Symmetries of Nonlinearity Constraints

Venkatesh Ramamoorthy[1], Marius C.Silaghi[1], Toshihiro Matsui[2], Katsutoshi Hirayama[3], and Makoto Yokoo[4]

[1] Florida Institute of Technology, Melbourne, FL 32901, United States of America
vramamoo@my.fit.edu, msilaghi@cs.fit.edu
[2] Nagoya Institute of Technology, Nagoya, Aichi, 466-8555, Japan
matsui.t@nitech.ac.jp
[3] Kobe University, Kobe, 657-8501, Japan hirayama@maritime.kobe-u.ac.jp
[4] Kyushu University, Hakozaki 6-10-1, Higashi-ku, Fukuoka, 812-8581, Japan
yokoo@is.kyushu-u.ac.jp

Abstract. We find symmetries for constraints that model the nonlinearity requirements of a discrete function $f : \mathbb{Z}_{2^n} \rightarrow \mathbb{Z}_{2^m} (n > m)$. Such constraints are very important, as the functions are employed in generating deterministic but difficult-to-analyze permutations used in symmetric cryptographic systems. There, such functions are referred to as Substitution Boxes (S-boxes). The nonlinearity is a complex requirement that has been traditionally formulated using a set of criteria (that we interpret as new constraints). Most of these constraints are found to exhibit symmetries that can be exploited for reducing the size of the search space, and for efficiently generating new solutions. Among discovered symmetries, a bit inversion symmetry (a special case of the value reversal symmetry) and a rotational symmetry (a special case of variable symmetry) are found to apply to all studied nonlinearity constraints without affecting their security metric, and quadruple the efficiency of solvers. Theoretical and experimental results on symmetry are reported.

1 Introduction

Nonlinearity requirements are complex constraints occurring commonly in practice. The problem is to find a nonlinear discrete function $f : Z_{2^n} \rightarrow \mathbb{Z}_{2^m} (n > m)$. An example is the nonlinearity requirements of cryptographic substitution boxes (S-boxes) for the Substitution-Permutation networks proposed by Shannon [15], of which the Feistel architecture sub-family [8] is one of the most common. Simply put, the values of a set of variables should not be expressible or easily approximated by a linear relation. The more successful an approximated linearization is, the easier could an attacker analyze its function [10]. The designer of a cipher has to maximize the error in the closest linear approximation.

In [13] we report on how to represent non-linearity criteria using constraints while in [14], we discuss soft global nonlinearity constraint decomposition propagators. Here we detail symmetries found to characterize the nonlinearity constraints, and the way we exploit them.

Constraint Satisfaction Problems (CSPs) Constraint satisfaction is a framework that is famous for naturally modeling many different problems, ranging from task scheduling to the verification of security protocols [5]. A CSP is defined as a triplet $\langle X, D, C\rangle$ where X is a set of variables, D a set of domains containing possible values for each variable in X and C, a set of constraints that involve some or all of the variables in X.

A constraint is a predicate specifying the acceptable combinations of assignments for the variables that it involves. It defines a relations between these variables, being a subset of the Cartesian product of the domains of the variables. The number of variables involved in a given constraint are referred to as its arity. For example, a binary constraint involves only two variables. An n-ary constraint is one that involves n variables. A solution to the CSP is a set of values assigned to each variable in X from its domain in D such that all constraints in C are satisfied. A partial assignment comprises values assigned to a subset $X' \subseteq X$ of variables such that constraints involving only variables in X' are satisfied. A *soft* constraint (as opposed to a *hard* constraint) is defined as one in which there exist some solutions that do not have to satisfy this constraint. Security protocols have been modeled in the past using Soft CSPs [3].

Once a problem is modeled as a CSP, it can be solved using any of the efficient generic algorithms developed during the last few decades. However, insight can help formulate constraints in ways that make the work of generic solvers much easier. One of the most common approaches consists of identifying and exploiting symmetries in individual constraints, between certain variables, between value of certain variables, or in the problem as a whole. The search effort on symmetric areas of the search space is identical. Therefore it is sufficient to explore one of each pair of symmetrical sub-problems. The conclusions extracted from one sub-problems can be applied to any of its symmetrical sub-problems using the corresponding symmetry relation. The speed-up can be proportional with the number of identified symmetry relations.

