}, x^{56} + x^{176}, x^{112} + x^{97}, x^{224} + x^{194}, x^{193} + x^{133}, x^{131} + x^{11}$
$x^7 + x^{37}$	$x^{14} + x^{74}, x^{28} + x^{148}, x^{56} + x^{41}, x^{112} + x^{82}, x^{224} + x^{164}, x^{193} + x^{73}, x^{131} + x^{146}$
$x^7 + x^{52}$	$x^{14} + x^{104}, x^{28} + x^{208}, x^{56} + x^{161}, x^{112} + x^{67}, x^{224} + x^{134}, x^{193} + x^{13}, x^{131} + x^{26}$
$x^7 + x^{67}$	$x^{14} + x^{134}, x^{28} + x^{13}, x^{56} + x^{26}, x^{112} + x^{52}, x^{224} + x^{104}, x^{193} + x^{208}, x^{131} + x^{161}$
$x^7 + x^{82}$	$x^{14} + x^{164}, x^{28} + x^{73}, x^{56} + x^{146}, x^{112} + x^{37}, x^{224} + x^{74}, x^{193} + x^{148}, x^{131} + x^{41}$
$x^7 + x^{97}$	$x^{14} + x^{194}, x^{28} + x^{133}, x^{56} + x^{11}, x^{112} + x^{22}, x^{224} + x^{44}, x^{193} + x^{88}, x^{131} + x^{176}$
$x^7 + x^{127}$	$x^{14} + x^{254}, x^{28} + x^{253}, x^{56} + x^{251}, x^{112} + x^{247}, x^{224} + x^{239}, x^{193} + x^{223}, x^{131} + x^{191}$
$x^7 + x^{142}$	$x^{14} + x^{29}, x^{28} + x^{58}, x^{56} + x^{116}, x^{112} + x^{232}, x^{224} + x^{209}, x^{193} + x^{163}, x^{131} + x^{71}$
$x^7 + x^{157}$	$x^{14} + x^{59}, x^{28} + x^{118}, x^{56} + x^{236}, x^{112} + x^{217}, x^{224} + x^{179}, x^{193} + x^{103}, x^{131} + x^{206}$
$x^7 + x^{172}$	$x^{14} + x^{89}, x^{28} + x^{178}, x^{56} + x^{101}, x^{112} + x^{202}, x^{224} + x^{149}, x^{193} + x^{43}, x^{131} + x^{86}$
$x^7 + x^{187}$	$x^{14} + x^{119}, x^{28} + x^{238}, x^{56} + x^{221}, x^{112} + x^{187}, x^{224} + x^{119}, x^{193} + x^{238}, x^{131} + x^{221}$
$x^7 + x^{202}$	$x^{14} + x^{149}, x^{28} + x^{43}, x^{56} + x^{86}, x^{112} + x^{172}, x^{224} + x^{89}, x^{193} + x^{178}, x^{131} + x^{101}$
$x^7 + x^{217}$	$x^{14} + x^{179}, x^{28} + x^{103}, x^{56} + x^{206}, x^{112} + x^{157}, x^{224} + x^{59}, x^{193} + x^{118}, x^{131} + x^{236}$
$x^7 + x^{232}$	$x^{14} + x^{209}, x^{28} + x^{163}, x^{56} + x^{71}, x^{112} + x^{142}, x^{224} + x^{29}, x^{193} + x^{58}, x^{131} + x^{116}$
$x^7 + x^{247}$	$x^{14} + x^{239}, x^{28} + x^{223}, x^{56} + x^{191}, x^{112} + x^{127}, x^{224} + x^{254}, x^{193} + x^{253}, x^{131} + x^{251}$
$x^{11} + x^{26}$	$x^{22} + x^{52}, x^{44} + x^{104}, x^{88} + x^{208}, x^{176} + x^{161}, x^{97} + x^{67}, x^{194} + x^{134}, x^{133} + x^{13}$
$x^{11} + x^{41}$	$x^{22} + x^{82}, x^{44} + x^{164}, x^{88} + x^{73}, x^{176} + x^{146}, x^{97} + x^{37}, x^{194} + x^{74}, x^{133} + x^{148}$
$x^{11} + x^{71}$	$x^{22} + x^{142}, x^{44} + x^{29}, x^{88} + x^{58}, x^{176} + x^{116}, x^{97} + x^{232}, x^{194} + x^{209}, x^{133} + x^{163}$
$x^{11} + x^{86}$	$x^{22} + x^{172}, x^{44} + x^{89}, x^{88} + x^{178}, x^{176} + x^{101}, x^{97} + x^{202}, x^{194} + x^{149}, x^{133} + x^{43}$
$x^{11} + x^{101}$	$x^{22} + x^{202}, x^{44} + x^{149}, x^{88} + x^{43}, x^{176} + x^{86}, x^{97} + x^{172}, x^{194} + x^{89}, x^{133} + x^{178}$
$x^{11} + x^{116}$	$x^{22} + x^{232}, x^{44} + x^{209}, x^{88} + x^{163}, x^{176} + x^{71}, x^{97} + x^{142}, x^{194} + x^{29}, x^{133} + x^{58}$
$x^{11} + x^{146}$	$x^{22} + x^{37}, x^{44} + x^{74}, x^{88} + x^{148}, x^{176} + x^{41}, x^{97} + x^{82}, x^{194} + x^{164}, x^{133} + x^{73}$
$x^{11} + x^{161}$	$x^{22} + x^{67}, x^{44} + x^{134}, x^{88} + x^{13}, x^{176} + x^{26}, x^{97} + x^{52}, x^{194} + x^{104}, x^{133} + x^{208}$
$x^{11} + x^{191}$	$x^{22} + x^{127}, x^{44} + x^{254}, x^{88} + x^{253}, x^{176} + x^{251}, x^{97} + x^{247}, x^{194} + x^{239}, x^{133} + x^{223}$
$x^{11} + x^{206}$	$x^{22} + x^{157}, x^{44} + x^{59}, x^{88} + x^{118}, x^{176} + x^{236}, x^{97} + x^{217}, x^{194} + x^{179}, x^{133} + x^{103}$
$x^{11} + x^{221}$	$x^{22} + x^{187}, x^{44} + x^{119}, x^{88} + x^{238}, x^{176} + x^{221}, x^{97} + x^{187}, x^{194} + x^{119}, x^{133} + x^{238}$
$x^{11} + x^{236}$	$x^{22} + x^{217}, x^{44} + x^{179}, x^{88} + x^{103}, x^{176} + x^{206}, x^{97} + x^{157}, x^{194} + x^{59}, x^{133} + x^{118}$
$x^{11} + x^{251}$	$x^{22} + x^{247}, x^{44} + x^{239}, x^{88} + x^{223}, x^{176} + x^{191}, x^{97} + x^{127}, x^{194} + x^{254}, x^{133} + x^{253}$
$x^{13} + x^{43}$	$x^{26} + x^{86}, x^{52} + x^{172}, x^{104} + x^{89}, x^{208} + x^{178}, x^{161} + x^{101}, x^{67} + x^{202}, x^{134} + x^{149}$
$x^{13} + x^{58}$	$x^{26} + x^{116}, x^{52} + x^{232}, x^{104} + x^{209}, x^{208} + x^{163}, x^{161} + x^{71}, x^{67} + x^{142}, x^{134} + x^{29}$
$x^{13} + x^{73}$	$x^{26} + x^{146}, x^{52} + x^{37}, x^{104} + x^{74}, x^{208} + x^{148}, x^{161} + x^{41}, x^{67} + x^{82}, x^{134} + x^{164}$
$x^{13} + x^{103}$	$x^{26} + x^{206}, x^{52} + x^{157}, x^{104} + x^{59}, x^{208} + x^{118}, x^{161} + x^{236}, x^{67} + x^{217}, x^{134} + x^{179}$
$x^{13} + x^{118}$	$x^{26} + x^{236}, x^{52} + x^{217}, x^{104} + x^{179}, x^{208} + x^{103}, x^{161} + x^{206}, x^{67} + x^{157}, x^{134} + x^{59}$
$x^{13} + x^{148}$	$x^{26} + x^{41}, x^{52} + x^{82}, x^{104} + x^{164}, x^{208} + x^{73}, x^{161} + x^{146}, x^{67} + x^{37}, x^{134} + x^{74}$
$x^{13} + x^{163}$	$x^{26} + x^{71}, x^{52} + x^{142}, x^{104} + x^{29}, x^{208} + x^{58}, x^{161} + x^{116}, x^{67} + x^{232}, x^{134} + x^{209}$
$x^{13} + x^{178}$	$x^{26} + x^{101}, x^{52} + x^{202}, x^{104} + x^{149}, x^{208} + x^{43}, x^{161} + x^{86}, x^{67} + x^{172}, x^{134} + x^{89}$
$x^{13} + x^{223}$	$x^{26} + x^{191}, x^{52} + x^{127}, x^{104} + x^{254}, x^{208} + x^{253}, x^{161} + x^{251}, x^{67} + x^{247}, x^{134} + x^{239}$
$x^{13} + x^{238}$	$x^{26} + x^{221}, x^{52} + x^{187}, x^{104} + x^{119}, x^{208} + x^{238}, x^{161} + x^{221}, x^{67} + x^{187}, x^{134} + x^{119}$
$x^{13} + x^{253}$	$x^{26} + x^{251}, x^{52} + x^{247}, x^{104} + x^{239}, x^{208} + x^{223}, x^{161} + x^{191}, x^{67} + x^{127}, x^{134} + x^{254}$
$x^{17} + x^{47}$	$x^{34} + x^{94}, x^{68} + x^{188}, x^{136} + x^{121}$
$x^{17} + x^{62}$	$x^{34} + x^{124}, x^{68} + x^{248}, x^{136} + x^{241}$
$x^{17} + x^{77}$	$x^{34} + x^{154}, x^{68} + x^{53}, x^{136} + x^{106}$
$x^{17} + x^{92}$	$x^{34} + x^{184}, x^{68} + x^{113}, x^{136} + x^{226}$
$x^{17} + x^{107}$	$x^{34} + x^{214}, x^{68} + x^{173}, x^{136} + x^{91}$
$x^{17} + x^{122}$	$x^{34} + x^{244}, x^{68} + x^{233}, x^{136} + x^{211}$
$x^{17} + x^{137}$	$x^{34} + x^{19}, x^{68} + x^{38}, x^{136} + x^{76}$
$x^{17} + x^{152}$	$x^{34} + x^{49}, x^{68} + x^{98}, x^{136} + x^{196}$
$x^{17} + x^{167}$	$x^{34} + x^{79}, x^{68} + x^{158}, x^{136} + x^{61}$
$x^{17} + x^{182}$	$x^{34} + x^{109}, x^{68} + x^{218}, x^{136} + x^{181}$
$x^{17} + x^{197}$	$x^{34} + x^{139}, x^{68} + x^{23}, x^{136} + x^{46}$
$x^{17} + x^{212}$	$x^{34} + x^{169}, x^{68} + x^{83}, x^{136} + x^{166}$
$x^{17} + x^{227}$	$x^{34} + x^{199}, x^{68} + x^{143}, x^{136} + x^{31}$
$x^{17} + x^{242}$	$x^{34} + x^{229}, x^{68} + x^{203}, x^{136} + x^{151}$
$x^{19} + x^{79}$	$x^{38} + x^{158}, x^{76} + x^{61}, x^{152} + x^{122}, x^{49} + x^{244}, x^{98} + x^{233}, x^{196} + x^{211}, x^{137} + x^{167}$
$x^{19} + x^{94}$	$x^{38} + x^{188}, x^{76} + x^{121}, x^{152} + x^{242}, x^{49} + x^{229}, x^{98} + x^{203}, x^{196} + x^{151}, x^{137} + x^{47}$
$x^{19} + x^{109}$	$x^{38} + x^{218}, x^{76} + x^{181}, x^{152} + x^{107}, x^{49} + x^{214}, x^{98} + x^{173}, x^{196} + x^{91}, x^{137} + x^{182}$
$x^{19} + x^{124}$	$x^{38} + x^{248}, x^{76} + x^{241}, x^{152} + x^{227}, x^{49} + x^{199}, x^{98} + x^{143}, x^{196} + x^{31}, x^{137} + x^{62}$
$x^{19} + x^{139}$	$x^{38} + x^{23}, x^{76} + x^{46}, x^{152} + x^{92}, x^{49} + x^{184}, x^{98} + x^{113}, x^{196} + x^{226}, x^{137} + x^{197}$

