Appendix A
Monoids, Groups, Rings and Fields

Information theory is the science of communication in the presence of noise.
- C.E. Shannon
Cryptology is the science of communication in the presence of adversaries.
- R.L. Rivest

A *monoid*, denoted $(S, \cdot)$, consists of a non-empty set S and a binary operation $\cdot$ defined on S satisfying the following properties:

(1) associativity: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all $a, b, c \in S$
(2) identity: there exist an element $1 \in S$ such that $a \cdot 1 = 1 \cdot a = a$ for all $a \in S$

An example of a monoid is the set $\mathbf{N}$ of natural numbers, with the addition operation. The neutral element is 0. This same set under multiplication has neutral element 1.

A *group*, denoted $(S, +)$, is composed of a non-empty set S and a binary operation, denoted $+$, defined on S. Groups can be either finite or infinite depending on the number of elements in S. The number of distinct elements in a group is called the *order of the group*.

The fundamental properties of a group are:

(3) *closure* under $+$: $a + b \in S$ for all $a, b \in S$.
(4) $(S, +)$ is a monoid.
(5) additive *inverse*: for each $x \in S$, there is an element $-x \in S$ such that $x + (-x) = (-x) + x = 0$.

If the following property holds, then $(S, +)$ is called a commutative or Abelian group, otherwise it is non-Abelian:

(6) *commutativity* of $+$: $a + b = b + a$ for all $a, b \in S$.

An example of a (finite) Abelian group is $(\mathbb{Z}_n, +)$, where

$$\mathbb{Z}_n = \{x \in \mathbb{Z} | 0 \leq x \leq n - 1\},$$

where $n > 1$ is a positive integer. The set $\mathbb{Z}$ of integers under multiplication, denoted $(\mathbb{Z}, *)$, is an example of an infinite Abelian group.

An example of a non-Abelian group is the set of invertible square matrices with entries in $\mathbb{Z}$, under the multiplication operation. It is an infinite non-Abelian group.

J. Nakahara Jr., *Lai-Massey Cipher Designs*, https://doi.org/10.1007/978-3-319-68273-0

In contrast, the group of permutations of n elements, for fixed $n > 1$, under composition of mappings is a finite non-Abelian group.

A *ring*, denoted $(S, +, *)$, is a group under two operations usually denoted $+$ (addition) and $*$ (multiplication). Besides properties (1)-(5), a ring satisfies the following properties:

(7) *closure* under $*$: $a * b \in S$ for all $a, b \in S$.
(8) *associativity* of $*$: $a * (b * c) = (a * b) * c$ for all $a, b, c \in S$.
(9) left- and right-*distributive* laws: $a * (b + c) = a * b + a * c$ and $(a + b) * c = a * c + b * c$ for all $a, b, c \in S$.

If both operations in a ring are commutative, then the ring is a commutative or an Abelian ring.

Examples of Abelian rings: $\mathbb{Z}[x]$, the set of polynomials in x with integer coefficients, is an infinite ring; the set $\mathbb{Z}_n^d[x]$ of polynomials of degree d with coefficients in $\mathbb{Z}_n$ (residue class ring or factor ring) for fixed integers d and n is a finite ring.

Examples of non-Abelian rings: the set $\mathrm{GL}(k, \mathbb{Z})$, the *general linear group* of invertible $k \times k$ matrices with entries from $\mathbb{Z}$, is an infinite ring; $\mathrm{GL}(k, \mathbb{Z}_n)$, the set of invertible $k \times k$ matrices with values in Z_n, for fixed n, is a finite ring.

A *field*, denoted $(F, +, *)$, consists of a non-empty set F and two binary operations, usually denoted $+$ (addition) and $*$ (multiplication) defined on F with the following properties:

- $(F, +, *)$ is a ring.
- commutativity of $*$: $a * b = b * a$ for all $a, b \in F$.
- multiplicative identity: there exist $1 \in F$ such that $a * 1 = 1 * a = a$ for all $a \in F$.
- multiplicative inverse: if $a \in F$ and $a \neq 0$, then there exist $b \in F$ such that $a * b = b * a = 1$.

Examples of infinite fields: $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ with the addition $(+)$ and multiplication $(*)$ operations in each field.

Examples of finite fields: (i) Z_q with q a prime number; (ii) $\mathrm{GF}(2^n) = \mathrm{GF}(2)[\mathrm{x}]/(p(x))$ where n is a positive integer and $p(x)$ is a primitive polynomial of degree n over $\mathrm{GF}(2)$.

The question whether all elements are invertible is the distinguishing feature between a *group* and a *monoid*, or between a *field* and a *ring*.

Appendix B
Differential and Linear Branch Number

Research is to see what everybody else has seen, and to think what nobody else has thought.
–Albert Szent-Györgi

Definition B.1. [55]
The *differential branch number* of a linear transformation θ is defined as

$$B(\theta) = \min_{a \neq 0}\{\mathrm{HW}(a) + \mathrm{HW}(\theta(a))\},$$

where $\mathrm{HW}(a)$ is the Hamming Weight of the vector a in the underlying vector space.

The Hamming distance between two vectors u and v from the n-dimensional vector space $\mathrm{GF}(2^m)^n$ is the number of m-bit coordinates where u and v differ.

The Hamming weight, $\mathrm{HW}(u)$ of an element $u \in \mathrm{GF}(2^m)$ is the Hamming Distance between u and the null vector of $\mathrm{GF}(2^m)^n$, that is, the number of non-zero components of u.

Definition B.2. [55]
A linear $[n, k, d]$-code $\mathcal{C}$ over $\mathrm{GF}(2^m)$ is a k-dimensional subspace of the vector space $\mathrm{GF}(2^m)^n$, where two different vectors of the subspace have a Hamming distance of at least d, and d is the largest number with this property.

A linear code $\mathcal{C}$ can be described by a generator matrix G, which is a $k \times n$ matrix whose rows form a vector space basis for the code. Since the choice of a basis in a vector space is not unique, a code has many different generator matrices that can be reduced to one another by elementary row operations and column permutations. The *echelon (or standard) form* of the generator matrix is the following:

$$G_e = [I_{k\times k} \| A_{k\times(n-k)}],$$

where $I_{k\times k}$ is the identity matrix of order k.

The dual code $\mathcal{C}^t$ of a code $\mathcal{C}$ is defined as the set of vectors that are orthogonal to all the vectors of $\mathcal{C}$: $\mathcal{C}^t = \{x| < x, y >= 0, \forall y \in \mathcal{C}\}$, where $< x, y >$ is the dot product between the vectors x and y.

J. Nakahara Jr., *Lai-Massey Cipher Designs*, https://doi.org/10.1007/978-3-319-68273-0

The *differential branch number* of a linear mapping can be related to the minimal distance of the associated linear code.

Definition B.3. [55]
Let θ be a linear mapping from $GF(2^m)^n$ to $GF(2^m)^n$. The associated code of θ, $\mathcal{C}_\theta$, is the linear code that has codewords given by the vectors $(x|\theta(x))$. The code $\mathcal{C}_\theta$ has 2^n codewords and has length $2n$.

It follows from the definition that the *differential branch number* of a mapping equals the minimal distance between two different codewords of its associated code. The upper bound on the differential branch number of a mapping corresponds to the Singleton bound on the minimal distance of a linear code. This relation between linear transformations and linear codes allows to efficiently construct mappings with a high differential branch number. Given a linear code $\mathcal{C}_\theta$, the associated mapping θ is given by $\theta(x) = x \cdot A$, where A is found from the echelon form of the generator matrix of $\mathcal{C}$. It can be proved that the *linear branch number* of a mapping θ is equal to the minimal distance of the dual code of $\mathcal{C}_\theta$.

From a linear cryptanalysis perspective, an analogous quantity can be defined:

Definition B.4. (Linear Branch Number)
The *linear branch number* of a transformation ϕ with respect to x is given by

$$L(\phi, x) = \min_{x \neq 0}\{HW(x) + HW(\phi(x))\}.$$

B.1 MDS Codes

A linear code over the Galois Field GF(2^m) is denoted by an $[n, k, d]$-code, where n is the symbol length of the encoded message, k is the length of the original message and d is the minimal symbol distance between any two encoded messages.

Definition B.5. [55][chap.11]
Linear $[n, k, d]$-codes obey the Singleton bound: $d \leq n - k + 1$. A code that meets the bound, $d = n - k + 1$, is called a *Maximum Distance Separable (MDS)* code. A linear $[n, k, d]$-code $\mathcal{C}$ with generator matrix $G = [I_{k\times k} \| A_{k\times(n-k)}]$ is MDS if and only if every square submatrix formed from i rows and i columns of A, for $i \in \{1, \ldots, \min(k, n-k)\}$, is nonsingular.

Consequently, by definition, each square submatrix of an MDS matrix is MDS as well. Therefore, a single $n \times n$ MDS matrix automatically provides several $m \times m$ MDS matrices for all $1 < m < n$. These $m \times m$ MDS matrices are not customized though, that is, their coefficients may not be small nor lead necessarily to an efficient implementation. On the other hand, this simple

observation allows one to find several MDS matrices as submatrices whose dimensions are not powers of 2 (the vast majority of MDS matrices used in cipher constructions in the literature have dimensions which are powers of 2).

Both the AES and the Twofish ciphers use $[8, 4, 5]$ MDS codes that is: $n = 8$, $k = 4$ and $d = n - k + 1 = 8 - 4 + 1 = 5$.

Known MDS codes are Reed-Solomon codes, (3,1) Hamming codes, (4,1) extended Hamming codes and dual MDS codes [55].

MDS codes are used to provide perfect diffusion. An earlier concept, closely related to MDS codes, is that of a multipermutation.

Definition B.6. [80]
A permutation $f : V^2 \to V^2$ defined as $f(x, y) = (f_1(x, y), f_2(x, y))$, is a *multipermutation* if for every $x, y \in V$, the mappings $f_i(a, *)$, $f_i(*, b)$ for $i \in \{1, 2\}$ are permutations on V.

Vaudenay in [92] generalized the concept of multipermutation to that of an (r, t)-multipermutation.

Definition B.7. [92]
A function $f : V^r \to V^t$ is an (r, t)-multipermutation if any two distinct $(r + t)$-tuples of the form $(x_1, \ldots, x_r, f(x_1, \ldots, x_r))$ cannot collide in any r positions.

The perfect diffusion with (r, t)-multipermutations is achieved because for any $x_1, \ldots, x_r \in V$ and any integer i such that $1 \leq i \leq r$, changing any i input values will change at least $t - i + 1$ output values of f.

From [64] and using Def. B.5, the set of all words of the form $(x_1, \ldots, x_r, f(x_1, \ldots, x_r))$ can be seen as a systematic error-correcting code of $|V|^r$ words of length $r + t$ with minimal distance $t + 1$, which matches the Singleton bound. The connection between multipermutations and MDS codes comes from the fact that if V is a finite field, a linear (r, t)-multipermutation is a $[r+t, r, t+1]$ MDS code in the sense that any word of length r is coded by the concatenation of the word and its multipermutation image. Equivalence holds only when the MDS code minimal distance is at least r, with $t+1 \geq r$. Linear (r, t)-multipermutations f can be represented using a $r \times t$ MDS matrix M as $f : x \to M \cdot x$.

MDS matrices form a pervasive component not only in modern block ciphers such as AES [33], BKSQ [27] and SHARK [74], but also in stream ciphers such as MUGI [94] as well as in hash functions such as Grostl [34] and Maelstrom [32].

Although MDS codes appear in many cipher designs, PES [49], IDEA [50], MESH [67], Keccak [10], PRESENT [15] and Serpent [3] are examples of modern cryptographic algorithms that do not use MDS matrices at all in their design.

MDS matrices do not need to be square matrices necessarily. Their use in SPN cipher designs though, requires that they be invertible, and thus, be square matrices, so that decryption is possible.

A convenient design criteria (for SPN cipher designs) is to impose that such matrices also be *involutory*. A square matrix X is *involutory* is $X^2 = I$ that is, $X = X^{-1}$, where I is the identity matrix. Otherwise, the matrix is non-involutory. If an involutory matrix is used as a cipher component in SPN designs, it means that both encryption and decryption will have the same (implementation) cost concerning this diffusion component.

For Lai-Massey cipher designs that incorporate MDS matrices, there is no need for the involutory property. For instance, the FOX cipher family uses non-involutory MDS matrices [40].

A square matrix S of order d is called *symmetric* if $A = A^T$, that is, A is identical to its transpose matrix, or equivalently, $a_{i,j} = a_{j,i}$ for all $0 \leq i, j < d$.

B.1.1 How to Construct MDS Matrices?

There are several known techniques to construct MDS matrices:

- **circulant matrices**: each row of a circulant matrix forms a cyclically rotated version of the other rows of the matrix. Similarly for the columns. An $m \times m$ circulant matrix C can be denoted

$$C = \begin{pmatrix} a_0 & a_1 & \dots & a_{m-1} \\ a_{m-1} & a_0 & \dots & a_{m-2} \\ \dots & \dots & \dots & \dots \\ a_1 & a_2 & \dots & a_0 \end{pmatrix}.$$

  Thus, a circulant matrix can be characterized by any of its rows. Without loss of generality, it is usual to denote $C = \text{circ}(a_0, a_1, \dots, a_{m-1})$ by its first, topmost row. If $C = [c_{i,j}]$, with $0 \leq i, j \leq m-1$, then $\text{circ}(a_0, a_1, \dots, a_{m-1}) = C$ means that $c_{i,j} = a_{(j-i) \bmod m}$.
  M_1 is an example of circulant MDS matrix used in AES/Rijndael [33], GRAND CRU [16] and Rainbow [52]. M_1 has entries in the finite field $\text{GF}(2^8) = \text{GF}(2)[x]/(x^8 + x^4 + x^3 + x + 1)$:

$$M_1 = \begin{pmatrix} \texttt{02}_\texttt{x} & \texttt{03}_\texttt{x} & \texttt{01}_\texttt{x} & \texttt{01}_\texttt{x} \\ \texttt{01}_\texttt{x} & \texttt{02}_\texttt{x} & \texttt{03}_\texttt{x} & \texttt{01}_\texttt{x} \\ \texttt{01}_\texttt{x} & \texttt{01}_\texttt{x} & \texttt{02}_\texttt{x} & \texttt{03}_\texttt{x} \\ \texttt{03}_\texttt{x} & \texttt{01}_\texttt{x} & \texttt{01}_\texttt{x} & \texttt{02}_\texttt{x} \end{pmatrix}$$

  and can be simply denoted $\text{circ}(\texttt{02}_\texttt{x}, \texttt{03}_\texttt{x}, \texttt{01}_\texttt{x}, \texttt{01}_\texttt{x})$. In this case, the entries of the MDS matrix were carefully chosen to be small in order to

minimize the implementation cost for encryption (number of finite field multiplications).

In block ciphers with 2-dimensional states such as the AES, the decision to multiply a row vector (of the state) on the left of an MDS matrix, or a column vector (of the state) on the right of the MDS matrix depends on the intended diffusion effect. Multiplying on the left of the MDS matrix means diffusion across the elements in each row of the state, while multiplying on the right means diffusion across the elements in each column of the state. In AES, each column of the state is multiplied on the right of a 4×4 MDS matrix:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = M_1 \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

M_1 is not involutory, and its inverse matrix has entries with higher Hamming Weight than those in M_1:

$$M_1^{-1} = \begin{pmatrix} \mathtt{0E_x} & \mathtt{09_x} & \mathtt{0D_x} & \mathtt{0B_x} \\ \mathtt{0B_x} & \mathtt{0E_x} & \mathtt{09_x} & \mathtt{0D_x} \\ \mathtt{0D_x} & \mathtt{0B_x} & \mathtt{0E_x} & \mathtt{09_x} \\ \mathtt{09_x} & \mathtt{0D_x} & \mathtt{0B_x} & \mathtt{0E_x} \end{pmatrix}.$$

M_1^{-1} can be denoted simply as $\mathrm{circ}(\mathtt{0E_x}, \mathtt{09_x}, \mathtt{0D_x}, \mathtt{0B_x})$.

The BKSQ block cipher [27] uses a non-involutory, but circulant 3×3 MDS matrix with entries in $\mathrm{GF}(2^8) = \mathrm{GF}(2)[x]/(x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1)$:

$$M_2 = \begin{pmatrix} \mathtt{03_x} & \mathtt{02_x} & \mathtt{02_x} \\ \mathtt{02_x} & \mathtt{03_x} & \mathtt{02_x} \\ \mathtt{02_x} & \mathtt{02_x} & \mathtt{03_x} \end{pmatrix}.$$

M_2 is a rare example of an MDS matrix whose order is not a power of 2. It can be denoted $M_2 = \mathrm{circ}(\mathtt{03_x}, \mathtt{02_x}, \mathtt{02_x})$.