In this paper, we identify particular features of the nonlinearity constraints occurring in the design of S-boxes, and use them to find and prove symmetry relations. The usefulness of the new symmetries is shown by studying their impact on the process of solving a well-known instance of the S-box design problem. A systematic CSP solver employing the proposed techniques is shown to yield S-boxes which are significantly better than those employed by *Triple*-Data Encryption Standard (3DES) [2], a widely-used cryptographic algorithm. 3DES employs eight S-boxes S_1 to S_8, with S_8 shown in Fig. 1. A 6×4 S-box substitution of 4 bits for a 6-bit input i is obtained by indexing into the row number formed by the first and last bits of i, and the column number formed by the remaining middle bits of i. For example, an input of 45 $(= 101101_2)$ to S-Box S_8 yields $8 = (1000_2)$, obtained by reading the entry in row 3 $(= 11_2)$, column 6 $(= 0110_2)$ of Fig. 1.

The S-boxes are so designed to satisfy criteria numbered **S-1**, **S-2**, and so on [7], which are listed in Table 1. Note that 3DES is one of the techniques currently used in protocols such as the Secure Sockets Layer (SSL), Transport Layer Security (TLS) that form the basis for the Internet protocol `https`. 3DES is also employed in the Secure Shell protocol used in applications such as `sftp` and `ssh`.

The detection of symmetry in constraints is an important research issue, as it enables one to avoid duplicating effort by exploring symmetric regions of the search space. To

S_1	$y_1y_2y_3y_4$															
y_0y_5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Fig. 1. S-box S_8 used in 3DES

limit the search to one among a set of symmetrical sub-problems one has to come out with new constraints rejecting all but one of these sub-problems.

Next, we introduce some of the more technical related background. The subsequent section introduces in detail the relevant concepts. Nonlinearity constraints are formulated, in conjunction with a theoretical exposition of heuristics and aspects of symmetry. Experiments are discussed, along with results and conclusions.

2 Concepts and New Properties

Here we first recapitulate on the nonlinearity constraints proposed in [13] and identify some new properties shown later to translate into useful symmetries.

Notations The notation $|x|$ represents the absolute value of a number x. For a set S, $|S|$ represents its cardinality while for a set expressed using braces, its cardinality is denoted by preceding the braces with a $\#$. The symbols $\cdot$ and $\oplus$ represent the bit-wise AND and exclusive-OR (XOR) operation respectively, on two identical-sized bit patterns. Bit pattern $\bar{x}$ denotes the one's-complement of x. For a bit b, $\bar{b} = b \oplus 1$. For any two bits (or identical-sized bit-patterns) a and b, $a \oplus b = b \oplus a$, and $a \oplus b = \bar{a} \oplus \bar{b}$.

A linear Boolean function $L_\omega(x)$ on an n-bit pattern $x = x_0 \ldots x_{n-1}$ selected by an n-bit pattern $\omega = \omega_0 \ldots \omega_{n-1}$ is defined [6] as:

$$L_\omega(x) = \omega_0 \cdot x_0 \oplus \ldots \oplus \omega_{n-1} \cdot x_{n-1} = \bigoplus_{i=0}^{n-1} \omega_i \cdot x_i \tag{1}$$

The *parity* $P(x)$ of an n-bit pattern $x = x_0x_1 \ldots x_{n-1}$ is equal to the exclusive-OR of the bits in x, that is, $P(x) = x_0 \oplus x_1 \oplus \ldots \oplus x_{n-1}$. Using these facts, the following property of $L_\omega(x)$ can be derived:

Property 1. $L_\omega(\bar{x}) = L_\omega(x) \oplus P(\omega)$

Proof. Let $L_\omega(x) = \omega_0 \cdot x_0 \oplus \omega_1 \cdot x_1 \oplus \ldots \oplus \omega_5 \cdot x_5$ Then,

$$\begin{aligned} L_\omega(\overline{x}) &= \omega_0 \cdot \overline{x_0} \oplus \omega_1 \cdot \overline{x_1} \oplus \ldots \oplus \omega_{n-1} \cdot \overline{x_{n-1}} \\ &= \omega_0 \cdot (x_0 \oplus 1) \oplus \omega_1 \cdot (x_1 \oplus 1) \oplus \ldots \oplus \omega_{n-1}(x_{n-1} \oplus 1) \\ &= (\omega_0 \cdot x_0 \oplus \omega_1 \cdot x_1 \oplus \ldots \oplus \omega_{n-1} \cdot x_{n-1}) \oplus (\omega_0 \oplus \omega_1 \oplus \ldots \oplus \omega_{n-1}) \\ &= L_\omega(x) \oplus P(\omega), \text{ from Equation 1.} \end{aligned}$$

S-1	Each S-box has six bits of input and four bits of output.
S-2	No output bit of an S-box should be too close to a linear function of the input bits.
S-3	If we fix the leftmost and rightmost input bits of the S-box and vary the four middle bits, each possible 4-bit output is attained exactly once as the middle four input bits range over their 16 possibilities.
S-4	If two inputs to an S-box differ in exactly one bit, the corresponding outputs must differ in at least two bits.
S-5	If two inputs differ in the two middle bits exactly, the outputs must differ in at least two bits.
S-6	If two inputs differ in the first two bits and are identical in the last two bits, the two outputs must be different.
S-7	For any nonzero 6-bit difference between inputs $\Delta I_{i,j}$, no more than eight of the 32 pairs of inputs exhibiting $\Delta I_{i,j}$ may result in the same output difference $\Delta O_{i,j}$.

Table 1. The nonlinearity criteria used by IBM for designing 3DES S-boxes [7]

$\mathbf{Q}.\mathbf{E}.\mathbf{D}.$

The *Hamming weight* of a bit pattern x, denoted by $wt(x)$, is equal to the number of **1**'s in x. The amount by which x and y differ, as mentioned in Table 1, equals $wt(x \oplus y)$.

	$i_1i_2i_3i_4$							
i_0i_5	0	1	2	3	...	13	14	15
0	x_0	x_2	x_4	x_6	...	x_{26}	x_{28}	x_{30}
1	x_1	x_3	x_5	x_7	...	x_{27}	x_{29}	x_{31}
2	x_{32}	x_{34}	x_{36}	x_{38}	...	x_{58}	x_{60}	x_{62}
3	x_{33}	x_{35}	x_{37}	x_{39}	...	x_{59}	x_{61}	x_{63}

Fig. 2. Diagrammatic relationship between the defined CSP variables and 6×4 S-box entries

The Search Space of Nonlinearity Constraints We now remind the nonlinearity constraints proposed in [13]. To model nonlinearity criteria [13] defines the set X of 2^n variables $X = \{x_0, x_1, \ldots, x_{2^n-1}\} = \{x_i | i \in \mathbb{Z}_{2^n}\}$, each representing an entry in the S-box. The domain $D_i \in D$ for each variable x_i is $\mathbb{Z}_{2^m} = \{0, 1, \ldots, 2^m - 1\}$. For the case of $n \times m$ S-boxes, the i^{th} variable x_i specifies the m-bit S-box output for an n-bit input i. Using the variables in X, a 6×4 S-box such as the ones used in 3DES, is organized as shown in Fig. 2, addressed by incrementing the input.

Nonlinearity Metrics for Variable Assignments Since for each input i the S-box returns the value of x_i, therefore the nonlinearity of the S-box can be stated as a nonlinearity between each index i and the value of x_i. The ability of expressing each bit of an m-bit value $e \in \mathbb{Z}_{2^m}$ in the assignment $x_i = e$, as a linear combination of the bits in the n-bit subscript $i \in \mathbb{Z}_{2^n}$ [10, 9], is now examined. Here, we use this measure as the score of a solution (to be optimized) and extend the definition to a partial assignment.