Table A1. Cont.

A	Linearly Equivalent Binomial Power Function to A
$x^{19} + x^{154}$	$x^{38} + x^{53}, x^{76} + x^{106}, x^{152} + x^{212}, x^{49} + x^{169}, x^{98} + x^{83}, x^{196} + x^{166}, x^{137} + x^{77}$
$x^{19} + x^{169}$	$x^{38} + x^{83}, x^{76} + x^{166}, x^{152} + x^{77}, x^{49} + x^{154}, x^{98} + x^{53}, x^{196} + x^{106}, x^{137} + x^{212}$
$x^{19} + x^{184}$	$x^{38} + x^{113}, x^{76} + x^{226}, x^{152} + x^{197}, x^{49} + x^{139}, x^{98} + x^{23}, x^{196} + x^{46}, x^{137} + x^{92}$
$x^{19} + x^{199}$	$x^{38} + x^{143}, x^{76} + x^{31}, x^{152} + x^{62}, x^{49} + x^{124}, x^{98} + x^{248}, x^{196} + x^{241}, x^{137} + x^{227}$
$x^{19} + x^{214}$	$x^{38} + x^{173}, x^{76} + x^{91}, x^{152} + x^{182}, x^{49} + x^{109}, x^{98} + x^{218}, x^{196} + x^{181}, x^{137} + x^{107}$
$x^{19} + x^{229}$	$x^{38} + x^{203}, x^{76} + x^{151}, x^{152} + x^{47}, x^{49} + x^{94}, x^{98} + x^{188}, x^{196} + x^{121}, x^{137} + x^{242}$
$x^{19} + x^{244}$	$x^{38} + x^{233}, x^{76} + x^{211}, x^{152} + x^{167}, x^{49} + x^{79}, x^{98} + x^{158}, x^{196} + x^{61}, x^{137} + x^{122}$
$x^{23} + x^{53}$	$x^{46} + x^{106}, x^{92} + x^{212}, x^{184} + x^{169}, x^{113} + x^{83}, x^{226} + x^{166}, x^{197} + x^{77}, x^{139} + x^{154}$
$x^{23} + x^{83}$	$x^{46} + x^{166}, x^{92} + x^{77}, x^{184} + x^{154}, x^{113} + x^{53}, x^{226} + x^{106}, x^{197} + x^{212}, x^{139} + x^{169}$
$x^{23} + x^{143}$	$x^{46} + x^{31}, x^{92} + x^{62}, x^{184} + x^{124}, x^{113} + x^{248}, x^{226} + x^{241}, x^{197} + x^{227}, x^{139} + x^{199}$
$x^{23} + x^{158}$	$x^{46} + x^{61}, x^{92} + x^{122}, x^{184} + x^{244}, x^{113} + x^{233}, x^{226} + x^{211}, x^{197} + x^{167}, x^{139} + x^{79}$
$x^{23} + x^{173}$	$x^{46} + x^{91}, x^{92} + x^{182}, x^{184} + x^{109}, x^{113} + x^{218}, x^{226} + x^{181}, x^{197} + x^{107}, x^{139} + x^{214}$
$x^{23} + x^{188}$	$x^{46} + x^{121}, x^{92} + x^{242}, x^{184} + x^{229}, x^{113} + x^{203}, x^{226} + x^{151}, x^{197} + x^{47}, x^{139} + x^{94}$
$x^{23} + x^{203}$	$x^{46} + x^{151}, x^{92} + x^{47}, x^{184} + x^{94}, x^{113} + x^{188}, x^{226} + x^{121}, x^{197} + x^{242}, x^{139} + x^{229}$
$x^{23} + x^{218}$	$x^{46} + x^{181}, x^{92} + x^{107}, x^{184} + x^{214}, x^{113} + x^{173}, x^{226} + x^{91}, x^{197} + x^{182}, x^{139} + x^{109}$
$x^{23} + x^{233}$	$x^{46} + x^{211}, x^{92} + x^{167}, x^{184} + x^{79}, x^{113} + x^{158}, x^{226} + x^{61}, x^{197} + x^{122}, x^{139} + x^{244}$
$x^{23} + x^{248}$	$x^{46} + x^{241}, x^{92} + x^{227}, x^{184} + x^{199}, x^{113} + x^{143}, x^{226} + x^{31}, x^{197} + x^{62}, x^{139} + x^{124}$
$x^{29} + x^{59}$	$x^{58} + x^{118}, x^{116} + x^{236}, x^{232} + x^{217}, x^{209} + x^{179}, x^{163} + x^{103}, x^{71} + x^{206}, x^{142} + x^{157}$
$x^{29} + x^{74}$	$x^{58} + x^{148}, x^{116} + x^{41}, x^{232} + x^{82}, x^{209} + x^{164}, x^{163} + x^{73}, x^{71} + x^{146}, x^{142} + x^{37}$
$x^{29} + x^{89}$	$x^{58} + x^{178}, x^{116} + x^{101}, x^{232} + x^{202}, x^{209} + x^{149}, x^{163} + x^{43}, x^{71} + x^{86}, x^{142} + x^{172}$
$x^{29} + x^{119}$	$x^{58} + x^{238}, x^{116} + x^{221}, x^{232} + x^{187}, x^{209} + x^{119}, x^{163} + x^{238}, x^{71} + x^{221}, x^{142} + x^{187}$
$x^{29} + x^{149}$	$x^{58} + x^{43}, x^{116} + x^{86}, x^{232} + x^{172}, x^{209} + x^{89}, x^{163} + x^{178}, x^{71} + x^{101}, x^{142} + x^{202}$
$x^{29} + x^{164}$	$x^{58} + x^{73}, x^{116} + x^{146}, x^{232} + x^{37}, x^{209} + x^{74}, x^{163} + x^{148}, x^{71} + x^{41}, x^{142} + x^{82}$
$x^{29} + x^{179}$	$x^{58} + x^{103}, x^{116} + x^{206}, x^{232} + x^{157}, x^{209} + x^{59}, x^{163} + x^{118}, x^{71} + x^{236}, x^{142} + x^{217}$
$x^{29} + x^{239}$	$x^{58} + x^{223}, x^{116} + x^{191}, x^{232} + x^{127}, x^{209} + x^{254}, x^{163} + x^{253}, x^{71} + x^{251}, x^{142} + x^{247}$
$x^{29} + x^{254}$	$x^{58} + x^{253}, x^{116} + x^{251}, x^{232} + x^{247}, x^{209} + x^{239}, x^{163} + x^{223}, x^{71} + x^{191}, x^{142} + x^{127}$
$x^{31} + x^{61}$	$x^{62} + x^{122}, x^{124} + x^{244}, x^{248} + x^{233}, x^{241} + x^{211}, x^{227} + x^{167}, x^{199} + x^{79}, x^{143} + x^{158}$
$x^{31} + x^{91}$	$x^{62} + x^{182}, x^{124} + x^{109}, x^{248} + x^{218}, x^{241} + x^{181}, x^{227} + x^{107}, x^{199} + x^{214}, x^{143} + x^{173}$
$x^{31} + x^{106}$	$x^{62} + x^{212}, x^{124} + x^{169}, x^{248} + x^{83}, x^{241} + x^{166}, x^{227} + x^{77}, x^{199} + x^{154}, x^{143} + x^{53}$
$x^{31} + x^{121}$	$x^{62} + x^{242}, x^{124} + x^{229}, x^{248} + x^{203}, x^{241} + x^{151}, x^{227} + x^{47}, x^{199} + x^{94}, x^{143} + x^{188}$
$x^{31} + x^{151}$	$x^{62} + x^{47}, x^{124} + x^{94}, x^{248} + x^{188}, x^{241} + x^{121}, x^{227} + x^{242}, x^{199} + x^{229}, x^{143} + x^{203}$