The Hierocrypt-3 block cipher [88] uses a 4×4 circulant, non-involutory MDS matrix, over $\mathrm{GF}(2^8) = \mathrm{GF}(2)[x]/(x^8 + x^6 + x^5 + x + 1)$:

$$M_3 = \begin{pmatrix} \mathtt{C4_x} & \mathtt{65_x} & \mathtt{C8_x} & \mathtt{8B_x} \\ \mathtt{8B_x} & \mathtt{C4_x} & \mathtt{65_x} & \mathtt{C8_x} \\ \mathtt{C8_x} & \mathtt{8B_x} & \mathtt{C4_x} & \mathtt{65_x} \\ \mathtt{65_x} & \mathtt{C8_x} & \mathtt{8B_x} & \mathtt{C4_x} \end{pmatrix},$$

whose inverse is also a circulant MDS matrix:

$$M_3^{-1} = \begin{pmatrix} 82_x & C4_x & 34_x & F6_x \\ F6_x & 82_x & C4_x & 34_x \\ 34_x & F6_x & 82_x & C4_x \\ C4_x & 34_x & F6_x & 82_x \end{pmatrix}.$$

Another example of a circulant MDS matrix is:

$$M_4 = \begin{pmatrix} 05_x & 19_x & 06_x & 1B_x \\ 1B_x & 05_x & 19_x & 06_x \\ 06_x & 1B_x & 05_x & 19_x \\ 19_x & 06_x & 1B_x & 61_x \end{pmatrix}.$$

The WIDEA-8 block cipher [39] uses an 8×8 circulant MDS matrix based on a $[16, 8, 9]$-code, whose entries are in $\mathrm{GF}(2^{16}) = \mathrm{GF}(2)[x]/(x^{16} + x^5 + x^3 + x^2 + 1)$ since the word size of WIDEA-8 is 16 bits:

$$M_5 = \begin{pmatrix} 01_x & 01_x & 04_x & 01_x & 08_x & 05_x & 02_x & 09_x \\ 09_x & 01_x & 01_x & 04_x & 01_x & 08_x & 05_x & 02_x \\ 02_x & 09_x & 01_x & 01_x & 04_x & 01_x & 08_x & 05_x \\ 05_x & 02_x & 09_x & 01_x & 01_x & 04_x & 01_x & 08_x \\ 08_x & 05_x & 02_x & 09_x & 01_x & 01_x & 04_x & 01_x \\ 01_x & 08_x & 05_x & 02_x & 09_x & 01_x & 01_x & 04_x \\ 04_x & 01_x & 08_x & 05_x & 02_x & 09_x & 01_x & 01_x \\ 01_x & 04_x & 01_x & 08_x & 05_x & 02_x & 09_x & 01_x \end{pmatrix}.$$

It can be denoted $M_5 = \mathrm{circ}(01_x, 01_x, 04_x, 01_x, 08_x, 05_x, 02_x, 09_x)$. Curiously, M_5 is also used in the Whirlpool hash function [7], but under the finite field $\mathrm{GF}(2^8)$=$\mathrm{GF}(2)[x]/(x^8 + x^4 + x^3 + x^2 + 1)$. So, the same matrix remains MDS under different finite fields.

- **Algebraic construction**: an example of an MDS matrix constructed algebraically is the one used in the Curupira block cipher [8]. It is an involutory 3×3 MDS matrix with entries in $\mathrm{GF}(2^8) = \mathrm{GF}(2)[x]/(x^8 + x^6 + x^3 + x^2 + 1)$:

$$M_6 = \begin{pmatrix} 03_x & 02_x & 02_x \\ 04_x & 05_x & 04_x \\ 06_x & 06_x & 07_x \end{pmatrix}.$$

M_6 is another rare example of an MDS matrix whose order is not a power of 2.

Due to its small size, the construction followed the definition of MDS matrices (B.5): every square submatrix of an MDS matrix should be nonsingular. By defining a generic 3×3 matrix

$$X = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

one can set conditions on the nine coefficients $\{a, b, c, \ldots, i\}$ such that all square submatrices of X are nonsingular. Moreover, to make X involutory, an additional condition is imposed: $X^2 = I$ or $X = X^{-1}$. On top of those conditions, one can have some degree of freedom in choosing the coefficients to be small in order to minimize the implementation costs. This approach is feasible only for small square matrices. It requires computing the determinant of all square submatrices of the 3×3 matrix. The number of 1×1 square submatrices of X is $\binom{3}{1}^2 = 9$, which means all individual coefficients should be nonzero; the number of 2×2 square submatrices is $\binom{3}{2}^2 = 9$. In general, for an $n \times n$ matrix, there are $\binom{n}{i}^2$ possible $i \times i$ square submatrices. So, the total number of determinants computed for all submatrices of order 1 up to $n-1$ is $S = \sum_{i=1}^{n-1} \binom{n}{i}^2$. That means a system of S equations in n^2 variables. For the particular case of X, that means 18 conditions on 9 variables.

- **Cauchy matrices** [97]: given $x_0, \ldots, x_{n-1}$ and $y_0, \ldots, y_{n-1}$, the matrix $A = [a_{i,j}]$, for $0 \leq i, j \leq n-1$, where $a_{i,j} = \frac{1}{x_i + y_j}$ is called a Cauchy matrix. It is known that

$$\det(A) = \frac{\prod_{0 \leq i < j \leq n-1} (x_j - x_i)(y_j - y_i)}{\prod_{0 \leq i,j \leq n-1} (x_i + y_j)}.$$

Thus, provided the x_i are distinct, the y_j are distinct and $x_i + y_j \neq 0$ for all i, j, it follows that any square submatrix of a Cauchy matrix is nonsingular over any field.

An example provided in [97] is the involutory, symmetric 8×8 MDS matrix M_7 over the finite field $\mathrm{GF}(2^8) = \mathrm{GF}(2)/(x^8 + x^4 + x^3 + x^2 + 1)$:

$$M_7 = \begin{pmatrix} 93_x & 13_x & 57_x & DA_x & 58_x & 47_x & 0C_x & 1F_x \\ 13_x & 93_x & DA_x & 57_x & 47_x & 58_x & 1F_x & 0C_x \\ 57_x & DA_x & 93_x & 13_x & 0C_x & 1F_x & 58_x & 47_x \\ DA_x & 57_x & 13_x & 93_x & 1F_x & 0C_x & 47_x & 58_x \\ 58_x & 47_x & 0C_x & 1F_x & 93_x & 13_x & 57_x & DA_x \\ 47_x & 58_x & 1F_x & 0C_x & 13_x & 93_x & DA_x & 57_x \\ 0C_x & 1F_x & 58_x & 47_x & 57_x & DA_x & 93_x & 13_x \\ 1F_x & 0C_x & 47_x & 58_x & DA_x & 57_x & 13_x & 93_x \end{pmatrix}.$$

In M_7, the coefficients $a_{i,j}$ are quite large which imply a high implementation cost.

In [36], Gupta and Ray provided more efficient constructions of Cauchy matrices that are MDS and whose coefficients have low Hamming Weight.

- **Vandermonde matrices** [48, 77]: a Vandermonde matrix V of order d is defined by successive powers of its entries:

$$V = \begin{pmatrix} 1 & m_0 & m_0^2 & \dots & m_0^{d-1} \\ 1 & m_1 & m_1^2 & \dots & m_1^{d-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & m_i & m_i^2 & \dots & m_i^{d-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & m_{d-1} & m_{d-1}^2 & \dots & m_{d-1}^{d-1} \end{pmatrix}$$

where multiplication is defined in some finite field. Thus, the Vandermonde matrix V can be denoted by d entries: $V = \text{van}(m_0, m_1, \dots, m_{d-1})$. The Anubis block cipher [5] uses a non-involutory 4×4 MDS matrix derived from a Vandermonde matrix, in its key schedule algorithm:

$$M_8 = \begin{pmatrix} \texttt{01}_\texttt{x} & \texttt{01}_\texttt{x} & \texttt{01}_\texttt{x}^2 & \texttt{01}_\texttt{x}^3 \\ \texttt{01}_\texttt{x} & \texttt{02}_\texttt{x} & \texttt{02}_\texttt{x}^2 & \texttt{02}_\texttt{x}^3 \\ \texttt{01}_\texttt{x} & \texttt{06}_\texttt{x} & \texttt{06}_\texttt{x}^2 & \texttt{06}_\texttt{x}^3 \\ \texttt{01}_\texttt{x} & \texttt{08}_\texttt{x} & \texttt{08}_\texttt{x}^2 & \texttt{08}_\texttt{x}^3 \end{pmatrix} = \begin{pmatrix} \texttt{01}_\texttt{x} & \texttt{01}_\texttt{x} & \texttt{01}_\texttt{x} & \texttt{01}_\texttt{x} \\ \texttt{01}_\texttt{x} & \texttt{02}_\texttt{x} & \texttt{04}_\texttt{x} & \texttt{08}_\texttt{x} \\ \texttt{01}_\texttt{x} & \texttt{06}_\texttt{x} & \texttt{14}_\texttt{x} & \texttt{78}_\texttt{x} \\ \texttt{01}_\texttt{x} & \texttt{08}_\texttt{x} & \texttt{40}_\texttt{x} & \texttt{5D}_\texttt{x} \end{pmatrix}.$$

- **Hadamard matrices** [36, 77]: in [77], Sajadieh *et al.* constructed MDS matrices called Finite Field Hadamard (FFHadamard), which is defined as follows: a $2^m \times 2^m$ matrix H is a Finite Field Hadamard matrix in $\text{GF}(2^n)$ if it can be represented as

$$H = \begin{pmatrix} U & V \\ V & U \end{pmatrix},$$

where the submatrices $U_{2^{m/2} \times 2^{m/2}}$ and $V_{2^{m/2} \times 2^{m/2}}$ are also FFHadamard. By definition, all FFHadamard matrices are symmetric.
From the symmetry in the FFHadamard matrix construction H, it is possible to characterize an FFHadamard matrix from any of its rows.
Without loss of generality, usually the first, topmost row is used [77]: let $H = [h_{i,j}]$ be a $2^m \times 2^m$ matrix whose topmost row is $(x_0, x_1, \dots, x_{2^m-1})$ and $h_{i,j} = x_{i \oplus j}$, for $0 \leq i, j < 2^m$. Then, H is FFHadamard, and can be uniquely characterized by 2^m elements and is denoted as $H = \text{had}(x_0, x_1, \dots, x_{2^m-1})$.
In [36], detailed algorithms and constructions for FFHadamard matrices of dimensions which are powers of 2 were presented, along with concrete examples.
An example of a 16×16 involutory FFHadamard MDS matrix is

$$M_9 = \begin{pmatrix} 01_x & 03_x & 08_x & B2_x & 0D_x & 60_x & E8_x & 1C_x & 0F_x & 2C_x & A2_x & 8B_x & C9_x & 7A_x & AC_x & 35_x \\ 03_x & 01_x & B2_x & 08_x & 60_x & 0D_x & 1C_x & E8_x & 2C_x & 0F_x & 8B_x & A2_x & 7A_x & C9_x & 35_x & AC_x \\ 08_x & B2_x & 01_x & 03_x & E8_x & 1C_x & 0D_x & 60_x & A2_x & 8B_x & 0F_x & 2C_x & AC_x & 35_x & C9_x & 7A_x \\ B2_x & 08_x & 03_x & 01_x & 1C_x & E8_x & 60_x & 0D_x & 8B_x & A2_x & 2C_x & 0F_x & 35_x & AC_x & 7A_x & C9_x \\ 0D_x & 60_x & E8_x & 1C_x & 01_x & 03_x & 08_x & B2_x & C9_x & 7A_x & AC_x & 35_x & 0F_x & 2C_x & A2_x & 8B_x \\ 60_x & 0D_x & 1C_x & E8_x & 03_x & 01_x & B2_x & 08_x & 7A_x & C9_x & 35_x & AC_x & 2C_x & 0F_x & 8B_x & A2_x \\ E8_x & 1C_x & 0D_x & 60_x & 08_x & B2_x & 01_x & 03_x & AC_x & 35_x & C9_x & 7A_x & A2_x & 8B_x & 0F_x & 2C_x \\ 1C_x & E8_x & 60_x & 0D_x & B2_x & 08_x & 03_x & 01_x & 35_x & AC_x & 7A_x & C9_x & 8B_x & A2_x & 2C_x & 0F_x \\ 0F_x & 2C_x & A2_x & 8B_x & C9_x & 7A_x & AC_x & 35_x & 01_x & 03_x & 08_x & B2_x & 0D_x & 60_x & E8_x & 1C_x \\ 2C_x & 0F_x & 8B_x & A2_x & 7A_x & C9_x & 35_x & AC_x & 03_x & 01_x & B2_x & 08_x & 60_x & 0D_x & 1C_x & E8_x \\ A2_x & 8B_x & 0F_x & 2C_x & AC_x & 35_x & C9_x & 7A_x & 08_x & B2_x & 01_x & 03_x & E8_x & 1C_x & 0D_x & 60_x \\ 8B_x & A2_x & 2C_x & 0F_x & 35_x & AC_x & 7A_x & C9_x & B2_x & 08_x & 03_x & 01_x & 1C_x & E8_x & 60_x & 0D_x \\ C9_x & 7A_x & AC_x & 35_x & 0F_x & 2C_x & A2_x & 8B_x & 0D_x & 60_x & E8_x & 1C_x & 01_x & 03_x & 08_x & B2_x \\ 7A_x & C9_x & 35_x & AC_x & 2C_x & 0F_x & 8B_x & A2_x & 60_x & 0D_x & 1C_x & E8_x & 03_x & 01_x & B2_x & 08_x \\ AC_x & 35_x & C9_x & 7A_x & A2_x & 8B_x & 0F_x & 2C_x & E8_x & 1C_x & 0D_x & 60_x & 08_x & B2_x & 01_x & 03_x \\ 35_x & AC_x & 7A_x & C9_x & 8B_x & A2_x & 2C_x & 0F_x & 1C_x & E8_x & 60_x & 0D_x & B2_x & 08_x & 03_x & 01_x \end{pmatrix}.$$

Notice the symmetric pattern of the submatrices in M_9 like in H. Note also that M_9 can be denoted simply had(01_x, 03_x, 08_x, $B2_x$, $0D_x$, 60_x, $E8_x$, $1C_x$, $0F_x$, $2C_x$, $A2_x$, $8B_x$, $C9_x$, $7A_x$, AC_x, 35_x), from its topmost row.

The CLEFIA block cipher [86] uses two 4×4 involutory, FFHadamard MDS matrices with entries in $\mathrm{GF}(2^8) = \mathrm{GF}(2)[x]/(x^8+x^4+x^3+x^2+1)$. The matrix M_{10} is also used in the Anubis block cipher [5]:

$$M_{10} = \begin{pmatrix} 01_x & 02_x & 04_x & 06_x \\ 02_x & 01_x & 06_x & 04_x \\ 04_x & 06_x & 01_x & 02_x \\ 06_x & 04_x & 02_x & 01_x \end{pmatrix}$$

and

$$M_{11} = \begin{pmatrix} 01_x & 08_x & 02_x & 0A_x \\ 08_x & 01_x & 0A_x & 02_x \\ 02_x & 0A_x & 01_x & 08_x \\ 0A_x & 02_x & 08_x & 01_x \end{pmatrix}.$$

Thus, M_{10} =had($01_x, 02_x, 04_x, 06_x$), M_{11} =had($01_x, 08_x, 02_x, 0A_x$).

The Khazad block cipher [6] uses an 8×8 involutory FFHadamard MDS matrix, with entries in $\mathrm{GF}(2^8) = \mathrm{GF}(2)[x]/(x^8 + x^4 + x^3 + x^2 + 1)$:

$$M_{12} = \begin{pmatrix} 01_x & 03_x & 04_x & 05_x & 06_x & 08_x & 0B_x & 07_x \\ 03_x & 01_x & 05_x & 04_x & 08_x & 06_x & 07_x & 0B_x \\ 04_x & 05_x & 01_x & 03_x & 0B_x & 07_x & 06_x & 08_x \\ 05_x & 04_x & 03_x & 01_x & 07_x & 0B_x & 08_x & 06_x \\ 06_x & 08_x & 0B_x & 07_x & 01_x & 03_x & 04_x & 05_x \\ 08_x & 06_x & 07_x & 0B_x & 03_x & 01_x & 05_x & 04_x \\ 0B_x & 07_x & 06_x & 08_x & 04_x & 05_x & 01_x & 03_x \\ 07_x & 0B_x & 08_x & 06_x & 05_x & 04_x & 03_x & 01_x \end{pmatrix}.$$

Thus, M_{12} = had($01_x, 03_x, 04_x, 05_x, 06_x, 08_x, 0B_x, 07_x$).

The Grostl hash function [34] uses the 8×8 FFHadamard matrix M_{13} = had(02_x, 02_x, 03_x, 04_x, 05_x, 03_x, 05_x, 07_x).

The Whirlwind hash function [4] uses two FFHadamard matrices:

$$M_{14} = \text{had}(5_x, 4_x, A_x, 6_x, 2_x, D_x, 8_x, 3_x)$$

and

$$M_{15} = \text{had}(5_x, E_x, 4_x, 7_x, 1_x, 3_x, F_x, 8_x),$$

both with entries in $\text{GF}(2^4) = \text{GF}(2)[x]/(x^4 + x + 1)$.
Although the 1024-bit state of Whirlwind contains 16-bit words, each of these words is further partitioned into four nibbles, and individual nibbles are diffused using the two FFHadamard matrices.