Consider an n-bit subscript $i = i_0 \dots i_{n-1}$ of a variable x_i, and a corresponding assignment to x_i of a value from $\mathbb{Z}_{2^m}$. The linear combinations to be checked for equality are obtained by selecting bits in i and the value assigned to x_i using selectors a and b respectively, $\forall a, b, 0 \leq a < 2^n$ and $0 \leq b < 2^m$. One denotes, by $L_\omega(x_i)$, the application of the function L_ω of Equation 1 on the value assigned to the CSP variable x_i. For a complete assignment Φ with all variables in X assigned, let $N_X^\Phi(a, b)$, quantifying the *success of linearization* of the relation between i to x_i using coefficients a and b, be:

$$N_X^\Phi(a,b) = \#\{i | x_i \in X; L_a(i) = L_b(x_i)\} \tag{2}$$

Observe that $0 \leq N_X^\Phi(a, b) \leq 2^n$.

Nonlinearity as a Probability Measure For *each* variable x_i corresponding to input i in a complete assignment Φ, given selectors a and b defined as above, $p(a, b)$ denotes the fraction of cases when $L_a(i) = L_b(x_i)$, computed as:

$$p(a,b) = \frac{N_X^\Phi(a,b)}{2^n} \tag{3}$$

$p(a, b) = 1$ is the condition where the linear combination of the bits in the value assigned to x_i selected by b equals a linear combination of the bits in i selected by a, i.e., $\forall i, L_a(i) = L_b(x_i)$. If $p(a, b)$ is equal to zero, the linear combination of the output bits selected by b is always equal to the negation of the linear combination of input bits selected by a. According to the nonlinearity requirement **S-2**, $p(a, b)$ should be near $\frac{1}{2}$.

Linear Approximation Table (LAT) The Linear Approximation Table [10] for a complete assignment is a $2^n \times 2^m$ matrix. Its rows are headed by selector a, $0 \leq a < 2^n$, and columns by selector b, $0 \leq b < 2^m$ (see Table 2). Each entry specifies the quantity $N_X^\Phi(a, b) - \frac{|X|}{2}$, with one entry in row a and column b representing an offset-ted measure of the correlation between the bits of x_i selected by b and the bits of i selected by a. As an example, for the 3DES S-box S_8, the first and last two rows of its LAT are in Table 2. The LAT of a solution is an arithmetic accumulation of individual contributions due to *each* variable assignment $x_i = e, i \in \mathbb{Z}_{2^n}, e \in \mathbb{Z}_{2^m}$. A contribution arising from an assignment $x_i = e$ is equal to $L_a(i) \oplus L_b(e) \oplus 1, a \in \mathbb{Z}_{2^n}, b \in \mathbb{Z}_{2^m}$, that is, 0 or 1. The offset quantity $\frac{|X|}{2}$ is subtracted from each entry in the LAT of a solution.

We now identify the following property, used in proving new constraint symmetries.

Property 2. Changing an assignment $x_i = e$ to the assignment $x_i = \bar{e}$ changes the LAT entry at index (a, b) by the amount $(-1)^{L_a(i) \oplus L_b(\bar{e})} \cdot P(b)$.

Proof. These follow from Property 1, and from the fact that the contribution of an assignment $x_i = e$ to each entry in the LAT equals $L_a(i) \oplus L_b(e) \oplus 1$. Therefore, the contribution lost due to removing $x_i = e$ is $[1 - L_a(i) \oplus L_b(e)]$, which based on Property 1 equals $[1 - L_a(i) \oplus L_b(\bar{e}) \oplus P(b)]$. The added contribution of $x_i = \bar{e}$ is $[1 - L_a(i) \oplus L_b(\bar{e})]$. The total impact is their difference:

$$L_a(i) \oplus L_b(\bar{e}) \oplus P(b) - L_a(i) \oplus L_b(\bar{e}) = (P(b))$$

b / a	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
62	0	-8	4	0	-2	2	-2	6	10	6	2	2	0	0	-4	0
63	0	-8	0	4	2	-2	-10	-2	-6	6	-2	6	-4	4	-4	0

Table 2. The Linear Approximation Table for the S-box S_8 of Fig. 1

Note that for any binary x and y, $x \oplus y - x = (1-2x) \cdot y = (-1)^x \cdot y$ since:

$$x \oplus y - x = \begin{cases} x - x = 0 = (-1)^x \cdot y & \text{if } y = 0 \\ 1 - x - x = 1 - 2x = \begin{cases} 1 = (-1)^x \cdot y & \text{if } x = 0 \\ -1 = (-1)^x \cdot y & \text{if } x = 1 \end{cases} & \text{for } y = 1. \end{cases}$$

In our case, $x = L_a(i) \oplus L_b(\bar{e})$ and $y = P(b)$, and we obtain the expression in the property.