$x^{31} + x^{166}$	$x^{62} + x^{77}, x^{124} + x^{154}, x^{248} + x^{53}, x^{241} + x^{106}, x^{227} + x^{212}, x^{199} + x^{169}, x^{143} + x^{83}$
$x^{31} + x^{181}$	$x^{62} + x^{107}, x^{124} + x^{214}, x^{248} + x^{173}, x^{241} + x^{91}, x^{227} + x^{182}, x^{199} + x^{109}, x^{143} + x^{218}$
$x^{31} + x^{211}$	$x^{62} + x^{167}, x^{124} + x^{79}, x^{248} + x^{158}, x^{241} + x^{61}, x^{227} + x^{122}, x^{199} + x^{244}, x^{143} + x^{233}$
$x^{37} + x^{127}$	$x^{74} + x^{254}, x^{148} + x^{253}, x^{41} + x^{251}, x^{82} + x^{247}, x^{164} + x^{239}, x^{73} + x^{223}, x^{146} + x^{191}$
$x^{37} + x^{157}$	$x^{74} + x^{59}, x^{148} + x^{118}, x^{41} + x^{236}, x^{82} + x^{217}, x^{164} + x^{179}, x^{73} + x^{103}, x^{146} + x^{206}$
$x^{37} + x^{172}$	$x^{74} + x^{89}, x^{148} + x^{178}, x^{41} + x^{101}, x^{82} + x^{202}, x^{164} + x^{149}, x^{73} + x^{43}, x^{146} + x^{86}$
$x^{37} + x^{187}$	$x^{74} + x^{119}, x^{148} + x^{238}, x^{41} + x^{221}, x^{82} + x^{187}, x^{164} + x^{119}, x^{73} + x^{238}, x^{146} + x^{221}$
$x^{37} + x^{202}$	$x^{74} + x^{149}, x^{148} + x^{43}, x^{41} + x^{86}, x^{82} + x^{172}, x^{164} + x^{89}, x^{73} + x^{178}, x^{146} + x^{101}$
$x^{37} + x^{217}$	$x^{74} + x^{179}, x^{148} + x^{103}, x^{41} + x^{206}, x^{82} + x^{157}, x^{164} + x^{59}, x^{73} + x^{118}, x^{146} + x^{236}$
$x^{37} + x^{247}$	$x^{74} + x^{239}, x^{148} + x^{223}, x^{41} + x^{191}, x^{82} + x^{127}, x^{164} + x^{254}, x^{73} + x^{253}, x^{146} + x^{251}$
$x^{43} + x^{103}$	$x^{86} + x^{206}, x^{172} + x^{157}, x^{89} + x^{59}, x^{178} + x^{118}, x^{101} + x^{236}, x^{202} + x^{217}, x^{149} + x^{179}$
$x^{43} + x^{118}$	$x^{86} + x^{236}, x^{172} + x^{217}, x^{89} + x^{179}, x^{178} + x^{103}, x^{101} + x^{206}, x^{202} + x^{157}, x^{149} + x^{59}$
$x^{43} + x^{223}$	$x^{86} + x^{191}, x^{172} + x^{127}, x^{89} + x^{254}, x^{178} + x^{253}, x^{101} + x^{251}, x^{202} + x^{247}, x^{149} + x^{239}$
$x^{43} + x^{238}$	$x^{86} + x^{221}, x^{172} + x^{187}, x^{89} + x^{119}, x^{178} + x^{238}, x^{101} + x^{221}, x^{202} + x^{187}, x^{149} + x^{119}$
$x^{43} + x^{253}$	$x^{86} + x^{251}, x^{172} + x^{247}, x^{89} + x^{239}, x^{178} + x^{223}, x^{101} + x^{191}, x^{202} + x^{127}, x^{149} + x^{254}$
$x^{47} + x^{77}$	$x^{94} + x^{154}, x^{188} + x^{53}, x^{121} + x^{106}, x^{242} + x^{212}, x^{229} + x^{169}, x^{203} + x^{83}, x^{151} + x^{166}$
$x^{47} + x^{107}$	$x^{94} + x^{214}, x^{188} + x^{173}, x^{121} + x^{91}, x^{242} + x^{182}, x^{229} + x^{109}, x^{203} + x^{218}, x^{151} + x^{181}$
$x^{47} + x^{122}$	$x^{94} + x^{244}, x^{188} + x^{233}, x^{121} + x^{211}, x^{242} + x^{167}, x^{229} + x^{79}, x^{203} + x^{158}, x^{151} + x^{61}$
$x^{47} + x^{167}$	$x^{94} + x^{79}, x^{188} + x^{158}, x^{121} + x^{61}, x^{242} + x^{122}, x^{229} + x^{244}, x^{203} + x^{233}, x^{151} + x^{211}$
$x^{47} + x^{182}$	$x^{94} + x^{109}, x^{188} + x^{218}, x^{121} + x^{181}, x^{242} + x^{107}, x^{229} + x^{214}, x^{203} + x^{173}, x^{151} + x^{91}$
$x^{47} + x^{212}$	$x^{94} + x^{169}, x^{188} + x^{83}, x^{121} + x^{166}, x^{242} + x^{77}, x^{229} + x^{154}, x^{203} + x^{53}, x^{151} + x^{106}$
$x^{53} + x^{158}$	$x^{106} + x^{61}, x^{212} + x^{122}, x^{169} + x^{244}, x^{83} + x^{233}, x^{166} + x^{211}, x^{77} + x^{167}, x^{154} + x^{79}$
$x^{53} + x^{173}$	$x^{106} + x^{91}, x^{212} + x^{182}, x^{169} + x^{109}, x^{83} + x^{218}, x^{166} + x^{181}, x^{77} + x^{107}, x^{154} + x^{214}$
$x^{53} + x^{218}$	$x^{106} + x^{181}, x^{212} + x^{107}, x^{169} + x^{214}, x^{83} + x^{173}, x^{166} + x^{91}, x^{77} + x^{182}, x^{154} + x^{109}$
$x^{53} + x^{233}$	$x^{106} + x^{211}, x^{212} + x^{167}, x^{169} + x^{79}, x^{83} + x^{158}, x^{166} + x^{61}, x^{77} + x^{122}, x^{154} + x^{244}$
$x^{59} + x^{119}$	$x^{118} + x^{238}, x^{236} + x^{221}, x^{217} + x^{187}, x^{179} + x^{119}, x^{103} + x^{238}, x^{206} + x^{221}, x^{157} + x^{187}$
$x^{59} + x^{239}$	$x^{118} + x^{223}, x^{236} + x^{191}, x^{217} + x^{127}, x^{179} + x^{254}, x^{103} + x^{253}, x^{206} + x^{251}, x^{157} + x^{247}$
$x^{59} + x^{254}$	$x^{118} + x^{253}, x^{236} + x^{251}, x^{217} + x^{247}, x^{179} + x^{239}, x^{103} + x^{223}, x^{206} + x^{191}, x^{157} + x^{127}$
$x^{61} + x^{91}$	$x^{122} + x^{182}, x^{244} + x^{109}, x^{233} + x^{218}, x^{211} + x^{181}, x^{167} + x^{107}, x^{79} + x^{214}, x^{158} + x^{173}$
$x^{61} + x^{181}$	$x^{122} + x^{107}, x^{244} + x^{214}, x^{233} + x^{173}, x^{211} + x^{91}, x^{167} + x^{182}, x^{79} + x^{109}, x^{158} + x^{218}$
$x^{119} + x^{239}$	$x^{238} + x^{223}, x^{221} + x^{191}, x^{187} + x^{127}$
$x^{119} + x^{254}$	$x^{238} + x^{253}, x^{221} + x^{251}, x^{187} + x^{247}$