- *ad hoc* **designs**: in [41], Junod and Vaudenay described a procedure to construct MDS matrices aimed at maximizing the number of entries equal to 1 in such matrices, and minimizing the number of larger entries. The objective is to minimize the implementation cost of the resulting MDS matrices in $\text{GF}(2^m)$, since multiplying by 1 costs nothing. One drawback is that there is no guarantee that such matrices will be involutory. Thus, there is no guarantee about the size of the entries in the inverse matrix. But, the authors intended to use such MDS matrices in ciphers whose design did not require the inverse matrix (for decryption).
  An example of such a construction is the 4×4 matrix denoted M_{16} with nine entries equal to 1 and only two other nonzero entries:

$$M_{16} = \begin{pmatrix} a & 1 & 1 & 1 \\ 1 & 1 & b & a \\ 1 & a & 1 & b \\ 1 & b & a & 1 \end{pmatrix},$$

  where a and b are nonzero constants in $\text{GF}(q)$, q a prime or a prime power, $a \neq 1$, $a \neq 0$. For $a = 2$ and $b = 3$, the matrix M_{16} has nine 1s compared with eight 1s in the AES MDS matrix M_1.
  The FOX64 block cipher [40] uses a 4×4 MDS matrix denoted M_{17} following this approach:

$$M_{17} = \begin{pmatrix} 1 & 1 & 1 & a \\ 1 & z & a & 1 \\ z & a & 1 & 1 \\ a & 1 & z & 1 \end{pmatrix},$$

  where $z = a^{-1} + 1 = a^7 + a^6 + a^5 + a^4 + a^3 + a^2 + 1$, with a a root of the irreducible polynomial $a^8 + a^7 + a^6 + a^5 + a^4 + a^3 + 1$. In fact, $a = 02_x$ and $z = FD_x$.
  The FOX128 block cipher [40], uses the 8×8 MDS matrix:

$$M_{18} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & a \\ 1 & a & b & c & d & e & f & 1 \\ a & b & c & d & e & f & 1 & 1 \\ b & c & d & e & f & 1 & a & 1 \\ c & d & e & f & 1 & a & b & 1 \\ d & e & f & 1 & a & b & c & 1 \\ e & f & 1 & a & b & c & d & 1 \\ f & 1 & a & b & c & d & e & 1 \end{pmatrix},$$

where $a = x+1$, $b = x^7+x$, $c = x$, $d = x^2$, $e = x^7+x^6+x^5+x^4+x^3+x^2$, $f = x^6+x^5+x^4+x^3+x^2+x$, and x is a root of the irreducible polynomial $x^8+x^7+x^6+x^5+x^4+x^3+1$. In fact, $a = 03_x$, $b = 82_x$, $c = 02_x$, $d = 04_x$, $e = fc_x$ and $f = 79_x$.

M_{19} is an example of a 3×3 involutory MDS matrix with unknown construction method:

$$M_{19} = \begin{pmatrix} \mathtt{02_x} & \mathtt{07_x} & \mathtt{04_x} \\ \mathtt{03_x} & \mathtt{06_x} & \mathtt{04_x} \\ \mathtt{03_x} & \mathtt{07_x} & \mathtt{05_x} \end{pmatrix}.$$

The Hierocrypt-L1 block cipher [89] uses a 4×4 MDS matrix, denoted M_{20}, with entries from $\mathrm{GF}(2^8) = \mathrm{GF}(2)[x]/(x^8 + x^6 + x^5 + x + 1)$:

$$M_{20} = \begin{pmatrix} \mathtt{6C_x} & \mathtt{25_x} & \mathtt{9B_x} & \mathtt{03_x} \\ \mathtt{6D_x} & \mathtt{06_x} & \mathtt{C8_x} & \mathtt{18_x} \\ \mathtt{75_x} & \mathtt{78_x} & \mathtt{9E_x} & \mathtt{1F_x} \\ \mathtt{42_x} & \mathtt{78_x} & \mathtt{EB_x} & \mathtt{61_x} \end{pmatrix}.$$

The Twofish block cipher [79] uses a non-involutory, non-symmetric 4×4 MDS matrix, denoted M_{21}, with entries from $\mathrm{GF}(2^8) = \mathrm{GF}(2)[x]/(x^8 + x^6 + x^5 + x^3 + 1)$:

$$M_{21} = \begin{pmatrix} \mathtt{01_x} & \mathtt{EF_x} & \mathtt{5B_x} & \mathtt{5B_x} \\ \mathtt{5B_x} & \mathtt{EF_x} & \mathtt{EF_x} & \mathtt{01_x} \\ \mathtt{EF_x} & \mathtt{5B_x} & \mathtt{01_x} & \mathtt{EF_x} \\ \mathtt{EF_x} & \mathtt{01_x} & \mathtt{EF_x} & \mathtt{5B_x} \end{pmatrix}.$$

The SHARK block cipher [74] uses a non-circulant, non-involutory, non-symmetric, 8×8 MDS matrix, denoted M_{22}, with entries from $\mathrm{GF}(2^8) = \mathrm{GF}(2)[x]/(x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1)$:

$$M_{22} = \begin{pmatrix} \mathtt{CE_x} & \mathtt{95_x} & \mathtt{57_x} & \mathtt{82_x} & \mathtt{8A_x} & \mathtt{19_x} & \mathtt{B0_x} & \mathtt{01_x} \\ \mathtt{E7_x} & \mathtt{FE_x} & \mathtt{05_x} & \mathtt{D2_x} & \mathtt{52_x} & \mathtt{C1_x} & \mathtt{88_x} & \mathtt{F1_x} \\ \mathtt{B9_x} & \mathtt{DA_x} & \mathtt{4D_x} & \mathtt{D1_x} & \mathtt{9E_x} & \mathtt{17_x} & \mathtt{83_x} & \mathtt{86_x} \\ \mathtt{D0_x} & \mathtt{9D_x} & \mathtt{26_x} & \mathtt{2C_x} & \mathtt{5D_x} & \mathtt{9F_x} & \mathtt{6D_x} & \mathtt{75_x} \\ \mathtt{52_x} & \mathtt{A9_x} & \mathtt{07_x} & \mathtt{6C_x} & \mathtt{B9_x} & \mathtt{8F_x} & \mathtt{70_x} & \mathtt{17_x} \\ \mathtt{87_x} & \mathtt{28_x} & \mathtt{3A_x} & \mathtt{5A_x} & \mathtt{F4_x} & \mathtt{33_x} & \mathtt{0B_x} & \mathtt{6C_x} \\ \mathtt{74_x} & \mathtt{51_x} & \mathtt{15_x} & \mathtt{CF_x} & \mathtt{09_x} & \mathtt{A4_x} & \mathtt{62_x} & \mathtt{09_x} \\ \mathtt{0B_x} & \mathtt{31_x} & \mathtt{7F_x} & \mathtt{86_x} & \mathtt{BE_x} & \mathtt{05_x} & \mathtt{83_x} & \mathtt{34_x} \end{pmatrix},$$

and its inverse is

$$M_{22}^{-1} = \begin{pmatrix} \mathtt{E7_x} & \mathtt{30_x} & \mathtt{90_x} & \mathtt{85_x} & \mathtt{D0_x} & \mathtt{4B_x} & \mathtt{91_x} & \mathtt{41_x} \\ \mathtt{53_x} & \mathtt{95_x} & \mathtt{9B_x} & \mathtt{A5_x} & \mathtt{96_x} & \mathtt{BC_x} & \mathtt{A1_x} & \mathtt{68_x} \\ \mathtt{02_x} & \mathtt{45_x} & \mathtt{F7_x} & \mathtt{65_x} & \mathtt{5C_x} & \mathtt{1F_x} & \mathtt{B6_x} & \mathtt{52_x} \\ \mathtt{A2_x} & \mathtt{CA_x} & \mathtt{22_x} & \mathtt{94_x} & \mathtt{44_x} & \mathtt{63_x} & \mathtt{2A_x} & \mathtt{A2_x} \\ \mathtt{FC_x} & \mathtt{67_x} & \mathtt{8E_x} & \mathtt{10_x} & \mathtt{29_x} & \mathtt{75_x} & \mathtt{85_x} & \mathtt{71_x} \\ \mathtt{24_x} & \mathtt{45_x} & \mathtt{A2_x} & \mathtt{CF_x} & \mathtt{2F_x} & \mathtt{22_x} & \mathtt{C1_x} & \mathtt{0E_x} \\ \mathtt{A1_x} & \mathtt{F1_x} & \mathtt{71_x} & \mathtt{40_x} & \mathtt{91_x} & \mathtt{27_x} & \mathtt{18_x} & \mathtt{A5_x} \\ \mathtt{56_x} & \mathtt{F4_x} & \mathtt{AF_x} & \mathtt{32_x} & \mathtt{D2_x} & \mathtt{A4_x} & \mathtt{DC_x} & \mathtt{71_x} \end{pmatrix}.$$

Note that the entries of both M_{22} and M_{22}^{-1} have quite heavy Hamming Weight.

Designing MDS matrices satisfying too many criteria at once may sometimes involve conflicts. Some attempts at designing such MDS matrices have failed, which means that the security claims coming from the assumption that the matrix is MDS cannot be upheld [36, 66, 84].

B.2 Implementation Costs

As discussed previously, involutory MDS matrices are useful when the inverse matrix is needed, for instance, in the decryption procedure of some blocks cipher e.g. those following the SPN structure.

Another practical criterion for efficient implementation is the effective size of each entry of the MDS matrix. One approach used in the FOX family of block ciphers [41] was to maximize the number of entries equal to 1, and minimize the number of other (larger) coefficients.

The cost of a matrix multiplication can be measured by the total number of two primitive operations: bitwise exclusive-or ($\oplus$) and xtime (which is multiplication by 2 in the underlying finite field). Note that every element in the finite field GF(2^m) can be decomposed in a binary form, for instance, $\mathtt{0E_x} = 00001110_2 = 8 \oplus 4 \oplus 2$. Thus, if $\mathtt{0E_x}$ is an entry of an MDS matrix, a multiplication by an element s means $\mathtt{0E_x} \cdot s = (8 \cdot s) \oplus (4 \cdot s) \oplus (2 \cdot s)$ where $\cdot$ stands for multiplication in GF(2^m). Consequently, it can be summarized by a series of multiplications of s by powers of 2, or repeated multiplication by 2.

Let $\mathrm{xtime}(s) = 2 \cdot s$. Then,

$$\mathtt{0E_x} \cdot s = \mathrm{xtime}(\mathrm{xtime}(\mathrm{xtime}(s))) \oplus \mathrm{xtime}(\mathrm{xtime}(s)) \oplus \mathrm{xtime}(s),$$

which means six xtimes calls and two xors. For higher efficiency, we can chain the calls since the same s is present in all cases. Thus,

$$\mathtt{0E_x} \cdot s = \mathrm{xtime}(\mathrm{xtime}(\mathrm{xtime}(s \oplus 1) \oplus 1),$$

and this multiplication between two m-bit values in GF(2^m) cost only two xors and three calls to xtimes.

Since the coefficients of the MDS matrix are fixed, it becomes clear that the smaller those coefficients are the lower the total cost of a single matrix multiplication. The overall size of the entries is more important than its Hamming Weight. Suppose $\texttt{11}_\texttt{x}$, which has a lower Hamming Weight than $\texttt{0E}_\texttt{x}$. The multiplication $\texttt{11}_\texttt{x} \cdot s = (16 \cdot s) \oplus s = \text{xtime}(\text{xtime}(\text{xtime}(\text{xtime}((s))))) \oplus s$ costs four xtimes and one xor.

The implementation of xtimes can be performed by an explicit algorithm or by a table look-up (which costs memory for storage). In either case, a single xtime call costs more than a single xor, and the multiplication by $\texttt{11}_\texttt{x}$ is more expensive than by $\texttt{0E}_\texttt{x}$.

In Table B.1, we summarize and compare properties and costs of some of the MDS matrices described previously.

Table B.1 Summary of properties of MDS matrices. The polynomial $p(x)$ or finite field is listed in the 5th column.

Matrix	Order	Type	Invol.	Finite Field GF(2)[x]/($p(x)$)	#xors	# xtime	Ref.
M_1	4×4	circulant	No	$x^8 + x^4 + x^3 + x + 1$	4	8	[33]
M_1^{-1}	4×4	circulant	No	$x^8 + x^4 + x^3 + x + 1$	28	48	[33]
M_2	3×3	circulant	No	$x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1$	3	9	[27]
M_5	8×8	circulant	No	$x^{16} + x^5 + x^3 + x^2 + 1$	16	88	[39]
M_6	3×3	algebraic	Yes	$x^8 + x^6 + x^3 + x^2 + 1$	6	15	[8]
M_7	8×8	Cauchy	Yes	$x^8 + x^4 + x^3 + x^2 + 1$	184	344	[97]
M_8	4×4	Vandermonde	No	$x^8 + x^4 + x^3 + x^2 + 1$	9	33	[5]
M_9	16×16	FFHadamard	Yes	GF(2^8)	544	1248	[36]
M_{10}	4×4	FFHadamard	Yes	$x^8 + x^4 + x^3 + x^2 + 1$	4	20	[86]
M_{14}	8×8	FFHadamard	Yes	$x^4 + x + 1$	24	68	[4]
M_{17}	4×4	*ad hoc*	No	$x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1$	24	32	[40]
M_{18}	8×8	*ad hoc*	No	$x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1$	78	169	[40]

Notice the implementation cost of matrices of the same order varies because of the different entries. For instance, M_6 costs more xtime operations than M_2, but M_6 is involutory, which may be advantageous if the inverse matrix is needed, since M_2 is not involutory.

The same reasoning applies to M_1 and M_8, as well as between M_{17} and M_{10}. Although M_1 implementation costs are the lowest, its inverse costs more than M_{10} which is involutory.

The same holds for M_5, M_7, M_{14} and M_{18}. Among the involutory matrices, M_{14} has the lowest implementation cost.

Appendix C
Substitution Boxes (S-boxes)

Anything is possible if approached with conviction.
– Konosuke Matsushita

This is a brief survey of cryptographic Substitution boxes, or simply S-boxes, and a number of relevant properties that are expected from S-boxes used in a cryptographic setting, such as the differential and linear profiles, the algebraic normal form and the nonlinearity.

S-boxes are the main nonlinear components in several cryptographic primitives, such as block and stream ciphers, hash functions and Message Authentication Codes (MAC) [62]. S-boxes fulfill the role of confusion as stated by Shannon in [81] and additionally provide (local) diffusion in the sense that all of its output bits depend (nonlinearly) on all input bits.

Previous research works on S-boxes include [13, 72, 73, 76, 95].

C.1 S-box Definition

An $n \times m$-bit Substitution Box (S-box) is a nonlinear mapping formally denoted $S : GF(2)^n \rightarrow GF(2)^m$ that is typically used to provide the confusion property [81] in several cryptographic primitives such as block ciphers, stream ciphers and hash functions.

In Substitution-Permutation Network (SPN) ciphers, the S-boxes are bijective mappings due to the need for decryption [33]. In Feistel Network ciphers, the S-boxes do not need to be bijective [68], but this can lead to a number of attacks that could be averted with the use of bijective S-boxes [28, 75, 65]. Some attacks though, depend exactly on the fact that S-boxes are bijective mappings, such as the reflection attack [37] against the GOST cipher [98] and the SASAS attack [14].

The majority of known S-boxes published in the literature are fixed mappings. Some S-boxes, though, are key dependent such as in Blowfish [78], or secret and variable depending on the application environment, such as the 4×4-bit S-boxes of GOST cipher [98]. The latest trend of lightweight cipher designs use small, fixed, 4×4-bit S-boxes [96, 26, 15, 83, 3].

If the S-boxes have small dimensions then they are often implemented as a static truth table (or look-up tables).

J. Nakahara Jr., *Lai-Massey Cipher Designs*, https://doi.org/10.1007/978-3-319-68273-0

Some S-boxes are involutory mappings, that is, they are their own inverses. This concept was adopted in ciphers such as Khazad and Anubis (also in the tweaked versions), to save memory for decryption. But, most block ciphers nowadays either do not adopt this strategy or do not need them, such as ciphers using the Feistel [68] or the Lai-Massey designs [40] which do not need the inverse S-box. The AES S-box without the affine transformation would be involutory, that is, if $S[x] = 1/x$, then $S[S[x]] = 1/(1/x) = x$.

In some cases, like in the DES cipher [68], there are eight contiguous S-boxes, and not only the individual S-box contents but also their relative position matters for the security of the DES [58] because some input bits are shared by neighboring S-boxes.