$\mathbf{Q.E.D.}$

Note that a similar expression can be obtained if any pair of assignments $x_i = e$ and $x_{\bar{i}} = f$ are swapped into $x_{\bar{i}} = e$ and $x_i = f$.

Property 3. The impact of swapping a pair of assignments $x_i = e$ and $x_{\bar{i}} = f$ on the LAT entry at index (a, b) is: $\left[(-1)^{L_a(i) \oplus L_b(\bar{e})} + (-1)^{L_a(\bar{i}) \oplus L_b(\bar{f})}\right] \cdot P(a)$.

The proof is similar to the one for Property 2.

The Score of an Assignment The most effective linear approximation of a complete assignment Φ containing $|X|$ variables is obtained if, for some a and b, $\left|N_X^{\Phi}(a,b) - \frac{|X|}{2}\right|$ is maximal. To reduce the weakest point of the assignment Φ, we use the so-called *effectiveness of linearization* [11] as the optimization score:

$$\sigma_X(\Phi) = \max_{a,b} \left\{ \left| N_X^{\Phi}(a,b) - \frac{|X|}{2} \right| : 1 \le a < |X|; 1 \le b < |D| \right\} \tag{4}$$

A complete assignment with a smaller score is considered better. We look for $\underset{\Phi}{\operatorname{argmin}}(\sigma_X(\Phi))$. We now identify the following two properties as useful kinds of symmetries, employed as described later:

1. Invariance of the score of a complete assignment with respect to bit inversion (replacing S-box entries by their one's-complements)
2. Invariance of the score of a complete assignment with respect to S-box rotation by two right angles (interchanging two variables whose subscripts are one's-complements of each other).

Property 4. The score $\sigma_X(\Phi)$ of a complete assignment Φ does not change if all of its assigned values are replaced by their one's-complements, into an assignment $\bar{\Phi}$.

Proof. For each LAT entry corresponding to an even parity b, where $P(b) = 0$, the entry will not be changed because the contribution of each S-box assignment changing from a value e to $\bar{e}$ is 0 (see Property 2).

For each LAT entry corresponding to an odd parity b, where $P(b) = 1$, all the assignments that were correctly linearized with their previous value e will be incorrectly linearized with the new assignment $\bar{e}$, and vice-versa. Therefore in this case $N_X^{\bar{\Phi}}(a,b) = |X| - N_X^{\Phi}(a,b)$, and:

$$\sigma_X(\bar{\Phi}) = \max_{a,b}\left\{\left||X| - N_X^{\Phi}(a,b) - \frac{|X|}{2}\right|\right\} = \max_{a,b}\left\{\left|\frac{|X|}{2} - N_X^{\Phi}(a,b)\right|\right\} = \sigma_X(\Phi).$$

Q.E.D.

Property 5. The score $\sigma_X(\Phi)$ of a complete assignment Φ does not change if all the values assigned to variables are reassigned to variables having subscripts equal to the one's-complements of the corresponding original variables, into an assignment $\hat{\Phi}$.

Proof. The proof is very similar to that for Property 4. For each LAT entry corresponding to an even parity a, where $P(a) = 0$, the entry will not be changed because the contribution of each S-box assignment $x_i = e$ getting reassigned to $x_{\bar{i}} = e$, is 0 (see Property 3).

For each LAT entry corresponding to an odd parity a, where $P(a) = 1$, all the assignments that were correctly linearized with their previous value $x_i = e$ will be incorrectly linearized with the new assignment $x_{\bar{i}} = e$, and vice-versa. Therefore in this case $N_X^{\hat{\Phi}}(a,b) = |X| - N_X^{\Phi}(a,b)$, and:

$$\sigma_X(\hat{\Phi}) = \max_{a,b}\left\{\left||X| - N_X^{\Phi}(a,b) - \frac{|X|}{2}\right|\right\} = \max_{a,b}\left\{\left|\frac{|X|}{2} - N_X^{\Phi}(a,b)\right|\right\} = \sigma_X(\Phi).$$

Q.E.D.