Appendix A.2

Table A2. Classification of Linearly Non-Equivalent Binomial Power Function with 17 to 1 Output.

B	Linearly Equivalent Binomial Power Function to B
$x^1 + x^{120}$	$x^2 + x^{240}, x^4 + x^{225}, x^8 + x^{195}, x^{16} + x^{135}, x^{32} + x^{15}, x^{64} + x^{30}, x^{128} + x^{60}$
$x^7 + x^{75}$	$x^{14} + x^{150}, x^{28} + x^{45}, x^{56} + x^{90}, x^{112} + x^{180}, x^{224} + x^{105}, x^{193} + x^{210}, x^{131} + x^{165}$
$x^{11} + x^{45}$	$x^{22} + x^{90}, x^{44} + x^{180}, x^{88} + x^{105}, x^{176} + x^{210}, x^{97} + x^{165}, x^{194} + x^{75}, x^{133} + x^{150}$
$x^{13} + x^{30}$	$x^{26} + x^{60}, x^{52} + x^{120}, x^{104} + x^{240}, x^{208} + x^{225}, x^{161} + x^{195}, x^{67} + x^{135}, x^{134} + x^{15}$
$x^{15} + x^{49}$	$x^{30} + x^{98}, x^{60} + x^{196}, x^{120} + x^{137}, x^{240} + x^{19}, x^{225} + x^{38}, x^{195} + x^{76}, x^{135} + x^{152}$
$x^{15} + x^{83}$	$x^{30} + x^{166}, x^{60} + x^{77}, x^{120} + x^{154}, x^{240} + x^{53}, x^{225} + x^{106}, x^{195} + x^{212}, x^{135} + x^{169}$
$x^{15} + x^{151}$	$x^{30} + x^{47}, x^{60} + x^{94}, x^{120} + x^{188}, x^{240} + x^{121}, x^{225} + x^{242}, x^{195} + x^{229}, x^{135} + x^{203}$
$x^{15} + x^{202}$	$x^{30} + x^{149}, x^{60} + x^{43}, x^{120} + x^{86}, x^{240} + x^{172}, x^{225} + x^{89}, x^{195} + x^{178}, x^{135} + x^{101}$
$x^{15} + x^{236}$	$x^{30} + x^{217}, x^{60} + x^{179}, x^{120} + x^{103}, x^{240} + x^{206}, x^{225} + x^{157}, x^{195} + x^{59}, x^{135} + x^{118}$
$x^{15} + x^{253}$	$x^{30} + x^{251}, x^{60} + x^{247}, x^{120} + x^{239}, x^{240} + x^{223}, x^{225} + x^{191}, x^{195} + x^{127}, x^{135} + x^{254}$
$x^{23} + x^{210}$	$x^{46} + x^{165}, x^{92} + x^{75}, x^{184} + x^{150}, x^{113} + x^{45}, x^{226} + x^{90}, x^{197} + x^{180}, x^{139} + x^{105}$
$x^{29} + x^{165}$	$x^{58} + x^{75}, x^{116} + x^{150}, x^{232} + x^{45}, x^{209} + x^{90}, x^{163} + x^{180}, x^{71} + x^{105}, x^{142} + x^{210}$
$x^{31} + x^{150}$	$x^{62} + x^{45}, x^{124} + x^{90}, x^{248} + x^{180}, x^{241} + x^{105}, x^{227} + x^{210}, x^{199} + x^{165}, x^{143} + x^{75}$
$x^{37} + x^{105}$	$x^{74} + x^{210}, x^{148} + x^{165}, x^{41} + x^{75}, x^{82} + x^{150}, x^{164} + x^{45}, x^{73} + x^{90}, x^{146} + x^{180}$
$x^{45} + x^{79}$	$x^{90} + x^{158}, x^{180} + x^{61}, x^{105} + x^{122}, x^{210} + x^{244}, x^{165} + x^{233}, x^{75} + x^{211}, x^{150} + x^{167}$
$x^{45} + x^{181}$	$x^{90} + x^{107}, x^{180} + x^{214}, x^{105} + x^{173}, x^{210} + x^{91}, x^{165} + x^{182}, x^{75} + x^{109}, x^{150} + x^{218}$

Appendix A.3

Table A3. Classification of Linearly Non-Equivalent Binomial Power Function with 85 to 1 Output.

D	Linearly Equivalent Binomial Power Function to D
$x^1 + x^{120}$	$x^2 + x^{240}, x^4 + x^{225}, x^8 + x^{195}, x^{16} + x^{135}, x^{32} + x^{15}, x^{64} + x^{30}, x^{128} + x^{60}$
$x^7 + x^{75}$	$x^{14} + x^{150}, x^{28} + x^{45}, x^{56} + x^{90}, x^{112} + x^{180}, x^{224} + x^{105}, x^{193} + x^{210}, x^{131} + x^{165}$
$x^{11} + x^{45}$	$x^{22} + x^{90}, x^{44} + x^{180}, x^{88} + x^{105}, x^{176} + x^{210}, x^{97} + x^{165}, x^{194} + x^{75}, x^{133} + x^{150}$
$x^{13} + x^{30}$	$x^{26} + x^{60}, x^{52} + x^{120}, x^{104} + x^{240}, x^{208} + x^{225}, x^{161} + x^{195}, x^{67} + x^{135}, x^{134} + x^{15}$
$x^{15} + x^{49}$	$x^{30} + x^{98}, x^{60} + x^{196}, x^{120} + x^{137}, x^{240} + x^{19}, x^{225} + x^{38}, x^{195} + x^{76}, x^{135} + x^{152}$
$x^{15} + x^{83}$	$x^{30} + x^{166}, x^{60} + x^{77}, x^{120} + x^{154}, x^{240} + x^{53}, x^{225} + x^{106}, x^{195} + x^{212}, x^{135} + x^{169}$
$x^{15} + x^{151}$	$x^{30} + x^{47}, x^{60} + x^{94}, x^{120} + x^{188}, x^{240} + x^{121}, x^{225} + x^{242}, x^{195} + x^{229}, x^{135} + x^{203}$
$x^{15} + x^{202}$	$x^{30} + x^{149}, x^{60} + x^{43}, x^{120} + x^{86}, x^{240} + x^{172}, x^{225} + x^{89}, x^{195} + x^{178}, x^{135} + x^{101}$
$x^{15} + x^{236}$	$x^{30} + x^{217}, x^{60} + x^{179}, x^{120} + x^{103}, x^{240} + x^{206}, x^{225} + x^{157}, x^{195} + x^{59}, x^{135} + x^{118}$
$x^{15} + x^{253}$	$x^{30} + x^{251}, x^{60} + x^{247}, x^{120} + x^{239}, x^{240} + x^{223}, x^{225} + x^{191}, x^{195} + x^{127}, x^{135} + x^{254}$
$x^{23} + x^{210}$	$x^{46} + x^{165}, x^{92} + x^{75}, x^{184} + x^{150}, x^{113} + x^{45}, x^{226} + x^{90}, x^{197} + x^{180}, x^{139} + x^{105}$
$x^{29} + x^{165}$	$x^{58} + x^{75}, x^{116} + x^{150}, x^{232} + x^{45}, x^{209} + x^{90}, x^{163} + x^{180}, x^{71} + x^{105}, x^{142} + x^{210}$
$x^{31} + x^{150}$	$x^{62} + x^{45}, x^{124} + x^{90}, x^{248} + x^{180}, x^{241} + x^{105}, x^{227} + x^{210}, x^{199} + x^{165}, x^{143} + x^{75}$
$x^{37} + x^{105}$	$x^{74} + x^{210}, x^{148} + x^{165}, x^{41} + x^{75}, x^{82} + x^{150}, x^{164} + x^{45}, x^{73} + x^{90}, x^{146} + x^{180}$
$x^{45} + x^{79}$	$x^{90} + x^{158}, x^{180} + x^{61}, x^{105} + x^{122}, x^{210} + x^{244}, x^{165} + x^{233}, x^{75} + x^{211}, x^{150} + x^{167}$
$x^{45} + x^{181}$	$x^{90} + x^{107}, x^{180} + x^{214}, x^{105} + x^{173}, x^{210} + x^{91}, x^{165} + x^{182}, x^{75} + x^{109}, x^{150} + x^{218}$