For 4×4 bijective S-boxes, exhaustive analysis over the whole space of such S-boxes has already been performed in [76]. In [13], Bilgin *et al.* performed analysis also for all 3×3 S-boxes and for the DES 6×4 S-boxes. Note that there are only $2^3! = 40320 \approx 2^{15.2992}$ possible permutations (bijective 3×3 S-boxes), and $2^4! = 16! = 20922789888000 \approx 2^{44.2501}$ permutations (bijective 4×4 S-boxes). For DES S-boxes, considered simply as surjective mappings, there are $2^6!/(2^6 - 2^4)! = 10221346459144248675287040000 \approx 2^{93.045}$ such mappings. But, since each DES S-box consists of four 4×4-bit S-boxes, there are in fact $4!\binom{16!}{4} = \approx 2^{177.00056}$ such mappings.

Tables C.1 and C.2 provide a non-exhaustive enumeration of parameters and properties of S-boxes, listed in alphabetical order of the cryptographic algorithm they are embedded in.

An $n \times m$ S-box can be interpreted as a vectorial Boolean mapping $S : GF(2)^n \rightarrow GF(2)^m$. Therefore, it can be described in terms of its m component Boolean functions $s_i : GF(2)^n \rightarrow GF(2)$ for $0 \leq i < m$. Thus, $S[x]=(s_0(x), s_1(x), \ldots, s_{m-1}(x))$.

Concerning the differential cryptanalysis technique [12, 71], relevant properties for an S-box include the *differential profile* and the *differential uniformity*.

The differential profile is related to the distribution of the differences such as:

$$\delta_S(a, b) = \#\{x \in GF(2^n) : S[x \oplus a] \oplus S[x] = b\}.$$

The value $S[x \oplus a] \oplus S[x]$ is called the *output xor-difference* to the S-box S, where a is the input difference. The value

$$\delta_{\max} = \max_{a\neq0,b}\delta_S(a, b)$$

is the *differential uniformity* of S, and identifies the most probable nontrivial input/output difference pair(s) (a, b) that can propagate across S.

An extensive listing of $\delta_S(a, b)$ for all $0 \leq a < 2^n$, for $b = S[x\oplus a]\oplus S[x]$ and for a given difference operator, such as $\oplus$, is called the *Difference Distribution Table (DDT)* of S. The S-box is said to be $\delta_{\max}$-differentially uniform, in the sense that all nontrivial entries in its DDT are less than or equal to $\delta_{\max}$.

Table C.1 General parameters and properties of S-boxes.

Algorithm	Dimensions (bits)	#S-box	Biject?	Fixed? ⋆	Invol.?	Ref./Year
3WAY	3×3	1	Yes	Yes	No	[24]/1993
AES/Rijndael	8×8	1	Yes	Yes	No	[33]/2001
ARIA	8×8	2	Yes	Yes	No	[47]/2003
BASEKING	3×3	1	Yes	Yes	No	[23]/1995
BKSQ	8×8	1	Yes	Yes	No	[27]/2000
BLOWFISH	8×32	1	No	No	No	[78]/1994
CAMELLIA	8×8	4	Yes	Yes	No	[61]/2004
CAST-128/256	8×32	4	No	Yes	No	[2]/1997
CIPHERUNICORN-A	8×8	4	Yes	Yes	No	[91]/2000
CIPHERUNICORN-E	8×8	4	Yes	Yes	No	[90]/1998
CLEFIA	8×8	2	Yes	Yes	No	[85]/2007
COCONUT98	8×24	1	No	Yes	No	[93]/1998
CRYPTON	8×8	2	Yes	Yes	No	[54]/1998
CS	8×8	1	Yes	Yes	No	[87]/1998
CTC	3×3	1	Yes	Yes	No	[22]/2006
DES	6×4	8	No	Yes	No	[68]/1999
s^2-DES	6×4	8	No	Yes	No	[42]/1991
s^3-DES	6×4	8	No	Yes	No	[44]/1993
s^5-DES	6×4	8	No	Yes	No	[43]/1995
DRAGON	8×32	2	No	Yes	No	[29]/2006
E2	8×8	1	Yes	Yes	No	[70]/1998
FOX/IDEA-NXT	8×8	1	Yes	Yes	No	[40]/2004
HAMSI	4×4	1	Yes	Yes	No	[46]/1998
HIEROCRYPT-L1/3	8×8	1	Yes	Yes	No	[88, 89]/2000
JH	4×4	2	Yes	Yes	No	[96]/2011
KASUMI	7×7	1	Yes	Yes	No	[1]/1999
KASUMI	9×9	1	Yes	Yes	No	[1]/1999
KECCAK	5×5	1	Yes	Yes	No	[23]/1995
KHAZAD/ANUBIS	8×8	1	Yes	Yes	Yes	[6]/2000
KHUFU	8×32	‡	No	No	No	[63]/1991
LED	4×4	1	Yes	Yes	No	[35]/2011
LOKI89	12×8	1	No	Yes	No	[19]/1990
LOKI91	12×8	1	No	Yes	No	[17]/1991
LOKI97	11×8	1	No	Yes	No	[18]/1998
LOKI97	13×8	1	No	Yes	No	[18]/1998
MAGENTA	8×8	1	Yes	Yes	No	[38]/1998
MARS	9×32	1	No	Yes	No	[20]/1999
MISTY1	7×7	1	Yes	Yes	No	[60]/1997
MISTY1	9×9	1	Yes	Yes	No	[60]/1997
NOEKEON	4×4	1	Yes	Yes	Yes	[26]/2000
PANAMA	3×3	1	Yes	Yes	No	[23]/1995
PRESENT	4×4	1	Yes	Yes	No	[15]/2007

⋆: if the S-box is not static/fixed, then it is key dependent.
‡: key dependent S-boxes; the number of S-boxes depends on the number of rounds

Table C.2 General parameters and properties of S-boxes (cont.).

Algorithms	Dimensions (bits)	#S-box	Biject?	Fixed?	Involu.?	Ref./Year
PRINTcipher	3×3	1	Yes	Yes	No	[45]/2010
RADIOGATÚN	3×3	1	Yes	Yes	No	[9]/2006
RAINBOW	8×8	1	Yes	Yes	No	[52]/1998
SAFER K/SK/+/++	8×8	2	Yes	Yes	No	[56]/1994
SC2000	4×4	1	Yes	Yes	No	[83]/2002
SC2000	5×5	1	Yes	Yes	No	[83]/2002
SC2000	6×6	1	Yes	Yes	No	[83]/2002
SEED	8×8	2	Yes	Yes	No	[53]/2005
SERPENT-0	4×4	32	Yes	Yes	No	[11]/1998
SERPENT	4×4	8	Yes	Yes	No	[3]/1998
SHARK	8×8	1	Yes	Yes	No	[74]/1996
SKIPJACK	8×8	1	Yes	Yes	No	[69]/1998
SMS4	8×8	1	Yes	Yes	No	[30]/2008
SQUARE	8×8	1	Yes	Yes	No	[25]/1997
TWOFISH	8×8	1	Yes	Yes	No	[79]/1997
WHIRLPOOL	8×8	1	Yes	Yes	No	[7]/2003
WHIRLWIND	16×16	1	Yes	Yes	Yes	[4]/2010
ZODIAC	8×8	2	Yes	Yes	No	[51]/2000
ZUC	8×8	2	Yes	Yes	No	[31]/2011

Concerning the linear cryptanalysis technique [57], relevant properties for an S-box include the *linear profile* and the *linear uniformity*. The linear profile means the distribution of the linear bias values. Let $< a, x >$ denote the dot product of two bit strings $a, x \in GF(2)^t$, that is, $< a, x > = < x, a > = \oplus_{j=0}^{t-1} x_j \cdot a_j$, where $a = (a_0, a_1, \ldots, a_{t-1})$. Let $\gamma_S(a, b) = \#\{x : 0 \leq x < 2^n, < x, a > = < S[x], b >\} - 2^{n/2}$, where $a \in GF(2)^n$ and $b \in GF(2)^m$. When $\gamma_S(a, b)$ is nonzero, there is a nonzero correlation between a linear combination of a input bits and b output bits. The value $\gamma_{\max} = \max_{a \neq 0, b \neq 0} \gamma_S(a, b)$ indicates the most biased nontrivial linear relation(s) across S for all bitmasks a and b. The value $\gamma_{\max}$ is the linear uniformity (the counterpart of the differential uniformity).

An extensive listing of $\gamma_S(a, b) \cdot 2^n$ values for all possible input/output bitmask pairs (a, b) is called the *Linear Distribution Table (LAT)* of S, following [57]. The S-box S is said to be $\gamma_{\max}$-linearly uniform in the sense that all nontrivial entries in its LAT are less than or equal to $\gamma_{\max}$.

Table C.3 gives a non-exhaustive listing of some cryptographic parameters of some S-boxes. We use the value $\delta_{\max}/2^n$ since it gives the probability of the best nontrivial difference propagating across each S-box. Similarly, we use $|\gamma_{\max}/2^n|$ since it gives the largest bias among the nontrivial bitmasks propagating across the S-box, where n is the input size of S.

Table C.3 Cryptographic parameters of S-boxes.

Cipher	Dimensions ($n \times m$ bits)	$\delta_{\max}/2^n$ $\star$	$\lvert\gamma_{\max}\rvert/2^n$
3WAY/BASEKING	3×3	2^{-2}	2^{-2}
CTC	3×3	2^{-2}	2^{-2}
PANAMA/RADIOGATÚN	3×3	2^{-2}	2^{-2}
PRINTcipher	3×3	2^{-2}	2^{-2}
HAMSI	4×4	2^{-2}	2^{-2}
NOEKEON	4×4	2^{-2}	2^{-2}
PRESENT	4×4	2^{-2}	2^{-2}
JH (S0,S1) †	4×4	2^{-2}	2^{-2}
SC2000	4×4	2^{-2}	2^{-2}
SERPENT ‡	4×4	2^{-2}	2^{-2}
SC2000	5×5	2^{-4}	2^{-3}
SC2000	6×6	2^{-4}	2^{-3}
DES (S1)	6×4	2^{-2}	$2^{-1.83}$
DES (S2)	6×4	2^{-2}	2^{-2}
DES (S3)	6×4	2^{-2}	2^{-2}
DES (S4)	6×4	2^{-2}	2^{-2}
DES (S5)	6×4	2^{-2}	$2^{-1.67}$
DES (S6)	6×4	2^{-2}	$2^{-2.19}$
DES (S7)	6×4	2^{-2}	$2^{-1.83}$
DES (S8)	6×4	2^{-2}	2^{-2}
s^2-DES (S1)	6×4	$2^{-2.19}$	$2^{-2.19}$
s^2-DES (S2)	6×4	$2^{-2.19}$	$2^{-2.19}$
s^2-DES (S3)	6×4	$2^{-2.19}$	$2^{-2.19}$
s^2-DES (S4)	6×4	2^{-2}	$2^{-2.19}$
s^2-DES (S5)	6×4	2^{-2}	$2^{-1.83}$
s^2-DES (S6)	6×4	2^{-2}	$2^{-2.19}$
s^2-DES (S7)	6×4	2^{-2}	2^{-2}
s^2-DES (S8)	6×4	2^{-2}	$2^{-2.19}$
s^3-DES (S1)	6×4	$2^{-1.67}$	2^{-2}
s^3-DES (S2)	6×4	$2^{-1.83}$	2^{-2}
s^3-DES (S3)	6×4	$2^{-1.67}$	2^{-2}
s^3-DES (S4)	6×4	$2^{-1.67}$	$2^{-1.41}$
s^3-DES (S5)	6×4	$2^{-1.67}$	$2^{-1.41}$
s^3-DES (S6)	6×4	$2^{-1.67}$	$2^{-1.67}$
s^3-DES (S7)	6×4	$2^{-1.67}$	$2^{-1.67}$
s^3-DES (S8)	6×4	$2^{-1.67}$	2^{-2}
s^5-DES (S1)	6×4	$2^{-1.83}$	2^{-2}
s^5-DES (S2)	6×4	$2^{-2.67}$	2^{-2}
s^5-DES (S3)	6×4	$2^{-2.67}$	2^{-2}
s^5-DES (S4)	6×4	$2^{-2.67}$	2^{-2}
s^5-DES (S5)	6×4	$2^{-2.67}$	2^{-2}
s^5-DES (S6)	6×4	$2^{-1.83}$	2^{-2}
s^5-DES (S7)	6×4	$2^{-1.83}$	2^{-2}
s^5-DES (S8)	6×4	$2^{-1.83}$	2^{-2}

$\star$: the inverse S-box, when it exists, has the same differential and linear profiles.
†: both s-boxes have the same differential and linear profiles.
‡: all eight S-boxes (S1 to S8) have the same δ_S and $|\gamma_S|$.

Table C.4 Cryptographic parameters of S-boxes (cont.).

Cipher	Dimensions ($n \times m$ bits)	$\delta_{\max}/2^n$	$\lvert\gamma_{\max}\rvert/2^n$
KASUMI (S7)	7×7	2^{-6}	2^{-4}
MISTY1 (S7)	7×7	2^{-6}	2^{-4}
AES/Rijndael	8×8	2^{-6}	2^{-4}
ANUBIS (original)	8×8	2^{-5}	$2^{-2.91}$
ANUBIS (tweak)	8×8	2^{-5}	2^{-3}
ARIA (S1)	8×8	2^{-6}	2^{-4}
ARIA (S2)	8×8	2^{-6}	2^{-4}
BKSQ	8×8	2^{-6}	2^{-4}
CAMELLIA	8×8	2^{-6}	2^{-4}
CIPHERUNICORN-A	8×8	2^{-6}	2^{-4}
CIPHERUNICORN-E	8×8	2^{-6}	2^{-4}
CLEFIA (S0)	8×8	$2^{-4.68}$	$2^{-3.19}$
CLEFIA (S1)	8×8	2^{-6}	2^{-4}
CRYPTON	8×8	2^{-5}	2^{-3}
CS	8×8	2^{-4}	2^{-3}
E2	8×8	$2^{-4.68}$	$2^{-3.19}$
FOX/IDEANXT	8×8	2^{-4}	2^{-3}
HIEROCRYPT-L1/3	8×8	2^{-6}	2^{-4}
KHAZAD (tweak)	8×8	2^{-5}	2^{-3}
MAGENTA	8×8	2^{-5}	$2^{-3.29}$
RAINBOW	8×8	$2^{-5.41}$	2^{-3}
SAFER K/SK/+/++	8×8	2^{-1}	$2^{-2.47}$
SEED	8×8	2^{-6}	2^{-4}
SHARK	8×8	2^{-6}	2^{-4}
SKIPJACK	8×8	$2^{-4.41}$	$2^{-3.19}$
SMS4	8×8	2^{-6}	2^{-4}
SQUARE	8×8	2^{-6}	2^{-4}
TWOFISH	8×8	$2^{-4.67}$	2^{-3}
WHIRLPOOL	8×8	2^{-5}	$2^{-3.19}$
ZODIAC (S1)	8×8	$2^{-4.67}$	$2^{-2.91}$
ZODIAC (S2)	8×8	$2^{-4.41}$	$2^{-2.87}$
ZUC (S0)	8×8	2^{-5}	2^{-3}
ZUC (S1)	8×8	2^{-6}	2^{-4}
COCONUT98	8×24	2^{-6}	$2^{-2.24}$
CAST-128/256	8×32	2^{-7}	2^{-1}
DRAGON	8×32	2^{-7}	2^{-1}
KASUMI (S9)	9×9	2^{-8}	2^{-5}
MISTY1 (S9)	9×9	2^{-8}	2^{-5}
MARS	9×32	2^{-7}	$2^{-2.61}$
LOKI97	11×8	2^{-7}	2^{-6}
LOKI89	12×8	2^{-4}	$2^{-4.67}$
LOKI91	12×8	$2^{-4.95}$	$2^{-4.64}$
LOKI97	13×8	2^{-7}	2^{-7}
WHIRLWIND	16×16	2^{-14}	2^{-8}

C.2 S-box Representations

There are several possible representations for an S-box:

- **Table Look-up** or *Truth Table*: an S-box $S : GF(2)^n \to GF(2)^m$ requires $m \cdot 2^n$ bits of storage. Depending on the application environment and the dimensions n, m, this representation may provide very fast access at an affordable storage space.
- **Algebraic Normal Form (ANF)** is a method of standardizing and normalizing logical formulas. The ANF of a Boolean mapping $f : GF(2)^n \to GF(2)$ is the unique representation of f as a polynomial over the polynomial ring $F_2[x_{n-1}, \ldots, x_0]/(x_{n-1}^2 + x_{n-1}, \ldots, x_0^2 + x_0)$. Assume the following bit numbering $(x_{n-1}, \ldots, x_1, x_0)$ for the S-box input and $(y_{n-1}, \ldots, y_1, y_0)$ for the S-box output, each of which has its own ANF. In general, the ANF of f has the form $f(x_{n-1}, \ldots, x_0) = a_0 + a_1.x_0 + \ldots + a_n.x_{n-1} + a_{0,1}.x_0.x_1 + \ldots + a_{n-2,n-1}.x_{n-2}.x_{n-1} + \ldots + a_{0,1,\ldots,n-1}.x_0.x_1 \ldots x_{n-1}$, where the coefficients $a_{i_1,i_2,\ldots} \in GF(2)$, $+$ denotes bitwise exclusive-or and . denotes the bitwise-AND operation.
  As an example, consider the 5-tuple input $(x_0, x_1, x_2, x_3, x_4)$ and the 5-tuple output $(y_0, y_1, y_2, y_3, y_4)$. The ANF of the 5×5-bit S-box of KECCAK (see Table C.8) is: $y_0 = x_0 + \overline{x_1}x_2 = x_0 + x_2 + x_1x_2$, $y_1 = x_1 + x_3 + x_2x_3$, $y_2 = x_2 + x_4 + x_3x_4$, $y_3 = x_0 + x_3 + x_0x_4$ and $y_4 = x_1 + x_4 + x_0x_1$.
  There are also many other normal forms. A propositional formula in Conjunctive Normal Form (CNF) is a formula in the form

$$\wedge_{i=1}^{n}(\vee_{j=1}^{m_i} A_{ij}) = (A_{11} \vee \ldots \vee A_{1m_1}) \wedge \ldots \wedge (A_{n1} \vee \ldots \vee A_{nm_n}),$$

  where each A_{ij}, with $1 \leq i \leq n$, $1 \leq j \leq m_i$, is either a symbol x or its negation $\overline{x}$; $\vee$ denotes bitwise-OR; $\wedge$ denotes bitwise-AND. A propositional formula in Disjunctive Normal Form (DNF) is a formula in the form

$$\vee_{i=1}^{n}(\wedge_{j=1}^{m_i} A_{ij}) = (A_{11} \wedge \ldots \wedge A_{1m_1}) \vee \ldots \vee (A_{n1} \wedge \ldots \wedge A_{nm_n}),$$

  where each A_{ij}, with $1 \leq i \leq n$, $1 \leq j \leq m_i$, is either a symbol or the negation of a symbol.
  To transform a formula in ANF to CNF, one has to account for the exclusive-or: $a + b \equiv (a \wedge \overline{b}) \vee (\overline{a} \wedge b)$, which is in CNF. Analogously, $a + b \equiv (\overline{a} \vee \overline{b}) \wedge (a \vee b)$ which is in DNF.
- **Polynomial expression over a finite field**: for instance, the AES [33] S-box relation $y = S[x]$, in GF(2^8) can be represented as the polynomial (with only nine nonzero coefficients) $y = 63_x + 8f_x \cdot x^{127} + b5_x \cdot x^{191} + x^{223} + f4_x \cdot x^{239} + 25_x \cdot x^{247} + f9_x \cdot x^{251} + 09_x \cdot x^{253} + 05_x \cdot x^{254}$.
  SHARK S-box [74], which is also based on the inversion operation in GF(2^8), plus an affine transformation, has a similarly sparse polynomial

expression (with only nine nonzero coefficients): $y = b1_x + bb_x.x^{127} + 1f_x.x^{191} + 99_x.x^{223} + 98_x.x^{239} + da_x.x^{247} + 9d_x.x^{251} + a3_x.x^{253} + 3e_x.x^{254}$. The sparsity becomes evident compared to the polynomial expression for the S-box S0 of CLEFIA (with 247 nonzero coefficients):

$$y = 57_x + 43_x.x + 13_x.x^2 + 73_x.x^3 + 0b_x.x^4 + 7e_x.x^5 + bf_x.x^6 + 7b_x.x^7 + 96_x.x^8 + 4e_x.x^9 + 87_x.x^{10} + de_x.x^{11} + fd_x.x^{12} + 7b_x.x^{13} + c4_x.x^{14} + ca_x.x^{15} + f2_x.x^{16} + 27_x.x^{17} + e7_x.x^{18} + 54_x.x^{19} + 82_x.x^{20} + 86_x.x^{21} + 5e_x.x^{22} + d4_x.x^{23} + 9a_x.x^{24} + ae_x.x^{25} + f1_x.x^{26} + d6_x.x^{27} + 12_x.x^{28} + 6a_x.x^{29} + 28_x.x^{30} + 6c_x.x^{31} + fa_x.x^{32} + 05_x.x^{33} + 3a_x.x^{34} + a9_x.x^{35} + 4d_x.x^{36} + e6_x.x^{37} + 7b_x.x^{38} + f6_x.x^{39} + 2a_x.x^{40} + b0_x.x^{41} + 37_x.x^{42} + ff_x.x^{43} + 23_x.x^{44} + 65_x.x^{45} + ef_x.x^{46} + 49_x.x^{47} + 68_x.x^{48} + 8f_x.x^{49} + 45_x.x^{50} + ba_x.x^{51} + f2_x.x^{52} + 67_x.x^{53} + cb_x.x^{54} + 7c_x.x^{55} + b6_x.x^{56} + 71_x.x^{57} + 9c_x.x^{58} + c6_x.x^{59} + 3f_x.x^{60} + 3c_x.x^{61} + 72_x.x^{62} + 25_x.x^{63} + f7_x.x^{64} + 0a_x.x^{65} + 9e_x.x^{66} + 9c_x.x^{67} + 71_x.x^{68} + e8_x.x^{69} + c8_x.x^{70} + 69_x.x^{71} + ad_x.x^{72} + 0a_x.x^{73} + 5b_x.x^{74} + ad_x.x^{75} + d1_x.x^{76} + 36_x.x^{77} + 6e_x.x^{78} + f1_x.x^{79} + 4f_x.x^{80} + 14_x.x^{81} + 7f_x.x^{82} + f8_x.x^{83} + c9_x.x^{84} + a8_x.x^{85} + ef_x.x^{86} + 7c_x.x^{87} + 03_x.x^{88} + 92_x.x^{89} + e4_x.x^{90} + 5d_x.x^{91} + 69_x.x^{92} + 9c_x.x^{93} + 49_x.x^{94} + e2_x.x^{95} + af_x.x^{96} + 83_x.x^{97} + af_x.x^{98} + 48_x.x^{99} + 88_x.x^{100} + d2_x.x^{101} + 8e_x.x^{102} + 5d_x.x^{103} + f1_x.x^{104} + 70_x.x^{105} + eb_x.x^{106} + 8e_x.x^{107} + 4c_x.x^{108} + e7_x.x^{109} + 0f_x.x^{110} + 06_x.x^{111} + ec_x.x^{112} + fa_x.x^{113} + 82_x.x^{114} + 31_x.x^{115} + 98_x.x^{116} + 40_x.x^{117} + bb_x.x^{118} + d2_x.x^{119} + e9_x.x^{120} + af_x.x^{121} + eb_x.x^{122} + 36_x.x^{123} + 0f_x.x^{124} + da_x.x^{125} + 51_x.x^{126} + a5_x.x^{128} + 85_x.x^{129} + 76_x.x^{130} + f4_x.x^{131} + 52_x.x^{132} + bd_x.x^{133} + 4c_x.x^{134} + 7d_x.x^{135} + 27_x.x^{136} + 78_x.x^{137} + a2_x.x^{138} + 53_x.x^{139} + fb_x.x^{140} + 68_x.x^{141} + ef_x.x^{142} + 07_x.x^{143} + da_x.x^{144} + d1_x.x^{145} + a7_x.x^{146} + 6b_x.x^{147} + de_x.x^{148} + b9_x.x^{149} + 4e_x.x^{150} + fb_x.x^{151} + 8a_x.x^{152} + 44_x.x^{153} + 0b_x.x^{154} + cc_x.x^{155} + 9b_x.x^{156} + 5f_x.x^{157} + 8d_x.x^{158} + a2_x.x^{159} + 71_x.x^{160} + dc_x.x^{161} + ce_x.x^{162} + eb_x.x^{163} + a7_x.x^{164} + 07_x.x^{165} + 28_x.x^{166} + dd_x.x^{167} + 1e_x.x^{168} + 9e_x.x^{169} + 12_x.x^{170} + 78_x.x^{171} + 04_x.x^{172} + 8a_x.x^{173} + 83_x.x^{174} + e2_x.x^{175} + 3d_x.x^{176} + 0e_x.x^{177} + 3b_x.x^{178} + 0a_x.x^{179} + 77_x.x^{180} + e6_x.x^{181} + 15_x.x^{182} + 57_x.x^{183} + 57_x.x^{184} + 76_x.x^{185} + 50_x.x^{186} + 1a_x.x^{187} + 9b_x.x^{188} + a6_x.x^{189} + bb_x.x^{190} + 32_x.x^{192} + 15_x.x^{193} + 1f_x.x^{194} + 39_x.x^{195} + 4c_x.x^{196} + 35_x.x^{197} + 8e_x.x^{198} + 73_x.x^{199} + f4_x.x^{200} + 92_x.x^{201} + 02_x.x^{202} + bd_x.x^{203} + 2b_x.x^{204} + cb_x.x^{205} + 40_x.x^{206} + 92_x.x^{207} + da_x.x^{208} + 5b_x.x^{209} + 7a_x.x^{210} + e0_x.x^{211} + 0b_x.x^{212} + 1e_x.x^{213} + 12_x.x^{214} + e6_x.x^{215} + 59_x.x^{216} + 50_x.x^{217} + 5d_x.x^{218} + 02_x.x^{219} + 20_x.x^{220} + e5_x.x^{221} + 6b_x.x^{222} + 79_x.x^{224} + 9a_x.x^{225} + a7_x.x^{226} + 3d_x.x^{227} + 21_x.x^{228} + 5b_x.x^{229} + 13_x.x^{230} + 21_x.x^{231} + b1_x.x^{232} + 1a_x.x^{233} + 0a_x.x^{234} + 24_x.x^{235} + c9_x.x^{236} + 1c_x.x^{237} + 58_x.x^{238} + 20_x.x^{240} + 2e_x.x^{241} + 7e_x.x^{242} + 15_x.x^{243} + 60_x.x^{244} + b9_x.x^{245} + 2f_x.x^{246} + 42_x.x^{248} + 71_x.x^{249} + 64_x.x^{250} + db_x.x^{252}$$

Analogously, for the S-box S1 of CLEFIA (with 253 nonzero coefficients):

$$y = 6c_x + d3_x.x + 47_x.x^2 + c8_x.x^3 + ca_x.x^4 + 5b_x.x^5 + 0b_x.x^6 + 9f_x.x^7$$

$$+ 05_x.x^8 + 8b_x.x^9 + f4_x.x^{10} + b9_x.x^{11} + d9_x.x^{12} + eb_x.x^{13} + 0d_x.x^{14}$$
$$+ a9_x.x^{15} + a7_x.x^{16} + f5_x.x^{17} + da_x.x^{18} + 14_x.x^{19} + 19_x.x^{20} + ce_x.x^{21}$$
$$+ 2c_x.x^{22} + cf_x.x^{23} + 3d_x.x^{24} + 0c_x.x^{25} + 99_x.x^{26} + fd_x.x^{27} + aa.x^{28}$$
$$+ 65_x.x^{29} + f4_x.x^{30} + 96_x.x^{31} + 32_x.x^{32} + cd_x.x^{33} + 56_x.x^{34} + 97_x.x^{35}$$
$$+ ba_x.x^{36} + 39.x^{37} + 84_x.x^{38} + 9b_x.x^{39} + 1a_x.x^{40} + 6a_x.x^{41} + a5_x.x^{42}$$
$$+ 1c_x.x^{43} + c8_x.x^{44} + 8e_x.x^{45} + 90_x.x^{46} + 4c_x.x^{47} + 96_x.x^{48} + 8a_x.x^{49}$$
$$+ e1_x.x^{50} + d5_x.x^{51} + 46_x.x^{52} + 1e_x.x^{53} + 04_x.x^{54} + 16_x.x^{55} + 30_x.x^{56}$$
$$+ c1_x.x^{58} + 90_x.x^{59} + 90_x.x^{60} + 10_x.x^{61} + 06_x.x^{62} + b2_x.x^{63} + 12_x.x^{64}$$
$$+ 69_x.x^{65} + 3e_x.x^{66} + 76_x.x^{67} + 73_x.x^{68} + 4e_x.x^{69} + c8_x.x^{70} + 78_x.x^{71}$$
$$+ 67_x.x^{72} + fa_x.x^{73} + 4c_x.x^{74} + 3c_x.x^{75} + e1_x.x^{76} + a5_x.x^{77} + 4d_x.x^{78}$$
$$+ 9a_x.x^{79} + df_x.x^{80} + 95_x.x^{81} + a3_x.x^{82} + 87_x.x^{83} + 2c_x.x^{84} + 91_x.x^{85}$$
$$+ ac_x.x^{86} + 4a_x.x^{87} + bc_x.x^{88} + ba_x.x^{89} + 57_x.x^{90} + f7_x.x^{91} + e5_x.x^{92}$$
$$+ 3c_x.x^{93} + c2_x.x^{94} + f8_x.x^{95} + c7_x.x^{96} + ce_x.x^{97} + e0_x.x^{98} + 28_x.x^{99}$$
$$+ 4b_x.x^{100} + 53_x.x^{101} + cb_x.x^{102} + 71_x.x^{103} + 26_x.x^{104} + e1_x.x^{105} +$$
$$e3_x.x^{106} + a7_x.x^{107} + eb_x.x^{108} + 03_x.x^{109} + ff_x.x^{110} + 70_x.x^{111} +$$
$$a9_x.x^{112} + 30_x.x^{113} + 69_x.x^{114} + 94_x.x^{115} + 4c_x.x^{116} + 6c_x.x^{117} +$$
$$08_x.x^{118} + 2f_x.x^{119} + 5a_x.x^{120} + 86_x.x^{121} + a3_x.x^{122} + 27_x.x^{123} +$$
$$dd_x.x^{124} + e9_x.x^{125} + c5_x.x^{126} + 46_x.x^{127} + 50_x.x^{128} + 95_x.x^{129} +$$
$$66_x.x^{130} + 31_x.x^{131} + c1_x.x^{132} + 1b_x.x^{133} + 40_x.x^{134} + f9_x.x^{135} +$$
$$af_x.x^{136} + 5b_x.x^{137} + 10_x.x^{138} + 58_x.x^{139} + de_x.x^{140} + 5b_x.x^{141} +$$
$$18_x.x^{142} + a7_x.x^{143} + 99_x.x^{144} + be_x.x^{145} + 78_x.x^{146} + 20_x.x^{147} +$$
$$86_x.x^{148} + e2_x.x^{149} + 11_x.x^{150} + 1b_x.x^{151} + 2e_x.x^{152} + f1_x.x^{154} +$$
$$cc_x.x^{155} + e7_x.x^{156} + a1_x.x^{157} + b0_x.x^{158} + bd_x.x^{159} + 93_x.x^{160} +$$
$$08_x.x^{161} + 9b_x.x^{162} + 1a_x.x^{163} + b2_x.x^{164} + 6c_x.x^{165} + ed_x.x^{166} +$$
$$96_x.x^{167} + df_x.x^{168} + d1_x.x^{169} + 7e_x.x^{170} + e8_x.x^{171} + ac_x.x^{172} +$$
$$10_x.x^{173} + 4d_x.x^{174} + ab_x.x^{175} + 3e_x.x^{176} + e4_x.x^{177} + 42_x.x^{178} +$$
$$28_x.x^{179} + 92_x.x^{180} + e7_x.x^{181} + b9_x.x^{182} + 61_x.x^{183} + 20_x.x^{184} +$$
$$7a_x.x^{185} + f6_x.x^{186} + e6_x.x^{187} + 29_x.x^{188} + 7b_x.x^{189} + f9_x.x^{190} +$$
$$34_x.x^{191} + 1f_x.x^{192} + f5_x.x^{193} + b0_x.x^{194} + 8b_x.x^{195} + a6_x.x^{196} +$$
$$09_x.x^{197} + 8f_x.x^{198} + bb_x.x^{199} + 35_x.x^{200} + bb_x.x^{201} + f4_x.x^{202} +$$
$$b3_x.x^{203} + a9_x.x^{204} + a3_x.x^{205} + 74_x.x^{206} + 3b_x.x^{207} + 1c_x.x^{208} +$$
$$25_x.x^{209} + 28_x.x^{210} + 64_x.x^{211} + 2c_x.x^{212} + be_x.x^{213} + 9f_x.x^{214} +$$
$$b6_x.x^{215} + 8f_x.x^{216} + 0e_x.x^{217} + fa_x.x^{218} + 12_x.x^{219} + 84_x.x^{220} +$$
$$70_x.x^{221} + 64_x.x^{222} + ca_x.x^{223} + 36_x.x^{224} + 17_x.x^{225} + 78_x.x^{226} +$$
$$e1_x.x^{227} + 32_x.x^{228} + 4b_x.x^{229} + 18_x.x^{230} + 68_x.x^{231} + a2_x.x^{232} +$$
$$fc_x.x^{233} + 38_x.x^{234} + fb_x.x^{235} + 64_x.x^{236} + db_x.x^{237} + 85_x.x^{238} +$$
$$1c_x.x^{239} + 11_x.x^{240} + d1_x.x^{241} + 6d_x.x^{242} + d7_x.x^{243} + d5_x.x^{244} +$$
$$02_x.x^{245} + 95_x.x^{246} + 44_x.x^{247} + a6_x.x^{248} + 39_x.x^{249} + 06_x.x^{250} +$$
$$5f_x.x^{251} + f1_x.x^{252} + 78_x.x^{253} + 1e_x.x^{254}$$

Additionally, algebraic expressions of low algebraic degree such as quadratic expressions are of particular interest [21, 82]. Courtois and Pieprzyk in [21] demonstrated how to check the existence of quadratic expressions for S-boxes.