3 Constraints and their symmetries

A soft global constraint was formulated [13] to model nonlinearity requirements of the kind presented in Table 1 for S-boxes.

The Soft n-ary Global Constraint Modeling the nonlinearity requirement of the kind specified in Table 1 for a 6×4 S-box, namely, criteria **S-2**, leads to a soft constraint that minimizes $\sigma_X(\Phi)$. When used as a hard constraint for a threshold τ, it becomes:

$$\sigma_X(\Phi) \leq \tau \tag{5}$$

Due to Property 4 of the score of a complete assignment Φ, this soft global n-ary constraint possesses bit-inversion symmetry. Property 5 similarly ensures that this constraint possesses rotational symmetry.

The `AllDiff` *Constraints for* **S-3***:* For a 6×4 S-box, **S-3** is directly expressible as a set of `Alldiff` constraints [12, 13]:

$$\texttt{Alldiff}(x_0, x_2, x_4, ..., x_{30}), \texttt{Alldiff}(x_1, x_3, x_5, ..., x_{31})$$
$$\texttt{Alldiff}(x_{32}, x_{34}, x_{36}, ..., x_{62}), \texttt{Alldiff}(x_{33}, x_{35}, x_{37}, ..., x_{63})$$

This constraint possesses variable and value symmetry.

The Binary Constraints With 6×4 S-boxes, for any two 4-bit variables x_i and x_j corresponding to 6-bit subscripts i and j, $x_i, x_j \in D$, [13] models requirements **S-4**, **S-5** and **S-6** listed in Table 1, with binary constraints.

S-4 is modeled into the following binary constraint:

$$(\forall i)(\forall j)(0 \leq i < j \leq 63) \wedge wt(i \oplus j) = 1 \Rightarrow wt(x_i \oplus x_j) \geq 2$$

For an $n \times m$ S-box, the number of binary constraints for this criterion is equal to $n \times 2^{n-1}$. This constraint possesses row symmetry. For example, if Row 1 and Row 2 of the example 6×4 S-box of Fig. 2 are interchanged and *simultaneously*, Rows 3 and 4 are interchanged, **S-4** is still satisfied.

S-5 is modeled into the following binary constraint:

$$(\forall i)(\forall j)(0 \leq i, j \leq 63) \wedge (i \neq j) \wedge (i \oplus j = 001100_2) \Rightarrow wt(x_i \oplus x_j) \geq 2$$

For an $n \times m$ S-box, the number of binary constraints for this criterion is equal to 2^{n-1} when n is even. This criterion possesses column symmetry but only across specified columns. With reference to the example 6×4 S-box of Fig. 2, upon interchanging Column 0 with 6, 1 with 7, 2 with 4, 3 with 5, 8 with 14, 9 with 15, 10 with 12, and 11 with column 13, **S-5** is still satisfied.

S-6 is modeled into the following binary constraint:

$$(\forall i)(\forall j)(0 \leq i < j \leq 63), (i \oplus j) \wedge 110011_2 = 110000_2 \Rightarrow x_i \neq x_j$$

For an $n \times m$ S-box, the number of binary constraints for this requirement is equal to $(n-2) \times 2^{n-1}$ where $n \geq 4$. This criterion possesses *diagonal* symmetry. If in the example 6×4 S-box of Fig. 2, the rectangle formed by Rows 0-1 and Columns 0-7 is interchanged with its diagonally-opposite rectangle formed by Rows 2-3 and Columns 8-15, and *simultaneously*, the other two rectangles be interchanged, the resulting arrangement satisfies **S-6**.

Since $y_i \oplus y_j = \bar{y_i} \oplus \bar{y_j}$, all constraints for **S-4**, **S-5** and **S-6** possess bit inversion symmetry, that is, they are satisfied by solutions having their bits inverted.