Appendix A.4

Table A4. Classification of Linearly Non-Equivalent Binomial Power Function with 51 to 1 Output.

C	Linearly Equivalent Binomial Power Function to C
$x^1 + x^{52}$	$x^2 + x^{104}, x^4 + x^{208}, x^8 + x^{161}, x^{16} + x^{67}, x^{32} + x^{134}, x^{64} + x^{13}, x^{128} + x^{26}$
$x^1 + x^{103}$	$x^2 + x^{206}, x^4 + x^{157}, x^8 + x^{59}, x^{16} + x^{118}, x^{32} + x^{236}, x^{64} + x^{217}, x^{128} + x^{179}$
$x^1 + x^{205}$	$x^2 + x^{155}, x^4 + x^{55}, x^8 + x^{110}, x^{16} + x^{220}, x^{32} + x^{185}, x^{64} + x^{115}, x^{128} + x^{230}$
$x^5 + x^{56}$	$x^{10} + x^{112}, x^{20} + x^{224}, x^{40} + x^{193}, x^{80} + x^{131}, x^{160} + x^{7}, x^{65} + x^{14}, x^{130} + x^{28}$
$x^5 + x^{107}$	$x^{10} + x^{214}, x^{20} + x^{173}, x^{40} + x^{91}, x^{80} + x^{182}, x^{160} + x^{109}, x^{65} + x^{218}, x^{130} + x^{181}$
$x^5 + x^{158}$	$x^{10} + x^{61}, x^{20} + x^{122}, x^{40} + x^{244}, x^{80} + x^{233}, x^{160} + x^{211}, x^{65} + x^{167}, x^{130} + x^{79}$
$x^5 + x^{209}$	$x^{10} + x^{163}, x^{20} + x^{71}, x^{40} + x^{142}, x^{80} + x^{29}, x^{160} + x^{58}, x^{65} + x^{116}, x^{130} + x^{232}$
$x^7 + x^{109}$	$x^{14} + x^{218}, x^{28} + x^{181}, x^{56} + x^{107}, x^{112} + x^{214}, x^{224} + x^{173}, x^{193} + x^{91}, x^{131} + x^{182}$
$x^7 + x^{211}$	$x^{14} + x^{167}, x^{28} + x^{79}, x^{56} + x^{158}, x^{112} + x^{61}, x^{224} + x^{122}, x^{193} + x^{244}, x^{131} + x^{233}$
$x^{11} + x^{62}$	$x^{22} + x^{124}, x^{44} + x^{248}, x^{88} + x^{241}, x^{176} + x^{227}, x^{97} + x^{199}, x^{194} + x^{143}, x^{133} + x^{31}$

Table A4. *Cont.*

C	Linearly Equivalent Binomial Power Function to C
$x^{11} + x^{113}$	$x^{22} + x^{226}, x^{44} + x^{197}, x^{88} + x^{139}, x^{176} + x^{23}, x^{97} + x^{46}, x^{194} + x^{92}, x^{133} + x^{184}$
$x^{11} + x^{215}$	$x^{22} + x^{175}, x^{44} + x^{95}, x^{88} + x^{190}, x^{176} + x^{125}, x^{97} + x^{250}, x^{194} + x^{245}, x^{133} + x^{235}$
$x^{13} + x^{115}$	$x^{26} + x^{230}, x^{52} + x^{205}, x^{104} + x^{155}, x^{208} + x^{55}, x^{161} + x^{110}, x^{67} + x^{220}, x^{134} + x^{185}$
$x^{13} + x^{166}$	$x^{26} + x^{77}, x^{52} + x^{154}, x^{104} + x^{53}, x^{208} + x^{106}, x^{161} + x^{212}, x^{67} + x^{169}, x^{134} + x^{83}$
$x^{19} + x^{70}$	$x^{38} + x^{140}, x^{76} + x^{25}, x^{152} + x^{50}, x^{49} + x^{100}, x^{98} + x^{200}, x^{196} + x^{145}, x^{137} + x^{35}$
$x^{19} + x^{172}$	$x^{38} + x^{89}, x^{76} + x^{178}, x^{152} + x^{101}, x^{49} + x^{202}, x^{98} + x^{149}, x^{196} + x^{43}, x^{137} + x^{86}$
$x^{19} + x^{223}$	$x^{38} + x^{191}, x^{76} + x^{127}, x^{152} + x^{254}, x^{49} + x^{253}, x^{98} + x^{251}, x^{196} + x^{247}, x^{137} + x^{239}$
$x^{23} + x^{74}$	$x^{46} + x^{148}, x^{92} + x^{41}, x^{184} + x^{82}, x^{113} + x^{164}, x^{226} + x^{73}, x^{197} + x^{146}, x^{139} + x^{37}$
$x^{23} + x^{125}$	$x^{46} + x^{250}, x^{92} + x^{245}, x^{184} + x^{235}, x^{113} + x^{215}, x^{226} + x^{175}, x^{197} + x^{95}, x^{139} + x^{190}$
$x^{25} + x^{127}$	$x^{50} + x^{254}, x^{100} + x^{253}, x^{200} + x^{251}, x^{145} + x^{247}, x^{35} + x^{239}, x^{70} + x^{223}, x^{140} + x^{191}$
$x^{25} + x^{178}$	$x^{50} + x^{101}, x^{100} + x^{202}, x^{200} + x^{149}, x^{145} + x^{43}, x^{35} + x^{86}, x^{70} + x^{172}, x^{140} + x^{89}$
$x^{25} + x^{229}$	$x^{50} + x^{203}, x^{100} + x^{151}, x^{200} + x^{47}, x^{145} + x^{94}, x^{35} + x^{188}, x^{70} + x^{121}, x^{140} + x^{242}$
$x^{29} + x^{182}$	$x^{58} + x^{109}, x^{116} + x^{218}, x^{232} + x^{181}, x^{209} + x^{107}, x^{163} + x^{214}, x^{71} + x^{173}, x^{142} + x^{91}$
$x^{29} + x^{233}$	$x^{58} + x^{211}, x^{116} + x^{167}, x^{232} + x^{79}, x^{209} + x^{158}, x^{163} + x^{61}, x^{71} + x^{122}, x^{142} + x^{244}$
$x^{31} + x^{82}$	$x^{62} + x^{164}, x^{124} + x^{73}, x^{248} + x^{146}, x^{241} + x^{37}, x^{227} + x^{74}, x^{199} + x^{148}, x^{143} + x^{41}$
$x^{31} + x^{235}$	$x^{62} + x^{215}, x^{124} + x^{175}, x^{248} + x^{95}, x^{241} + x^{190}, x^{227} + x^{125}, x^{199} + x^{250}, x^{143} + x^{245}$
$x^{37} + x^{190}$	$x^{74} + x^{125}, x^{148} + x^{250}, x^{41} + x^{245}, x^{82} + x^{235}, x^{164} + x^{215}, x^{73} + x^{175}, x^{146} + x^{95}$
$x^{43} + x^{94}$	$x^{86} + x^{188}, x^{172} + x^{121}, x^{89} + x^{242}, x^{178} + x^{229}, x^{101} + x^{203}, x^{202} + x^{151}, x^{149} + x^{47}$
$x^{47} + x^{251}$	$x^{94} + x^{247}, x^{188} + x^{239}, x^{121} + x^{223}, x^{242} + x^{191}, x^{229} + x^{127}, x^{203} + x^{254}, x^{151} + x^{253}$
$x^{53} + x^{155}$	$x^{106} + x^{55}, x^{212} + x^{110}, x^{169} + x^{220}, x^{83} + x^{185}, x^{166} + x^{115}, x^{77} + x^{230}, x^{154} + x^{205}$
$x^{53} + x^{206}$	$x^{106} + x^{157}, x^{212} + x^{59}, x^{169} + x^{118}, x^{83} + x^{236}, x^{166} + x^{217}, x^{77} + x^{179}, x^{154} + x^{103}$
$x^{55} + x^{157}$	$x^{110} + x^{59}, x^{220} + x^{118}, x^{185} + x^{236}, x^{115} + x^{217}, x^{230} + x^{179}, x^{205} + x^{103}, x^{155} + x^{206}$