For Serpent S-box S_0, 21 linearly independent quadratic algebraic relations were obtained (as predicted for 4×4-bit S-boxes). These algebraic relations are multivariate equations holding with certainty and mixing both the input

and the output variables. A common notation for all of Serpent S-boxes: the input is denoted (x_3, x_2, x_1, x_0) and the output is denoted (y_3, y_2, y_1, y_0). The following 21 multivariate quadratic relations for Serpent S_0 were obtained[1]:

- $x_0 + x_1 + x_2 + x_3 + y_3 + x_0.x_3 = 0,$
- $x_2 + x_3 + y_0 + y_1 + x_0.x_1 + x_1.x_3 = 0,$
- $x_0 + x_0.x_1 + x_0.x_2 + x_0.y_3 = 0,$
- $x_0.x_1 + x_1.x_2 + x_1.y_0 + x_1.y_1 = 0,$
- $1 + y_0 + y_3 + x_0.x_2 + x_0.y_1 + x_0.y_2 + x_1.x_2 + x_1.y_2 = 0,$
- $x_0 + x_3 + y_0 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.y_2 + x_1.x_2 + x_1.y_0 + x_1.y_3 = 0,$
- $x_3 + y_2 + x_0.x_2 + x_0.y_0 + x_0.y_1 + x_0.y_2 + x_1.y_0 = 0,$
- $x_0 + x_3 + y_0 + y_1 + y_2 + y_3 + x_0.x_2 + x_0.y_1 + x_0.y_2 + x_1.x_2 + x_2.x_3 + x_2.y_0 = 0,$
- $x_0 + x_1 + x_3 + y_3 + x_0.y_1 + x_0.y_2 + x_1.x_2 + x_2.y_1 = 0,$
- $x_0.x_2 + x_1.x_2 + x_2.x_3 + x_2.y_2 = 0,$
- $1 + x_0 + x_1 + x_2 + x_3 + y_1 + y_2 + x_0.x_1 + x_0.x_2 + x_2.x_3 + x_2.y_3 = 0,$
- $1 + x_2 + y_1 + y_2 + y_3 + x_0.x_1 + x0.x_2 + x_0.y_1 + x_1.x_2 + x_2.x_3 + x_3.y_0 = 0,$
- $1 + x_0 + x_1 + x_2 + y_0 + x_0.x_1 + x0.x_2 + x_0.y_1 + x_0.y_2 + x_1.x_2 + x_1.y_0 + x_3.y_1 = 0,$
- $1 + x_1 + x_3 + y_0 + y_3 + x_0.x_1 + x_0.y_2 + x_1.x_2 + x_3.y_2 = 0,$
- $x_2 + y_0 + y_1 + x_0.x_1 + x_2.x_3 + x_3.y_3 = 0,$
- $1 + x_2 + y_0 + y_1 + y_2 + y_3 + y_0.y_1 = 0,$
- $1 + x_0 + x_1 + y_1 + x_0.x_1 + x_0.x_2 + x_0.y_1 + x_0.y_2 + x_1.x_2 + x_2.x_3 = 0,$
- $1 + x_1 + x_3 + y_0 + y_2 + y_3 + x_0.x_1 = 0,$
- $x_2 + x_3 + y_0 + y_1 + y_2 + x_0.x_1 + x_0.y_1 + x_0.y_2 + x_1.x_2 = 0,$

[1] Each of them holds with certainty and they are all linearly independent relations. These same properties apply to the quadratic relations of all the other S-boxes of Serpent.

- $x_0 + x_1 + x_2 + y_0 + y_1 + y_2 + y_3 + x_0.x_1 = 0,$
- $1 + x_0 + x_3 + y_1 + x_0.x_1 + x_0.y_1 + x_1.y_0 + x_2.x_3 = 0.$

For Serpent S-box S_1, the following 21 quadratic expressions were obtained:

- $1 + x_1 + x_2 + x_3 + y_2 + x_0.x_1 = 0,$
- $x_0 + x_1 + x_2 + y_1 + y_3 + x_0.x_1 + x_0.x_3 + x_1.x_3 = 0,$
- $x_0 + x_0.x_2 + x_0.x_3 + x_0.y_2 = 0,$
- $x_0 + x_1 + x_2 + y_2 + y_3 + x_0.y_0 + x_0.y_3 + x_1.y_0 + x_1.y_1 = 0,$
- $x_0 + x_1 + x_2 + y_1 + y_3 + x_0.x_3 + x_1.x_2 + x_1.y_2 = 0,$
- $x_0+x_1+y_1+y_2+x_0.x_1+x_0.x_2+x_0.y_0+x_0.y_1+x_1.x_2+x_1.y_0+x_1.y_3 = 0,$
- $1+x_1+x_2+y_0+y_2+y_3+x_0.x_1+x_0.x_3+x_0.y_1+x_0.y_3+x_1.x_2+x_2.x_3 = 0,$
- $1 + x_1 + x_2 + y_0 + x_0.x_3 + x_0.y_3 + x_1.y_0 + x_2.y_0 = 0,$
- $1 + x_0 + x_1 + x_2 + y_0 + y_2 + y_3 + x_0.x_2 + x_0.y_0 + x_0.y_1 + x_1.y_0 + x_2.y_1 = 0,$
- $1 + x_0 + x_1 + x_2 + y_0 + y_2 + y_3 + x_0.x_1 + x_0.x_2 + x_0.x_3 + x_0.y_0 + x_1.x_2 + x_1.y_0 + x_2.y_2 = 0,$
- $1 + x_0 + x_1 + x_2 + y_0 + x_0.x_1 + x_0.x_2 + x_0.y_1 + x_2.y_3 = 0,$
- $x_1 + x_2 + y_0 + y_1 + x_0.x_1 + x_0.x_2 + x_0.y_1 + x_1.y_0 + x_3.y_0 = 0,$
- $1+x_0+x_2+y_0+y_1+y_3+x_0.x_1+x_0.x_2+x_0.x_3+x_0.y_3+x_1.y_0+x_3.y_1 = 0,$
- $1 + y_0 + y_1 + y_2 + x_0.x_2 + x_0.x_3 + x_1.x_2 + x_3.y_2 = 0,$
- $x_1 + y_1 + y_2 + x_0.x_3 + x_0.y_3 + x_1.x_2 + x_1.y_0 + x_3.y_3 = 0,$
- $1 + x_0 + y_0 + y_1 + x_0.x_1 + x_0.x_3 + y_0.y_1 = 0,$
- $1 + x_0 + y_0 + y_1 + x_0.x_1 + x_0.x_3 + x_0.y_0 + x_1.y_0 = 0,$
- $x_1 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.x_3 = 0,$
- $x_2 + y_3 + x_0.x_2 + x_0.y_3 + x_1.y_0 = 0,$

- $1 + x_1 + x_2 + y_0 + y_3 + x_0.x_1 = 0,$
- $1 + x_1 + x_2 + y_0 + y_3 + x_0.y_3 + x_1.x_2 + x_1.y_0 = 0.$

For Serpent S-box S_2, the following 21 quadratic expressions were obtained:

- $1 + x_0 + x_1 + x_3 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.y_0 = 0,$
- $x_0 + x_2 + x_3 + y_1 + y_2 + x_0.x_1 + x_0.y_2 + x_1.x_3 = 0,$
- $1 + x_0 + x_2 + y_3 + x_1.x_2 + x_1.y_0 = 0,$
- $x_1 + x_2 + x_3 + y_0 + x_0.x_2 = 0,$
- $x_1 + x_2 + x_3 + y_0 + x_0.y_1 + x_0.y_2 + x_1.x_2 + x_1.y_1 + x_1.y_2 = 0,$
- $1 + x_0 + x_1 + x_2 + y_3 + x_0.x_1 + x_1.x_2 + x_1.y_3 = 0,$
- $1 + x_0 + x_1 + x_3 + y_1 + y_2 + y_3 + x_0.x_3 = 0,$
- $x_1 + x_3 + y_0 + x_1.x_2 + x_2.x_3 + x_2.y_0 = 0,$
- $x_0.x_1 + x_0.y_1 + x_0.y_2 + x_0.y_3 = 0,$
- $x_2 + y_1 + x_0.y_1 + x_1.x_2 + x_1.y_1 + x_2.x_3 + x_2.y_2 = 0,$
- $1+x_2+x_3+y_1+y_2+y_3+x_0.y_2+x_1.x_2+x_1.y_1+x_2.x_3+x_2.y_1+x_2.y_3 = 0,$
- $1 + x_0 + x_1 + x_2 + y_0 + y_1 + y_3 + x_0.x_1 + x_0.y_1 + x_1.x_2 + x_3.y_0 = 0,$
- $1 + x_0 + x_1 + y_1 + y_3 + x_0.y_2 + x_1.x_2 + x_1.y_1 + x_2.x_3 + x_3.y_1 = 0,$
- $1+x_0+x_2+x_3+y_0+y_2+y_3+x_0.y_2+x_1.x_2+x_1.y_1+x_2.x_3+x_2.y_1+x_3.y_2 = 0,$
- $x_0 + x_1 + y_0 + x_0.x_1 + x_0.y_2 + x_2.y_1 + x_3.y_3 = 0,$
- $x_0 + x_1 + x_3 + y_1 + y_2 + x_0.y_1 + x_2.x_3 + y_0.y_1 = 0,$
- $x_0 + x_1 + x_3 + y_1 + x_0.y_1 + x_0.y_2 + x_1.y_1 + x_2.x_3 + x_2.y_1 = 0,$
- $1 + x_1 + x_2 + y_0 + y_3 + x_0.x_1 + x_0.y_2 + x_1.y_1 + x_2.y_1 = 0,$
- $x_1 + x_3 + y_0 + y_1 + x_2.x_3 = 0,$

- $x_0 + x_1 + x_3 + y_2 + x_2.x_3 = 0,$
- $1 + x_0 + x_2 + x_3 + y_0 + y_1 + y_2 + y_3 + x_2.x_3 = 0.$

For Serpent S-box S_3, the following 21 quadratic expressions were obtained:

- $x_0.x_1 + x_0.x_2 + x_0.y_2 + x_0.y_3 = 0,$
- $x_0 + y_1 + y_2 + y_3 + x_0.x_3 + x_1.x_3 = 0,$
- $x_1 + x_2 + x_3 + y_2 + x_0.x_1 + x_0.y_0 + x_1.x_2 + x_1.y_0 = 0,$
- $x_1 + x_2 + y_0 + y_3 + x_0.x_1 + x_0.x_2 + x_0.y_0 + x_0.y_3 + x_1.x_2 + x_1.y_1 = 0,$
- $x_2 + x_3 + y_1 + y_3 + x_0.x_1 + x_0.x_3 + x_1.x_2 + x_1.y_2 = 0,$
- $x_1 + x_2 + y_0 + y_1 + y_2 + x_0.x_2 + x_0.y_0 + x_1.y_3 = 0,$
- $x_2 + x_3 + y_1 + y_3 + x_0.x_2 + x_0.x_3 + x_0.y_1 + x_0.y_3 = 0,$
- $x_0 + x_2 + y_0 + y_3 + x_0.x_1 + x_0.y_3 + x_1.x_2 + x_2.y_0 = 0,$
- $x_0.x_1 + x_0.x_3 + x_0.y_0 + x_0.y_3 + x_1.x_2 + x_2.x_3 + x_2.y_1 = 0,$
- $x_0 + y_0 + y_3 + x_0.x_3 + x_0.y_0 + x_1.x_2 + x_2.x_3 + x_2.y_2 = 0,$
- $x_1 + x_3 + y_3 + x_0.x_1 + x_0.x_2 + x_0.x_3 + x_0.y_3 + x_1.x_2 + x_2.x_3 + x_2.y_3 = 0,$
- $x_0 + x_1 + x_2 + y_1 + y_2 + x_0.x_1 + x_0.x_2 + x_0.y_3 + x_3.y_0 = 0,$
- $x_2 + x_3 + y_2 + x_0.x_1 + x_0.x_3 + x_0.y_3 + x_2.x_3 + x_3.y_1 = 0,$
- $x_0 + x_3 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.x_3 + x_0.y_0 + x_0.y_3 + x_2.x_3 + x_3.y_2 = 0,$
- $x_2 + y_2 + x_0.x_3 + x_0.y_0 + x_3.y_3 = 0,$
- $x_1 + x_2 + y_0 + y_1 + y_2 + x_0.x_1 + x_0.x_3 + y_0.y_1 = 0,$
- $x_0 + x_1 + x_2 + y_1 + y_2 + x_0.x_3 = 0,$
- $x_0 + x_3 + y_1 + x_0.x_2 + x_0.y_3 + x_1.x_2 = 0,$
- $x_0 + x_3 + y_0 + y_1 + x_0.x_1 + x_0.x_3 = 0,$

- $x_1 + y_1 + y_2 + y_3 + x_0.x_2 + x_1.x_2 + x_2.x_3 = 0,$
- $x_1 + y_1 + y_2 + x_0.x_2 + x_0.y_3 + x_1.x_2 + x_2.x_3 = 0.$

For Serpent S-box S_4, the following 21 quadratic expressions were obtained:

- $1 + x_3 + y_0 + y_3 + x_0.x_1 + x_0.x_2 + x_0.y_0 + x_1.x_2 = 0,$
- $1 + x_1 + x_2 + x_3 + y_0 + x_0.x_1 + x_0.x_3 + x_1.x_3 = 0,$
- $1 + x_0 + x_3 + y_0 + y_3 + x_1.y_0 = 0,$
- $1 + x_3 + y_0 + y_3 + x_0.x_2 + x_0.x_3 + x_0.y_1 + x_1.y_1 = 0,$
- $1 + x_1 + x_2 + x_3 + y_0 + x_0.y_0 + x_0.y_3 + x_1.y_2 = 0,$
- $1 + x_2 + x_3 + y_0 + x_0.x_3 + x_1.y_3 = 0,$
- $x_2 + x_3 + y_1 + y_2 + x_0.x_2 + x_0.x_3 + x_0.y_1 + x_0.y_2 = 0,$
- $1 + x_1 + x_2 + y_0 + y_1 + x_0.y_3 + x_2.y_0 = 0,$
- $1 + x_2 + y_0 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.x_3 + x_0.y_3 + x_2.y_1 = 0,$
- $1 + x_0 + x_2 + x_3 + y_0 + y_3 + x_0.x_2 + x_0.y_1 + x_0.y_3 + x_2.x_3 + x_2.y_2 = 0,$
- $x_1 + x_3 + y_1 + y_3 + x_0.x_1 + x_0.x_3 + x_0.y_1 + x_0.y_3 + x_2.x_3 + x_2.y_3 = 0,$
- $x_0 + x_0.x_2 + x_0.x_3 + x_0.y_0 + x_2.x_3 + x_3.y_0 = 0,$
- $1 + x_1 + x_2 + y_0 + x_0.x_1 + x_2.x_3 + x_3.y_1 = 0,$
- $1 + x_0 + x_1 + x_2 + x_3 + y_0 + x_0.x_2 + x_0.y_0 + x_0.y_1 + x_0.y_3 + x_3.y_2 = 0,$
- $x_1 + y_2 + y_3 + x_0.y_0 + x_0.y_3 + x_3.y_3 = 0,$
- $1 + x_2 + y_0 + y_1 + y_2 + y_3 + x_0.x_3 + y_0.y_1 = 0,$
- $x_0 + x_1 + y_2 + y_3 + x_0.x_2 + x_0.x_3 + x_2.x_3 = 0,$
- $1 + x_3 + y_0 + y_3 + x_0.x_2 + x_0.x_3 + x_2.x_3 = 0,$
- $x_0 + x_1 + y_2 + y_3 + x_0.x_1 + x_0.x_2 + x_0.x_3 + x_0.y_0 + x_0.y_3 + x_2.x_3 = 0,$

- $1 + x_1 + x_3 + y_0 + y_2 + x_0.x_2 + x_0.x_3 + x_0.y_0 + x_0.y_3 + x_2.x_3 = 0$,
- $1 + x_0 + x_1 + y_0 + y_1 + x_0.y_0 = 0$.