Rotating the S-box by two right angles results is equivalent to interchanging variables x_i and $x_{\bar{i}}$ for all permissible values of i. Since $i \oplus j = \bar{i} \oplus \bar{j}$, one can observe, upon replacing i with $\bar{i}$ and j with $\bar{j}$ in the constraints for **S-4**, **S-5** and **S-6**, that they do not change, for all permissible i, j. In other words, these constraints are satisfied by solutions obtained by two rotations of each original solution.

Table 3 summarizes the constraints that are violated when the row, column, or diagonal interchanges are not simultaneously applied for separate pairs of row, columns, and diagonals, respectively.

Type of Conditional Symmetry	Criteria violated by Remaining Assignments
Row (**S-4**)	**S-4**
Column (**S-5**)	**S-4**
Diagonal (**S-6**)	

Table 3. Conditional Symmetries

The n-ary Global Constraint The last constraint that we have to investigate for symmetries is the one proposed in [13] for **S-7**. $O_7 = \left\{(x_i, x_{2^n-1-i}) : 0 \leq i < 2^{n-1}\right\}$ are pairs of variables corresponding to pairs of subscripts $(i, 2^n - 1 - i)$, that differ by n bits with $|O_7| = 2^{n-1}$. The requirement applies to m-bit differences $d = x_i \oplus x_{2^n-1-i}$, $0 \leq d < 2^m$. $f : \mathbb{Z}_{2^m} \rightarrow \mathbb{Z}_{2^{n-1}}$ denotes a *count* function, with $f(d)$ signifying the *frequency of occurrence* of an m-bit number $d = x_i \oplus x_{2^n-1-i}$ where $(x_i, x_{2^n-1-i}) \in O_7$, $0 \leq i < 2^{n-1}$. $\Sigma_{i=0}^{2^{n-1}-1} f(x_i \oplus x_{2^n-1-i}) = 2^{n-1}$.

According to this requirement, at most eight elements in O_7 should evaluate to the same m-bit difference d. This requirement is modeled as a global, n-ary, Boolean constraint in the following way:

$$\bigwedge_{i=0}^{2^{n-1}-1} (f(x_i \oplus x_{2^n-i-1}) \leq 8) \tag{6}$$

This constraint possesses bit inversion symmetry. Since $x_i \oplus x_{2^n-i-1} = x_{2^n-i-1} \oplus x_i$, these pairs of variables can be interchanged. But x_{2^n-i-1} is the same as $x_{\bar{i}}$, this implies x_i and $x_{\bar{i}}$ are interchangeable, or in other words, **S-7** possesses rotational symmetry.

4 Results

The experimentation setup consists of an Intel Pentium Core-2 Duo 3-GHz CPU, 3.3 GB RAM and GNU/Linux Ubuntu 9.04 operating system. We had modeled the constraints using the programming language Mozart-Oz [1]. However we quickly discovered infeasibility of modeling nonlinearity constraints of the kind specified in **S-2**. For this reason, the aforementioned nonlinearity requirements are precompiled and input to the Maintenance of Arc Consistency (MAC) with AC2001 solver [4]. The soft constraint in Equation 5 is transformed into a hard constraint by setting the threshold τ. The maximum value for τ is equal to $\frac{|X|}{2}$ while the "worst" score for a 3-DES S-box is equal to 18, for S-box S_7 [13]. We have experimented with lesser values of τ signifying "better" S-box scores, namely, with $\tau = 16$ and $\tau = 10$. Using the constraints methodology for automatic generation of S-boxes, we have obtained S-boxes having Matsui's score equal to 8, "better" than the "best" 3-DES S-box S_4 that equals 10 [14]. Propagators that have resulted in generation of these better-quality S-boxes are reported in [14] for criteria **S-2** and **S-7**.

We have experimentally confirmed our theoretical results on bit inversion and rotational symmetry, on all standard 3DES S-boxes as well as S-boxes generated by our solver (by verifying that bit-inverted S-boxes yield the same scores as the originals). By

breaking the bit inversion symmetry, the search space (computed as the cross-product of the domains) is halved. Computationally, this is expected to approximately halve the effort needed to exhaust this search space.