References

- Shannon, C.E. Communication Theory of Secrecy Systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
- Biham, E.; Shamir, A. Differential Cryptanalysis of DES-like Cryptosystems. In *Advances in Cryptology—CRYPTO '90*; Lecture Notes in Computer Science; Menezes, A.J., Vanstone, S.A., Eds.; Springer: Berlin/Heidelberg, Germany, 1991; Volume 537, pp. 2–21.
- Matsui, M. Linear Cryptanalysis Method for DES Cipher. In *EUROCRYPT '93*; Lecture Notes in Computer Science; Helleseht, T., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; Volume 765, pp. 386–397.
- Lai, X. Higher Order Derivatives and Differential Cryptanalysis. In *Communications and Cryptography*; The Springer International Series in Engineering and Computer Science; Blahut, R.E., Costello, D.J.J., Maurer, U., Mittelholzer, T., Eds.; Springer: New York, NY, USA, 1994; Volume 276, pp. 227–233.
- Knudsen, L.R. Truncated and Higher Order Differentials. In *Fast Software Encryption*; Lecture Notes in Computer Science; Preneel, B., Ed.; Springer: Berlin/Heidelberg, Germany, 1995; Volume 1008, pp. 196–211.
- Jakobsen, T.; Knudsen, L.R. The Interpolation Attack on Block Ciphers. In *Fast Software Encryption*; Lecture Notes in Computer Science; Biham, E., Ed.; Springer: Berlin/Heidelberg, Germany, 1997; Volume 1267, pp. 28–40.
- Courtois, N.T.; Pieprzyk, J. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In *ASIACRYPT 2002*; Lecture Notes in Computer Science; Zheng, Y., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2501, pp. 267–287. [CrossRef]
- Federal Information Processing Standard (FIPS) 197*; Advanced Encryption Standard. National Institute of Standards and Technology: Gaithersburg, MD, USA, 2001. [CrossRef]
- Dobbertin, H.; Daum, M.; Felke, P.; Lange, T.; Leander, G. S-Boxes and Filters. STORK Project, IST-2002-38273. 2002. Available online: <http://www.stork.eu.org/papers/> (accessed on 20 July 2022).
- Carlet, C. *Vectorial Boolean Functions for Cryptography*; Cambridge University Press: Cambridge, UK, 2010; Chapter Boolean Models and Methods in Mathematics, Computer Science, and Engineering; pp. 398–470.
- Mamadolimov, A.; Isa, H.; Mohamad, M.S. Practical Bijective S-box Design. *arXiv* **2013**, arXiv:1301.4723.
- Isa, H.; Jamil, N.; Z'aba, M.R. S-box Construction from Non-Permutation Power Functions. In Proceedings of the 6th International Conference on Security of Information and Networks, Aksaray, Turkey, 26–28 November 2013; ACM: New York, NY, USA, 2013; pp. 46–53. [CrossRef]
- Roslan, M.F.; Seman, K.; Halim, A.; Sayuti, M. Current implementation of advance encryption standard (AES) S-Box. *J. Fundam. Appl. Sci.* **2017**, *9*, 518–542.
- Carlet, C. On Known and New Differentially Uniform Functions. In *Information Security and Privacy*; Lecture Notes in Computer Science; Parampalli, U., Hawkes, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6812, pp. 1–15.
- Piret, G.; Roche, T.; Carlet, C. PICARO—A Block Cipher Allowing Efficient Higher-Order Side-Channel Resistance. In *Applied Cryptography and Network Security*; Lecture Notes in Computer Science; Bao, F., Samarati, P., Zhou, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7341, pp. 311–328.
- Daemen, J.; Knudsen, L.; Rijmen, V. The Block Cipher SQUARE. In *Fast Software Encryption*; Lecture Notes in Computer Science; Biham, E., Ed.; Springer: Berlin/Heidelberg, Germany, 1997; Volume 1267, pp. 149–165.
- Daemen, J.; Rijmen, V. The Block Cipher BKSQ. In *Smart Card Research and Applications*; Lecture Notes in Computer Science; Quisquater, J.J., Schneier, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1820, pp. 236–245.