For Serpent S-box S_5, the following 21 quadratic expressions were obtained:

- $1 + x_0 + x_1 + x_2 + x_3 + y_3 + x_0.x_1 + x_0.x_2 + x_0.x_3 + x_0.y_3 = 0$,
- $1 + x_1 + x_2 + x_3 + y_0 + x_0.x_1 + x_0.x_3 + x_1.x_3 = 0$,
- $x_0 + x_0.x_1 + x_0.x_2 + x_0.y_0 + x_1.x_2 + x_1.y_0 = 0$,
- $1+x_1+x_3+y_0+y_1+y_2+x_0.x_1+x_0.x_2+x_0.y_0+x_0.y_1+x_1.x_2+x_1.y_1 = 0$,
- $y_0 + y_3 + x_0.x_1 + x_0.y_1 + x_0.y_2 + x_1.y_2 = 0$,
- $y_0 + y_3 + x_0.y_1 + x_0.y_2 + x_1.x_2 + x_1.y_3 = 0$,
- $x_1 + y_0 + y_1 + x_0.x_2 + x_0.x_3 + x_0.y_0 + x_2.x_3 = 0$,
- $x_2 + y_2 + y_3 + x_0.x_1 + x_0.x_2 + x_0.y_0 + x_0.y_1 + x_1.x_2 + x_2.y_0 = 0$,
- $x_1 + x_2 + y_0 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.x_3 + x_0.y_0 + x_0.y_2 + x_2.y_1 = 0$,
- $1+x_0+x_2+x_3+y_0+y_1+y_2+x_0.x_1+x_0.y_0+x_0.y_2+x_1.x_2+x_2.y_2 = 0$,
- $1 + x_0 + x_2 + x_3 + y_0 + y_1 + y_3 + x_0.x_1 + x_0.x_2 + x_0.x_3 + x_0.y_0 + x_0.y_1 + x_1.x_2 + x_2.y_3 = 0$,
- $x_0 + x_1 + y_0 + y_1 + x_3.y_0 = 0$,
- $1 + x_1 + x_2 + x_3 + y_0 + x_0.x_1 + x_3.y_1 = 0$,
- $x_2 + y_0 + y_2 + x_0.y_1 + x_3.y_2 = 0$,
- $1 + x_2 + x_3 + y_1 + x_0.x_1 + x_0.y_1 + x_0.y_2 + x_3.y_3 = 0$,
- $x_1 + y_1 + x_0.x_1 + x_0.x_2 + x_1.x_2 + y_0.y_1 = 0$,
- $1 + x_0 + x_3 + y_1 + y_2 + x_0.x_1 + x_0.x_2 + x_1.x_2 = 0$,
- $x_0 + x_1 + x_2 + y_1 + y_2 + y_3 + x_0.x_1 = 0$,
- $1 + x_0 + x_1 + x_3 + y_2 + y_3 + x_0.x_1 + x_0.x_2 + x_0.y_0 + x_1.x_2 = 0$,

- $x_0 + x_1 + y_0 + y_1 + y_3 + x_0.x_1 + x_0.x_2 + x_0.y_1 + x_1.x_2 = 0,$
- $x_1 + x_2 + y_0 + y_1 + y_2 + x_0.x_2 + x_0.x_3 + x_0.y_1 + x_1.x_2 = 0.$

For Serpent S-box S_6, the following 21 quadratic expressions were obtained:

- $1 + x_0 + x_1 + x_2 + y_1 + x_0.x_1 + x_0.x_2 + x_0.y_1 = 0,$
- $1 + x_1 + x_2 + y_1 + x_0.x_3 = 0,$
- $x_0 + x_3 + y_0 + y_1 + x_0.y_2 + x_0.y_3 + x_1.x_2 + x_1.x_3 + x_1.y_0 = 0,$
- $1 + x_0 + x_1 + x_3 + y_2 + y_3 + x_0.x_2 + x_1.x_3 + x_1.y_1 = 0,$
- $x_0 + x_1 + x_2 + x_3 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.y_2 + x_0.y_3 + x_1.x_2 + x_1.y_2 = 0,$
- $x_1 + x_0.x_1 + x_1.x_2 + x_1.x_3 + x_1.y_3 = 0,$
- $x_1 + x_3 + y_0 + y_2 + x_0.x_1 + x_0.x_2 + x_1.x_3 + x_2.x_3 = 0,$
- $x_0 + x_1 + y_1 + y_2 + x_0.y_0 + x_1.x_2 + x_1.x_3 + x_2.y_0 = 0,$
- $x_0 + x_2 + x_3 + y_1 + y_2 + y_3 + x_0.y_0 + x_0.y_2 + x_1.x_3 + x_2.y_1 = 0,$
- $x_0 + x_1 + x_2 + y_0 + y_1 + y_3 + x_0.x_2 + x_0.y_0 + x_2.y_2 = 0,$
- $1 + x_1 + x_2 + y_0 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.y_2 + x_0.y_3 + x_1.x_2 + x_1.x_3 + x_2.y_3 = 0,$
- $x_0 + x_1 + x_2 + y_0 + y_1 + y_3 + x_0.x_2 + x_0.y_2 + x_1.x_2 + x_1.x_3 + x_3.y_0 = 0,$
- $1 + x_2 + y_0 + y_1 + y_2 + x_0.x_1 + x_0.x_2 + x_3.y_1 = 0,$
- $1 + x_0 + x_1 + y_2 + y_3 + x_0.x_1 + x_0.x_2 + x_0.y_0 + x_1.x_2 + x_3.y_2 = 0,$
- $x_0 + y_0 + y_1 + x_0.x_2 + x_0.y_3 + x_1.x_2 + x_1.x_3 + x_3.y_3 = 0,$
- $1 + x_0 + y_3 + x_0.x_1 + x_0.x_2 + x_0.y_2 + x_0.y_3 + x_1.x_2 + x_1.x_3 + y_0.y_1 = 0,$
- $1 + x_1 + y_1 + y_2 + y_3 = 0,$
- $1 + x_3 + y_1 + y_3 + x_0.x_1 + x_0.x_2 = 0,$
- $1 + x_1 + x_2 + x_3 + y_0 + y_1 + x_0.x_2 + x_0.y_2 + x_0.y_3 + x_1.x_2 = 0,$

- $x_0 + x_1 + x_2 + x_3 + y_0 + y_1 + y_3 + x_0.y_0 + x_0.y_3 + x_1.x_2 + x_1.x_3 = 0,$
- $x_0 + x_1 + y_0 + y_1 + x_0.x_1 + x_0.x_2 = 0.$

For Serpent S-box S_7, the following 21 quadratic expressions were obtained:

- $x_0 + x_3 + y_1 + y_3 + x_0.x_1 + x_0.y_1 + x_1.x_2 = 0,$
- $x_1 + x_2 + y_3 + x_0.x_1 + x_0.x_2 + x_0.y_3 = 0,$
- $x_1 + x_3 + y_1 + y_3 + x_0.x_1 + x_0.x_2 + x_0.x_3 + x_0.y_0 + x_0.y_1 + x_1.x_3 + x_1.y_0 = 0,$
- $x_0 + x_1 + x_0.x_2 + x_0.y_2 + x_1.x_3 + x_1.y_1 = 0,$
- $x_0 + x_1 + y_1 + y_2 + x_0.x_2 + x_0.x_3 + x_0.y_0 + x_0.y_1 + x_0.y_2 + x_1.x_3 + x_1.y_2 = 0,$
- $x_2 + x_3 + y_1 + x_0.x_2 + x_0.y_0 + x_0.y_1 + x_0.y_2 + x_1.y_3 = 0,$
- $1 + x_0 + x_1 + x_3 + y_0 + y_2 + x_0.x_1 + x_0.y_1 + x_2.x_3 = 0,$
- $1 + x_2 + x_3 + y_0 + y_2 + y_3 + x_0.x_1 + x_0.x_2 + x_0.x_3 + x_0.y_1 + x_2.y_0 = 0,$
- $1 + x_1 + x_2 + x_3 + y_0 + y_2 + x_0.y_1 + x_0.y_2 + x_2.y_1 = 0,$
- $1 + x_0 + x_1 + x_2 + y_0 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.x_3 + x_0.y_0 + x_0.y_1 + x_2.y_2 = 0,$
- $x_1 + x_3 + y_1 + x_0.x_1 + x_0.x_2 + x_0.y_0 + x_0.y_2 + x_2.y_3 = 0,$
- $y_2 + y_3 + x_0.x_2 + x_0.y_0 + x_0.y_1 + x_0.y_2 + x_3.y_0 = 0,$
- $1 + x_2 + x_3 + y_0 + x_0.x_1 + x_0.x_3 + x_0.y_0 + x_0.y_1 + x_0.y_2 + x_3.y_1 = 0,$
- $1 + x_0 + x_2 + x_3 + y_0 + x_0.x_2 + x_0.x_3 + x_0.y_1 + x_0.y_2 + x_1.x_3 + x_3.y_2 = 0,$
- $1 + x_0 + x_2 + x_3 + y_0 + y_2 + y_3 + x_0.y_0 + x_1.x_3 + x_3.y_3 = 0,$
- $1 + x_3 + y_0 + y_1 + y_2 + y_3 + x_0.x_1 + x_0.x_3 + y_0.y_1 = 0,$
- $1 + x_0 + y_0 + y_1 + x_1.x_3 = 0,$
- $1 + x_1 + x_2 + y_0 + y_1 + y_2 + x_1.x_3 = 0,$
- $x_0 + y_2 + x_0.x_1 + x_0.x_2 + x_0.x_3 + x_0.y_0 + x_0.y_2 + x_1.x_3 = 0,$

- $1 + x_1 + x_3 + y_0 + y_2 + y_3 + x_0.y_0 + x_0.y_2 + x_1.x_3 = 0,$
- $1 + x_0 + x_1 + y_0 + x_0.x_1 + x_0.y_1 + x_1.x_3 = 0.$

Concerning the ANF of S-boxes output bits (as Boolean mappings), the following are the ANFs of SERPENT S-boxes S_i, for $0 \leq i \leq 7$, where the input is denoted (x_3, x_2, x_1, x_0) and the output is denoted (y_3, y_2, y_1, y_0).

For S-box S_0:

$y_3 = x_0 + x_1 + x_2 + x_3 + x_0.x_3,$
$y_2 = x_1 + x_0.x_1 + x_0.x_2 + x_0.x_1.x_2 + x_3 + x_1.x_3 + x_1.x_2.x_3,$
$y_1 = 1 + x_0 + x_0.x_2 + x_1.x_2 + x_0.x_1.x_2 + x_1.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3,$
$y_0 = 1 + x_0 + x_0.x_1 + x_2 + x_0.x_2 + x_1.x_2 + x_0.x_1.x_2 + x_3 + x_0.x_2.x_3 + x_1.x_2.x_3.$

For S-box S_1:

$y_3 = 1 + x_1 + x_0.x_2 + x_3 + x_0.x_3 + x_0.x_1.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3,$
$y_2 = 1 + x_1 + x_0.x_1 + x_2 + x_3,$
$y_1 = 1 + x_0 + x_0.x_1 + x_2 + x_0.x_2 + x_3 + x_1.x_3 + x_0.x_1.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3,$
$y_0 = 1 + x_0 + x_1 + x_1.x_2 + x_0.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3.$

For S-box S_2:

$y_3 = 1 + x_0 + x_1 + x_2 + x_0.x_1.x_2 + x_1.x_3,$
$y_2 = x_0 + x_1 + x_1.x_2 + x_3 + x_1.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_2.x_3,$
$y_1 = x_0 + x_1 + x_2 + x_1.x_2 + x_0.x_1.x_2 + x_0.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_2.x_3,$
$y_0 = x_1 + x_2 + x_0.x_2 + x_3.$

For S-box S_3:

$y_3 = x_0 + x_1 + x_0.x_1 + x_2 + x_0.x_2 + x_0.x_1.x_2 + x_3 + x_2.x_3 + x_0.x_2.x_3,$
$y_2 = x_0 + x_0.x_1 + x_2 + x_0.x_1.x_2 + x_3 + x_1.x_3 + x_0.x_1.x_3,$
$y_1 = x_0 + x_1 + x_0.x_2 + x_0.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_2.x_3,$
$y_0 = x_0 + x_1 + x_1.x_2 + x_3 + x_0.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3.$

For S-box S_4:

$y_3 = x_0 + x_1 + x_2 + x_1.x_2 + x_0.x_3 + x_1.x_3 + x_0.x_1.x_3,$
$y_2 = x_0 + x_0.x_1 + x_2 + x_1.x_2 + x_0.x_1.x_2 + x_1.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_1.x_2.x_3,$
$y_1 = x_0 + x_0.x_2 + x_1.x_2 + x_3 + x_1.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3,$
$y_0 = 1 + x_1 + x_0.x_1 + x_2 + x_3 + x_0.x_3 + x_1.x_3.$

For S-box S_5:

$y_3 = 1 + x_0 + x_1 + x_2 + x_0.x_1.x_2 + x_3 + x_0.x_3 + x_0.x_2.x_3,$
$y_2 = 1 + x_1 + x_0.x_2 + x_3 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3,$
$y_1 = 1 + x_0 + x_0.x_1 + x_2 + x_3 + x_1.x_3 + x_0.x_1.x_3 + x_2.x_3,$
$y_0 = 1 + x_1 + x_0.x_1 + x_2 + x_3 + x_0.x_3 + x_1.x_3.$

For S-box S_6:

$y_3 = x_1 + x_0.x_1 + x_2 + x_0.x_2 + x_0.x_1.x_2 + x_3 + x_2.x_3 + x_1.x_2.x_3,$
$y_2 = 1 + x_0 + x_0.x_1 + x_2 + x_1.x_2 + x_0.x_1.x_2 + x_1.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_1.x_2.x_3,$
$y_1 = 1 + x_1 + x_2 + x_0.x_3,$
$y_0 = 1 + x_0 + x_1 + x_2 + x_0.x_2 + x_1.x_2 + x_0.x_1.x_2 + x_3 + x_0.x_1.x_3 + x_1.x_2.x_3.$

For S-box S_7:

$y_3 = x_0 + x_1 + x_2 + x_0.x_2 + x_0.x_1.x_2 + x_0.x_3,$
$y_2 = x_0 + x_1 + x_2 + x_0.x_1.x_2 + x_3 + x_0.x_3 + x_1.x_3 + x_0.x_1.x_3 + x_1.x_2.x_3,$

$y_1 = x_1 + x_0.x_1 + x_2 + x_0.x_2 + x_1.x_2 + x_3 + x_0.x_3 + x_0.x_1.x_3 + x_0.x_2.x_3,$
$y_0 = 1 + x_0.x_1 + x_2 + x_0.x_3 + x_1.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3.$

The ANFs of the 4×4 S-box of the PRESENT cipher are:
$y_3 = x_0 + x_2 + x_1.x_2 + x_3,$
$y_2 = x_1 + x_0.x_1.x_2 + x_3 + x_1.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_2.x_3,$
$y_1 = 1 + x_0.x_1 + x_2 + x_3 + x_0.x_3 + x_1.x_3 + x_0.x_1.x_3 + x_0.x_2.x_3,$
$y_0 = 1 + x_0 + x_1 + x_1.x_2 + x_0.x_1.x_2 + x_3 + x_0.x_1.x_3 + x_0.x_2.x_3.$

The ANFs of the 7×7-bit S-box of Misty1 [59] are:
$y_0 = x_0 + x_1.x_3 + x_0.x_3.x_4 + x_1.x_5 + x_0.x_2.x_5 + x_4.x_5 + x_0.x_1.x_6 + x_2.x_6 + x_0.x_5.x_6 + x_3.x_5.x_6 + 1,$
$y_1 = x_0.x_2 + x_0.x_4 + x_3.x_4 + x_1.x_5 + x_2.x_4.x_5 + x_6 + x_0.x_6 + x_3.x_6 + x_2.x_3.x_6 + x_1.x_4.x_6 + x_0.x_5.x_6 + 1,$
$y_2 = x_1.x_2 + x_0.x_2.x_3 + x_4 + x_1.x_4 + x_0.x_1.x_4 + x_0.x_5 + x_0.x_4.x_5 + x_3.x_4.x_5 + x_1.x_6 + x_3.x_6 + x_0.x_3.x6 + x_4.x_6 + x_2.x_4.x_6,$
$y_3 = x_0 + x_1 + x_0.x_1.x_2 + x_0.x_3 + x_2.x_4 + x_1.x_4.x_5 + x_2.x_6 + x_1.x_3.x_6 + x_0.x_4.x_6 + x_5.x_6 + 1$
$y_4 = x_2.x_3 + x_0.x_4 + x_1.x_3.x_4 + x_5 + x_2.x_5 + x_1.x_2.x_5 + x_0.x_3.x_5 + x_1.x_6 + x_1.x_5.x_6 + x_4.x_5.x_6 + 1,$
$y_5 = x_0 + x_1 + x_2 + x_0.x_1.x_2 + x_0.x_3 + x_1.x_2.x_3 + x_1.x_4 + x_0.x_2.x_4 + x_0.x_5 + x_0.x_1.x_5 + x_3.x_5 + x_0.x_6 + x_2.x_5.x_6,$
$y_6 = x_0.x_1 + x_3 + x_0.x_3 + x_2.x_3.x_4 + x_0.x_5 + x_2.x_5 + x_3.x_5 + x_1.x_3.x_5 + x_1.x_6 + x_1.x_2.x_6 + x_0.x_3.x_6 + x_4.x_6 + x_2.x_5.x_6.$