To break this symmetry in the 6×4 S-box CSP solver, we restrict the domain of x_0 to $\{0, 1, 2, 3, 4, 5, 6, 7\}$. Each result Φ specifies two S-boxes: Φ and $\bar{\Phi}$. By breaking the rotational symmetry, the search space is further halved to yield two more S-boxes: Φ and $\hat{\Phi}$ for each result Φ.

For the standard 3DES S-boxes, we have similarly confirmed the theoretically discovered row, column, and diagonal symmetry properties. For each S-box, an average of three new S-boxes have been obtained, due to these conditional symmetries.

In future work, we plan on using these symmetries by avoiding checking of the symmetrical constraints.

5 Conclusions

We have analyzed the nonlinearity constraints proposed in [13] and have used the obtained insights to prove new properties that translate into a set of constraint symmetries.

We have identified *bit inversion symmetry*, a special case of value reversal symmetry, demonstrating that they apply to all studied (soft and hard) S-box nonlinearity constraints. Bit inversion symmetry is obtained by replacing values assigned to variables by their one's-complements. This symmetry doubles the efficiency of S-box generation, and halves the search space.

We have identified a second form of symmetry, namely, *rotational symmetry* that is a special case of variable symmetry. This is achieved by interchanging variables whose suffixes differ in all bits in their binary representations. We have shown that all (soft and hard) S-box constraints possess rotational symmetry. The efficiency of S-box generation is further doubled by this symmetry, and the search space is halved further.

The first of three S-box nonlinear, binary constraints satisfies row symmetry, the second satisfies column and the third, diagonal symmetry. For these symmetries, appropriate rows, columns and diagonal-quadrants of a solution organized in the manner of Fig. 2 should be interchanged *simultaneously* to preserve the respective constraints. The n-ary global nonlinearity constraint possesses a restricted kind of diagonal symmetry, the investigation of which could form an extension to this work.

Together, the identified set of symmetries, can potentially generate up to 32 solutions from each given S-box.

References

1. The mozart-oz programming system, http://www.mozart-oz.org
2. Data encryption standard (DES). Federal Information Processing Standard 46-2 (January 1988)
3. Bella, G., Bistarelli, S.: Soft constraints for security protocol analysis: Confidentiality. In: PADL 2001. pp. 108–122 (2001)
4. Bessière, C., Régin, J.C.: Refining the basic constraint propagation algorithm. In: Nebel, B. (ed.) IJCAI. pp. 309–315. Morgan Kaufmann (2001)

5. Bistarelli, S.: Semirings for Soft Constraint Solving and Programming, Lecture Notes in Computer Science, vol. 2962. Springer (2004)
6. Clark, J., Jacob, J., Maitra, S., Stanica, P.: Almost boolean functions: the design of boolean functions by spectral inversion. Evolutionary Computation 3, 2173–2180 Vol.3 (Dec 2003)
7. Coppersmith, D.: The data encryption standard (des) and its strength against attacks. IBM J. Res. Dev. 38(3), 243–250 (1994)
8. Feistel, H.: Cryptography and computer privacy 228, 15–23 (1973)
9. Heys, H.M.: A tutorial on linear and differential cryptanalysis. Cryptologia XXVI(3), 189–221 (2002)
10. Matsui, M.: Linear cryptanalysis method for des cipher. In: EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology. pp. 386–397. Springer-Verlag (1994)
11. O'Connor, L.: Properties of linear approximation tables 1008 (1995)
12. Puget, J.F.: A fast algorithm for the bound consistency of alldiff constraints. In: AAAI '98. pp. 359–366 (1998)
13. Ramamoorthy, V., Silaghi, M., Matsui, T., Hirayama, K., Yokoo, M.: The design of cryptographic S-Boxes using CSPs. In: CP (2011), to appear
14. Ramamoorthy, V., Silaghi, M., Matsui, T., Hirayama, K., Yokoo, M.: Soft nonlinearity constraints and their lower-arity decomposition. In: CP (2011), workshop on Soft Constraints (SofT 2011). To appear
15. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal 28, 656–715 (1949)