18. Daemen, J.; Rijmen, V. AES Proposal: Rijndael. NIST AES Proposal. 1999. Available online: <https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development> (accessed on 20 July 2022).
19. Daemen, J.; Rijmen, V. The Block Cipher Rijndael. In *Smart Card Research and Applications*; Lecture Notes in Computer Science; Quisquater, J.J., Schneier, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1820, pp. 277–284.
20. Nyberg, K. Differentially Uniform Mappings for Cryptography. In *Advances in Cryptology—EUROCRYPT '93*; Lecture Notes in Computer Science; Helleseht, T., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; Volume 765, pp. 55–64.
21. Aoki, K.; Ichikawa, T.; Kanda, M.; Matsui, M.; Moriai, S.; Nakajima, J.; Tokita, T. Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms—Design and Analysis. In *Selected Areas in Cryptography*; Lecture Notes in Computer Science; Stinson, D.R., Tavares, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2012, pp. 39–56.
22. Crowley, P. Mercy: A Fast Large Block Cipher for Disk Sector Encryption. In *Fast Software Encryption*; Lecture Notes in Computer Science; Goos, G., Hartmanis, J., van Leeuwen, J., Schneier, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 1978, pp. 49–63.
23. Kwon, D.; Kim, J.; Park, S.; Sung, S.; Sohn, Y.; Song, J.; Yeom, Y.; Yoon, E.J.; Lee, S.; Lee, J.; et al. New Block Cipher: ARIA. In *Information Security and Cryptology—ICISC 2003*; Lecture Notes in Computer Science; Lim, J.I., Lee, D.H., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 2971, pp. 432–445.
24. Diffie, W.; Ledin, G. SMS4 Encryption Algorithm for Wireless Networks. Cryptology ePrint Archive, Report 2008/329. 2008. Available online: <http://eprint.iacr.org/> (accessed on 25 July 2022).
25. Kurniawan, Y.; Mardiyanto, M.S.; Supriana, I.; Sutikno, S. The New Block Cipher: BC2. *Int. J. Netw. Secur.* **2009**, *8*, 16–24.
26. Hirata, K. The 128-bit Block Cipher HyRAL (Hybrid Randomization Algorithm): Common Key Block Cipher. In Proceedings of the International Symposium on Intelligence Information Processing and Trusted Computing (IPTC), Wuhan, China, 28–29 October 2010; pp. 9–14. [[CrossRef](#)]
27. Bucholc, K.; Chmiel, K.; Grochowska-Czuryło, A.; Idzikowska, E.; Janicka-Lipska, I.; Stokłosa, J. Scalable PP-1 Block Cipher. *Int. J. Appl. Math. Comput. Sci.* **2010**, *20*, 401–411.
28. Fuller, J.; Millan, W. Linear Redundancy in S-boxes. In *Fast Software Encryption*; Lecture Notes in Computer Science; Johansson, T., Ed.; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2887, pp. 74–86.
29. Stern, J.; Vaudenay, S. CS-Cipher. In *Fast Software Encryption*; Lecture Notes in Computer Science; Vaudenay, S., Ed.; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1372, pp. 189–204.
30. Lim, C.H. A Revised Version of CRYPTON: CRYPTON V1.0. In *Fast Software Encryption*; Lecture Notes in Computer Science; Knudsen, L., Ed.; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1636, pp. 31–45.
31. Gérard, B.; Grosso, V.; Naya-Plasencia, M.; Standaert, F.X. Block Ciphers That Are Easier to Mask: How Far Can We Go? In *Cryptographic Hardware and Embedded Systems—CHES 2013*; Lecture Notes in Computer Science; Bertoni, G., Coron, J.S., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8086, pp. 383–399.
32. NIST. Skipjack and KEA Algorithm Specifications. 1998. Available online: <https://csrc.nist.gov/Presentations/1998/Skipjack-and-KEA-Algorithm-Specifications/> (accessed on 20 July 2022).
33. Junod, P.; Vaudenay, S. FOX: A New Family of Block Ciphers. In *Selected Areas in Cryptography*; Lecture Notes in Computer Science; Handschuh, H., Hasan, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3357, pp. 114–129.
34. Biryukov, A.; Perrin, L. On Reverse-Engineering S-Boxes with Hidden Design Criteria or Structure. In *Advances in Cryptology—CRYPTO 2015*; Lecture Notes in Computer Science; Gennaro, R., Robshaw, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9215, pp. 116–140. [[CrossRef](#)]
35. Peyravian, M.; Coppersmith, D. A Structured Symmetric-key Block Cipher. *Comput. Secur.* **1999**, *18*, 134–147.
36. Barreto, P.S.L.M.; Rijmen, V. The ANUBIS Block Cipher. 2000. Available online: <https://www.cosic.esat.kuleuven.be/nessie/workshop/> (accessed on 20 July 2022).
37. Barreto, P.S.L.M.; Rijmen, V. The KHAZAD Legacy-Level Block Cipher. 2000. Available online: <https://www.cosic.esat.kuleuven.be/nessie/workshop/> (accessed on 20 July 2022).
38. Standaert, F.X.; Piret, G.; Rouvroy, G.; Quisquater, J.J.; Legat, J.D. ICEBERG: An Involutional Cipher Efficient for Block Encryption in Reconfigurable Hardware. In *Fast Software Encryption*; Lecture Notes in Computer Science; Roy, B., Meier, W., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3017, pp. 279–298.
39. Elkamchouchi, H.M.; Makar, M.A. Kamkar Symmetric Block Cipher. In Proceedings of the Twenty-First National Radio Science Conference (NRSC), Cairo, Egypt, 18 March 2004; pp. C1–C18.
40. Ohkuma, K.; Muratani, H.; Sano, F.; Kawamura, S. The Block Cipher Hierocrypt. In *Selected Areas in Cryptography*; Lecture Notes in Computer Science; Stinson, D., Tavares, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2012, pp. 72–88.
41. Shirai, T.; Shibutani, K.; Akishita, T.; Moriai, S.; Iwata, T. The 128-Bit Blockcipher CLEFIA (Extended Abstract). In *Fast Software Encryption: 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, 26–28 March 2007*; Revised Selected Papers; Springer: Berlin/Heidelberg, Germany, 2007; pp. 181–195.
42. Oliynykov, R.; Gorbenko, I.; Kazymyrov, O.; Ruzhentsev, V.; Kuznetsov, O.; Gorbenko, Y.; Dyrda, O.; Dolgov, V.; Pushkaryov, A.; Mordvinov, R.; et al. A New Encryption Standard of Ukraine: The Kalyna Block Cipher; Cryptology ePrint Archive, Report 2015/650. 2015. Available online: <https://eprint.iacr.org/2015/650> (accessed on 30 July 2022).
43. Kazymyrov, O.; Kazymyrova, V.; Oliynykov, R. A Method for Generation of High-Nonlinear S-Boxes Based on Gradient Descent; Cryptology ePrint Archive, Report 2013/578. 2013. Available online: <https://eprint.iacr.org/2013/578> (accessed on 30 July 2022).

44. Gao, S.; Ma, W.; Feng, J.; Guo, N.; Yan, Y. Improved Hill-Climbing Methods in the Design of Bijective S-boxes. In Proceedings of the Sixth International Conference on Natural Computation (ICNC), Yantai, China, 10–12 August 2010; Volume 5, pp. 2378–2380.
45. El-Ramly, S.H.; El-Garf, T.; Soliman, A.H. Dynamic Generation of S-boxes in Block Cipher Systems. In Proceedings of the Eighteenth National Radio Science Conference, Mansoura, Egypt, 27–29 March 2001; Volume 2, pp. 389–397. [[CrossRef](#)]
46. Kazlauskas, K.; Vaicekuskas, G.; Smaliukas, R. An Algorithm for Key-Dependent S-Box Generation in Block Cipher System. *Informatica* **2015**, *26*, 51–65.
47. Balajee, M.K.; Gnanasekar, J.M. Evaluation of Key Dependent S-Box based Data Security Algorithm using Hamming Distance and Balanced Output. *TEM J.* **2016**, *5*, 67–75. [[CrossRef](#)]
48. El-Latif, A.A.A.; Ramadoss, J.; Abd-El-Atty, B.; Khalifa, H.S.; Nazarimehr, F. A Novel Chaos-Based Cryptography Algorithm and Its Performance Analysis. *Mathematics* **2022**, *10*, 2434.
49. Gong, G.; Gupta, K.C.; Hell, M.; Nawaz, Y. Towards a General RC4-Like Keystream Generator. In *Lecture Notes in Computer Science, Proceedings of Information Security and Cryptology: First SKLOIS Conference, CISC 2005, Beijing, China, 15–17 December 2005*; Feng, D., Lin, D., Yung, M., Eds.; Springer: Berlin/ Heidelberg, Germany, 2005; Volume 3822, pp. 162–174.
50. Mamadolimov, A.; Isa, H.; Ahmad, M.M.; Mohamad, M.S. Nonlinear Boolean Permutations. *Pertanika J. Sci. Technol.* **2011**, *19*, 1–9. [[CrossRef](#)]
51. Zhou, Q.; Wong, K.; Liao, X.; Xiang, T.; Hu, Y. Parallel Image Encryption Algorithm based on Discretized Chaotic Map. *Chaos* **2008**, *38*, 1081–1092.
52. Xu, G.; Zhao, G.; Min, L. The Design of Dynamical S-boxes based on Discrete Chaos Map System. In Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems, Shanghai, China, 20–22 November 2009; Volume 2, pp. 473–478.
53. Hung, P.A.; Klomkarn, K.; Sooraksa, P. Image Encryption based on Chaotic Map and Dynamic S-box. In Proceedings of the International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), Okinawa, Japan, 12–15 November 2013; pp. 435–439. [[CrossRef](#)]
54. Gondal, M.A.; Raheem, A.; Hussain, I. A Scheme for Obtaining Secure S-Boxes Based on Chaotic Baker's Map. *3D Research* **2014**, *5*, 17. [[CrossRef](#)]
55. Khan, M.; Shah, T.; Mahmood, H.; Gondal, M.A.; Hussain, I. A Novel Technique for the Construction of Strong S-boxes based on Chaotic Lorenz Systems. *Nonlinear Dyn.* **2012**, *70*, 2303–2311. [[CrossRef](#)]
56. Khan, M.; Shah, T.; Batool, S. A New Implementation of Chaotic S-boxes in CAPTCHA. *Signal Image Video Process.* **2015**, *10*, 293–300. [[CrossRef](#)]
57. Hussain, I.; Shah, T.; Gondal, M.A.; Mahmood, H. An Efficient Approach for the Construction of LFT S-boxes using Chaotic Logistic Map. *Nonlinear Dyn.* **2013**, *71*, 133–140. [[CrossRef](#)]
58. Ali, R.S.; Akif, O.Z.; Jassim, S.A.; Farhan, A.K.; El-Kenawy, E.S.M.; Ibrahim, A.; Ghoneim, M.E.; Abdelhamid, A.A. Enhancement of the CAST Block Algorithm Based on Novel S-Box for Image Encryption. *Sensors* **2022**, *22*, 8527.
59. Wang, Y.; Yang, L.; Li, M.; Song, S. A Method for Designing S-box based on Chaotic Neural Network. In Proceedings of the Sixth International Conference on Natural Computation (ICNC), Yantai, China, 10–12 August 2010; Volume 2, pp. 1033–1037.
60. Noughabi, M.; Sadeghiyan, B. Design of S-boxes based on Neural Networks. In Proceedings of the International Conference on Electronics and Information Engineering (ICEIE), Kyoto, Japan, 1–3 August 2010; Volume 2, pp. V2-172–V2-178.
61. Zaibi, G.; Kachouri, A.; Peyrard, F.; Fournier-Prunaret, D. On Dynamic Chaotic S-box. In Proceedings of the Global Information Infrastructure Symposium, Hammamet, Tunisia, 23–26 June 2009; pp. 1–5.
62. Ahmad, M.; Rizvi, D.R.; Ahmad, Z. PWLCM-Based Random Search for Strong Substitution-Box Design. In *Proceedings of the Second International Conference on Computer and Communication Technologies: IC3T 2015*; Springer: New Delhi, India, 2016; Volume 1, pp. 471–478. [[CrossRef](#)]
63. Tang, G.; Liao, X. A Method for Designing Dynamical S-boxes based on Discretized Chaotic Map. *Chaos Solitons Fractals* **2005**, *23*, 1901–1909.
64. Rîncu, C.I.; Iana, V.G. S-box Design based on Chaotic Maps Combination. In Proceedings of the 10th International Conference on Communications (COMM), Bucharest, Romania, 29–31 May 2014; pp. 1–4. [[CrossRef](#)]
65. Anees, A.; Ahmed, Z. A Technique for Designing Substitution Box Based on Van der Pol Oscillator. *Wirel. Pers. Commun.* **2015**, *82*, 1497–1503.
66. Millan, W. How to Improve the Nonlinearity of Bijective S-boxes. In *Information Security and Privacy; Lecture Notes in Computer Science*; Boyd, C., Dawson, E., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1438, pp. 181–192.
67. Wu, Y.; Noonan, J.P.; Agaian, S.S. Dynamic and Implicit Latin Square Doubly Stochastic S-boxes with Reversibility. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Anchorage, AL, USA, 9–12 October 2011; pp. 3358–3364. [[CrossRef](#)]
68. Radhakrishnan, S.V.; Subramanian, S. An Analytical Approach to S-box Generation. *Comput. Electr. Eng.* **2013**, *39*, 1006–1015.
69. Picek, S.; Ege, B.; Batina, L.; Jakobovic, D.; Chmielewski, L.; Golub, M. On Using Genetic Algorithms for Intrinsic Side-channel Resistance: The Case of AES S-box. In Proceedings of the First Workshop on Cryptography and Security in Computing Systems, Vienna, Austria, 20 January 2014; ACM: New York, NY, USA, 2014; pp. 13–18.

70. Ivanov, G.; Nikolov, N.; Nikova, S. Reversed Genetic Algorithms for Generation of Bijective S-Boxes with Good Cryptographic Properties. Cryptology ePrint Archive, Report 2014/801. 2014. Available online: <https://eprint.iacr.org/2014/801> (accessed on 30 July 2022). [CrossRef]
71. Clark, J.A.; Jacob, J.L.; Stepney, S. The Design of S-boxes by Simulated Annealing. *New Gen. Comput.* **2005**, *23*, 219–231. [CrossRef]
72. Kuznetsov, A.; Wieclaw, L.; Poluyanenko, N.; Hamera, L.; Kandiy, S.; Lohachova, Y. Optimization of a Simulated Annealing Algorithm for S-Boxes Generating. *Sensors* **2022**, *22*, 6073.
73. Szaban, M.; Seredynski, F. Cryptographically Strong S-Boxes based on Cellular Automata. In *Cellular Automata; Lecture Notes in Computer Science*; Umeo, H., Morishita, S., Nishinari, K., Komatsuzaki, T., Bandini, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5191, pp. 478–485. [CrossRef]
74. Szaban, M.; Seredynski, F. Designing Cryptographically Strong S-boxes with the use of Cellular Automata. *Ann. UMCS Inf.* **2009**, *8*, 27–41. [CrossRef]
75. Ahmad, M.; Bhatia, D.; Hassan, Y. A Novel Ant Colony Optimization Based Scheme for Substitution Box Design. *Procedia Comput. Sci.* **2015**, *57*, 572–580.
76. Ivanov, G.; Nikolov, N.; Nikova, S. Cryptographically Strong S-Boxes Generated by Modified Immune Algorithm. In *Cryptography and Information Security in the Balkans: Second International Conference, BalkanCryptSec 2015, Koper, Slovenia, 3–4 September 2015; Revised Selected Papers*; Springer International Publishing: Cham, Switzerland, 2016; pp. 31–42. [CrossRef]
77. Isa, H.; Jamil, N.; Z'aba, M.R. Construction of Cryptographically Strong S-Boxes Inspired by Bee Waggle Dance. *New Gen. Comput.* **2016**, *34*, 221–238.
78. Sikdar, D. S-box Optimization Technique with a Primitive Irreducible Polynomial. *Int. J. Emerg. Trends Technol. Comput. Sci.* **2014**, *3*, 97–99. [CrossRef]
79. Khan, M.; Azam, N.A. S-Boxes based on Affine Mapping and Orbit of Power Function. *3D Research* **2015**, *6*, 12.
80. Yang, M.; Wang, Z.; Meng, Q.; Han, L. Evolutionary Design of S-box with Cryptographic Properties. In Proceedings of the Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops (ISPAW), Busan, Republic of Korea, 26–28 May 2011; pp. 12–15.
81. Chun, Y.; Guo, Y. A Research and Improvement Based on Rijndael Algorithm. In Proceedings of the 1st International Conference on Information Science and Engineering (ICISE), Nanjing, China, 26–28 December 2009; pp. 1585–1588.
82. Cui, J.; Huang, L.; Zhong, H.; Chang, C.; Yang, W. An Improved AES S-box and Its Performance Analysis. *Int. J. Innov. Comput. Inf. Control* **2011**, *7*, 2291–2302.
83. Kumar, A.; Tiwari, N. AES Security Enhancement by Using Double S-Box. *Int. J. Comput. Sci. Inf. Technol.* **2012**, *3*, 3980–3984. [CrossRef]
84. Kapalova, N.; Sakan, K.; Algazy, K.; Dyusenbayev, D. Development and Study of an Encryption Algorithm. *Computation* **2022**, *10*, 198.
85. Hussain, I.; Shah, T.; Gondal, M.A.; Khan, M.; Khan, W.A. Construction of New S-box using a Linear Fractional Transformation. *World Appl. Sci. J.* **2011**, *14*, 1779–1785. [CrossRef]
86. Hussain, I.; Shah, T.; Gondal, M.A.; Khan, W.A.; Mahmood, H. A Group Theoretic Approach to Construct Cryptographically Strong Substitution Boxes. *Neural Comput. Appl.* **2013**, *23*, 97–104. [CrossRef]
87. Hussain, I.; Shah, T.; Mahmood, H.; Gondal, M.A. A Projective General Linear Group based Algorithm for the Construction of Substitution Box for Block Ciphers. *Neural Comput.* **2013**, *22*, 1085–1093.
88. Jin, S.Y.; Baek, J.M.; Song, H.Y. Improved Rijndael-Like S-Box and Its Transform Domain Analysis. In *Sequences and Their Applications—SETA 2006: 4th International Conference, Beijing, China, 24–28 September 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 153–167.
89. Tran, M.T.; Bui, D.K.; Duong, A.D. Gray S-box for Advanced Encryption Standard. In Proceedings of the International Conference on Computational Intelligence and Security (CIS'08), Suzhou, China, 13–17 December 2008; Volume 1, pp. 253–258.
90. Dumas, J.G.; Orfila, J.B. Generating S-Boxes from Semi-fields Pseudo-Extensions. *arXiv* **2014**, arXiv:1411.2503.
91. Li, Y.; Wang, M. Constructing Differentially 4-Uniform Permutations over $GF(2^{2m})$ from Quadratic APN Permutations over $GF(2^{2m+1})$. *Des. Codes Cryptogr.* **2014**, *72*, 249–264. [CrossRef]
92. Fuller, J.; Millan, W.; Dawson, E. Multi-Objective Optimisation of Bijective S-boxes. *New Gen. Comput.* **2005**, *23*, 201–218.
93. Isa, H.; Jamil, N.; Z'aba, M.R. Improved S-Box Construction from Binomial Power Functions. *Malays. J. Math. Sci.* **2015**, *9*, 21–35.
94. Aslan, B.; Sakalli, M.T.; Bulus, E. Classifying 8-Bit to 8-Bit S-Boxes based on Power Mappings from the Point of DDT and LAT Distributions. In *Arithmetic of Finite Fields; Lecture Notes in Computer Science*; von zur Gathen, J., Imaña, J.L., Koç, c.K., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5130, pp. 123–133.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.