The ANFs of the 7×7-bit S-box of Kasumi [1] are:
$y_0 = x_1.x_3 + x_4 + x_0.x_1.x_4 + x_5 + x_2.x_5 + x_3.x_4.x_5 + x_6 + x_0.x_6 + x_1.x_6 + x_3.x_6 + x_2.x_4.x_6 + x_1.x_5.x_6 + x_4.x_5.x_6,$
$y_1 = x_0.x_1 + x_0.x_4 + x_2.x_4 + x_5 + x_1.x_2.x_5 + x_0.x_3.x_5 + x_6 + x_0.x_2.x_6 + x_3.x_6 + x_4.x_5.x_6 + 1,$
$y_2 = x_0 + x_0.x_3 + x_2.x_3 + x_1.x_2.x_4 + x_0.x_3.x_4 + x_1.x_5 + x_0.x_2.x_5 + x_0.x_6 + x_0.x_1.x_6 + x_2.x_6 + x_4.x_6 + 1,$
$y_3 = x_1 + x_0.x_1.x_2 + x_1.x_4 + x_3.x_4 + x_0.x_5 + x_0.x_1.x_5 + x_2.x_3.x_5 + x_1.x_4.x_5 + x_2.x_6 + x_1.x_3.x_6,$
$y_4 = x_0.x_2 + x_3 + x_1.x_3 + x_1.x_4 + x_0.x_1.x_4 + x_2.x_3.x_4 + x_0.x_5 + x_1.x_3.x_5 + x_0.x_4.x_5 + x_1.x_6 + x_3.x_6 + x_0.x_3.x_6 + x_5.x_6 + 1,$
$y_5 = x_2 + x_0.x_2 + x_0.x_3 + x_1.x_2.x_3 + x_0.x_2.x_4 + x_0.x_5 + x_2.x_5 + x_4.x_5 + x_1.x_6 + x_1.x_2.x_6 + x_0.x_3.x_6 + x_3.x_4.x_6 + x_2.x_5.x_6 + 1,$
$y_6 = x_1.x_2 + x_0.x_1.x_3 + x_0.x_4 + x_1.x_5 + x_3.x_5 + x_6 + x_0.x_1.x_6 + x_2.x_3.x_6 + x_1.x_4.x_6 + x_0.x_5.x_6.$

The ANFs of the 9×9-bit S-box of Misty1 [59] are:
$y_0 = x_0.x_4 + x_0.x_5 + x_1.x_5 + x_1.x_6 + x_2.x_6 + x_2.x_7 + x_3.x_7 + x_3.x_8 + x_4.x_8 + 1,$
$y_1 = x_0.x_2 + x_3 + x_1.x_3 + x_2.x_3 + x_3.x_4 + x_4.x_5 + x_0.x_6 + x_2.x_6 + x_7 + x_0.x_8 + x_3.x_8 + x_5.x_8 + 1,$
$y_2 = x_0.x_1 + x_1.x_3 + x_4 + x_0.x_4 + x_2.x_4 + x_3.x_4 + x_4.x_5 + x_0.x_6 + x_5.x_6 + x_1.x_7 + x_3.x_7 + x_8,$
$y_3 = x_0 + x_1.x_2 + x_2.x_4 + x_5 + x_1.x_5 + x_3.x_5 + x_4.x_5 + x_5.x_6 + x_1.x_7 + x_6.x_7 + x_2.x_8 + x_4.x_8,$

$y_4 = x_1 + x_0.x_3 + x_2.x_3 + x_0.x_5 + x_3.x_5 + x_6 + x_2.x_6 + x_4.x_6 + x_5.x_6 + x_6.x_7 + x_2.x_8 + x_7.x_8$,
$y_5 = x_2 + x_0.x_3 + x_1.x_4 + x_3.x_4 + x_1.x_6 + x_4.x_6 + x_7 + x_3.x_7 + x_5.x_7 + x_6.x_7 + x_0.x_8 + x_7.x_8$,
$y_6 = x_0.x_1 + x_3 + x_1.x_4 + x_2.x_5 + x_4.x_5 + x_2.x_7 + x_5.x_7 + x_8 + x_0.x_8 + x_4.x_8 + x_6.x_8 + x_7.x_8 + 1$,
$y_7 = x_1 + x_0.x_1 + x_1.x_2 + x_2.x_3 + x_0.x_4 + x_5 + x_1.x_6 + x_3.x_6 + x_0.x_7 + x_4.x_7 + x_6.x_7 + x_1.x_8 + 1$,
$y_8 = x_0 + x_0.x_1 + x_1.x_2 + x_4 + x_0.x_5 + x_2.x_5 + x_3.x_6 + x_5.x_6 + x_0.x_7 + x_0.x_8 + x_3.x_8 + x_6.x_8 + 1$.

For the ANFs of the AES S-box, the input is the ordered 8-tuple (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) and the output is the ordered 8-tuple (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7):

- $y_7 = x_2 + x_0.x_2 + x_1.x_2 + x_0.x_2.x_3 + x_1.x_2.x_3 + x_0.x_1.x_2.x_3 + x_4 + x_0.x_1.x_4 + x_2.x_4 + x_0.x_1.x_2.x_4 + x_0.x_1.x_3.x_4 + x_1.x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_5 + x_0.x_1.x_5 + x_0.x_2.x_5 + x_0.x_1.x_2.x_5 + x_3.x_5 + x_0.x_3.x_5 + x_1.x_3.x_5 + x_2.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_4.x_5 + x_1.x_2.x_4.x_5 + x_3.x_4.x_5 + x_0.x_1.x_3.x_4.x_5 + x_0.x_2.x_3.x_4.x_5 + x_0.x_6 + x_0.x_1.x_6 + x_2.x_6 + x_1.x_2.x_6 + x_0.x_3.x_6 + x_1.x_3.x_6 + x_0.x_1.x_3.x_6 + x_0.x_2.x_3.x_6 + x_0.x_1.x_2.x_3.x_6 + x_4.x_6 + x_2.x_4.x_6 + x_0.x_2.x_4.x_6 + x_1.x_2.x_4.x_6 + x_0.x_1.x_2.x_4.x_6 + x_0.x_3.x_4.x_6 + x_0.x_2.x_3.x_4.x_6 + x_0.x_5.x_6 + x_2.x_5.x_6 + x_0.x_3.x_5.x_6 + x_1.x_2.x_3.x_5.x_6 + x_4.x_5.x_6 + x_0.x_1.x_4.x_5.x_6 + x_1.x_2.x_4.x_5.x_6 + x_3.x_4.x_5.x_6 + x_0.x_3.x_4.x_5.x_6 + x_2.x_3.x_4.x_5.x_6 + x_0.x_2.x_3.x_4.x_5.x_6 + x_7 + x_0.x_7 + x_1.x_7 + x_0.x_2.x_7 + x_0.x_1.x_2.x_7 + x_0.x_3.x_7 + x_0.x_2.x_3.x_7 + x_1.x_2.x_3.x_7 + x_1.x_4.x_7 + x_0.x_1.x_4.x_7 + x_0.x_2.x_4.x_7 + x_0.x_3.x_4.x_7 + x_1.x_3.x_4.x_7 + x_0.x_1.x_3.x_4.x_7 + x_2.x_3.x_4.x_7 + x_0.x_2.x_3.x_4.x_7 + x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_3.x_5.x_7 + x_0.x_3.x_5.x_7 + x_4.x_5.x_7 + x_1.x_4.x_5.x_7 + x_0.x_1.x_4.x_5.x_7 + x_0.x_1.x_2.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_0.x_2.x_3.x_4.x_5.x_7 + x_0.x_6.x_7 + x_2.x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_0.x_3.x_6.x_7 + x_1.x_3.x_6.x_7 + x_0.x_1.x_2.x_3.x_6.x_7 + x_4.x_6.x_7 + x_0.x_4.x_6.x_7 + x_1.x_4.x_6.x_7 + x_2.x_4.x_6.x_7 + x_1.x_2.x_4.x_6.x_7 + x_0.x_3.x_4.x_6.x_7 + x_0.x_1.x_3.x_4.x_6.x_7 + x_0.x_2.x_3.x_4.x_6.x_7 + x_1.x_5.x_6.x_7 + x_0.x_1.x_5.x_6.x_7 + x_2.x_5.x_6.x_7 + x_1.x_2.x_5.x_6.x_7 + x_0.x_3.x_5.x_6.x_7 + x_1.x_3.x_5.x_6.x_7 + x_0.x_1.x_3.x_5.x_6.x_7 + x_0.x_2.x_3.x_5.x_6.x_7 + x_4.x_5.x_6.x_7 + x_1.x_4.x_5.x_6.x_7 + x_0.x_2.x_4.x_5.x_6.x_7 + x_3.x_4.x_5.x_6.x_7 + x_1.x_3.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7 + x_0.x_2.x_3.x_4.x_5.x_6.x_7$,
- $y_6 = 1 + x_3 + x_1.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_4 + x_0.x_1.x_4 + x_0.x_2.x_4 + x_0.x_1.x_2.x_4 + x_1.x_3.x_4 + x_0.x_1.x_3.x_4 + x_2.x_3.x_4 + x_1.x_2.x_3.x_4 + x_5 + x_0.x_5 + x_0.x_1.x_5 + x_0.x_2.x_5 + x_1.x_2.x_5 + x_0.x_1.x_2.x_5 + x_3.x_5 + x_0.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_1.x_2.x_3.x_5 + x_0.x_4.x_5 + x_0.x_2.x_4.x_5 + x_1.x_2.x_4.x_5 + x_0.x_1.x_3.x_4.x_5 + x_2.x_3.x_4.x_5 + x_0.x_2.x_3.x_4.x_5 + x_1.x_2.x_3.x_4.x_5 + x_6 + x_0.x_2.x_6 + x_1.x_2.x_6 + x_0.x_3.x_6 + x_1.x_3.x_6 + x_0.x_1.x_3.x_6 + x_0.x_2.x_3.x_6 + x_1.x_2.x_3.x_6 + x_0.x_1.x_2.x_3.x_6 + x_4.x_6 + x_0.x_4.x_6 + x_1.x_4.x_6 + x_2.x_4.x_6 + x_0.x_2.x_4.x_6 + x_3.x_4.x_6 + x_0.x_3.x_4.x_6 + x_1.x_3.x_4.x_6 + x_2.x_3.x_4.x_6 + x_0.x_5.x_6 + x_1.x_5.x_6 + x_2.x_3.x_5.x_6 + x_0.x_2.x_3.x_5.x_6$

$+ x_0.x_1.x_2.x_3.x_5.x_6 + x_4.x_5.x_6 + x_1.x_2.x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_0.x_1.x_3.x_4.x_5.x_6 + x_0.x_7 + x_1.x_7 + x_1.x_2.x_7 + x_3.x_7 + x_0.x_1.x_3.x_7 + x_2.x_3.x_7 + x_0.x_2.x_3.x_7 + x_0.x_1.x_2.x_3.x_7 + x_0.x_4.x_7 + x_1.x_4.x_7 + x_2.x_4.x_7 + x_1.x_2.x_4.x_7 + x_0.x_3.x_4.x_7 + x_1.x_3.x_4.x_7 + x_1.x_2.x_3.x_4.x_7 + x_0.x_1.x_2.x_3.x_4.x_7 + x_5.x_7 + x_0.x_5.x_7 + x_0.x_2.x_5.x_7 + x_3.x_5.x_7 + x_1.x_3.x_5.x_7 + x_2.x_3.x_5.x_7 + x_1.x_2.x_3.x_5.x_7 + x_1.x_4.x_5.x_7 + x_0.x_1.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_0.x_1.x_2.x_4.x_5.x_7 + x_1.x_3.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_1.x_6.x_7 + x_0.x_1.x_6.x_7 + x_3.x_6.x_7 + x_0.x_1.x_3.x_6.x_7 + x_0.x_4.x_6.x_7 + x_1.x_4.x_6.x_7 + x_0.x_2.x_4.x_6.x_7 + x_0.x_1.x_2.x_4.x_6.x_7 + x_0.x_3.x_4.x_6.x_7 + x_0.x_1.x_3.x_4.x_6.x_7 + x_2.x_3.x_4.x_6.x_7 + x_5.x_6.x_7 + x_0.x_2.x_5.x_6.x_7 + x_1.x_2.x_5.x_6.x_7 + x_0.x_1.x_2.x_5.x_6.x_7 + x_0.x_1.x_3.x_5.x_6.x_7 + x_2.x_3.x_5.x_6.x_7 + x_4.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_1.x_4.x_5.x_6.x_7 + x_0.x_1.x_2.x_4.x_5.x_6.x_7 + x_3.x_4.x_5.x_6.x_7 + x_1.x_3.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7,$

- $y_5 = 1 + x_0.x_1.x_2 + x_0.x_3 + x_0.x_1.x_3 + x_0.x_1.x_2.x_3 + x_4 + x_0.x_1.x_4 + x_2.x_4 + x_0.x_2.x_4 + x_1.x_2.x_4 + x_3.x_4 + x_0.x_1.x_3.x_4 + x_0.x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_1.x_5 + x_0.x_1.x_5 + x_1.x_2.x_5 + x_0.x_1.x_2.x_5 + x_0.x_3.x_5 + x_1.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_1.x_2.x_3.x_5 + x_0.x_1.x_4.x_5 + x_2.x_4.x_5 + x_0.x_2.x_4.x_5 + x_1.x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_3.x_4.x_5 + x_2.x_3.x_4.x_5 + x_0.x_1.x_2.x_3.x_4.x_5 + x_6 + x_1.x_6 + x_0.x_1.x_6 + x_0.x_2.x_6 + x_1.x_2.x_6 + x_1.x_3.x_6 + x_2.x_3.x_6 + x_1.x_2.x_3.x_6 + x_0.x_1.x_2.x_3.x_6 + x_4.x_6 + x_1.x_4.x_6 + x_0.x_1.x_4.x_6 + x_0.x_1.x_2.x_4.x_6 + x_1.x_3.x_4.x_6 + x_2.x_3.x_4.x_6 + x_1.x_2.x_3.x_4.x_6 + x_0.x_5.x_6 + x_1.x_5.x_6 + x_0.x_1.x_5.x_6 + x_1.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_0.x_1.x_2.x_3.x_5.x_6 + x_0.x_1.x_4.x_5.x_6 + x_1.x_2.x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_3.x_4.x_5.x_6 + x_0.x_3.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_2.x_3.x_4.x_5.x_6 + x_7 + x_0.x_1.x_7 + x_1.x_3.x_7 + x_0.x_1.x_3.x_7 + x_2.x_3.x_7 + x_0.x_1.x_2.x_3.x_7 + x_0.x_4.x_7 + x_1.x_4.x_7 + x_0.x_1.x_4.x_7 + x_2.x_4.x_7 + x_1.x_2.x_4.x_7 + x_0.x_3.x_4.x_7 + x_1.x_3.x_4.x_7 + x_2.x_3.x_4.x_7 + x_0.x_2.x_3.x_4.x_7 + x_1.x_2.x_3.x_4.x_7 + x_5.x_7 + x_1.x_5.x_7 + x_0.x_1.x_5.x_7 + x_2.x_5.x_7 + x_3.x_5.x_7 + x_1.x_3.x_5.x_7 + x_0.x_1.x_3.x_5.x_7 + x_0.x_2.x_3.x_5.x_7 + x_0.x_1.x_2.x_3.x_5.x_7 + x_4.x_5.x_7 + x_1.x_4.x_5.x_7 + x_0.x_1.x_4.x_5.x_7 + x_2.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_0.x_1.x_2.x_4.x_5.x_7 + x_3.x_4.x_5.x_7 + x_0.x_3.x_4.x_5.x_7 + x_1.x_6.x_7 + x_2.x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_0.x_2.x_3.x_6.x_7 + x_0.x_4.x_6.x_7 + x_0.x_1.x_4.x_6.x_7 + x_3.x_4.x_6.x_7 + x_1.x_3.x_4.x_6.x_7 + x_0.x_2.x_3.x_4.x_6.x_7 + x_1.x_2.x_3.x_4.x_6.x_7 + x_0.x_1.x_2.x_3.x_4.x_6.x_7,$
- $y_4 = x_0 + x_1 + x_0.x_1 + x_2 + x_3 + x_2.x_3 + x_0.x_2.x_3 + x_0.x_4 + x_1.x_4 + x_1.x_2.x_4 + x_3.x_4 + x_0.x_3.x_4 + x_2.x_3.x_4 + x_0.x_2.x_3.x_4 + x_1.x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_5 + x_0.x_5 + x_1.x_5 + x_2.x_5 + x_3.x_5 + x_0.x_3.x_5 + x_1.x_3.x_5 + x_0.x_1.x_3.x_5 + x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_4.x_5 + x_0.x_4.x_5 + x_1.x_4.x_5 + x_0.x_1.x_4.x_5 + x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_3.x_4.x_5 + x_0.x_3.x_4.x_5 + x_1.x_3.x_4.x_5 + x_0.x_2.x_3.x_4.x_5 + x_0.x_1.x_2.x_3.x_4.x_5 + x_0.x_6 + x_1.x_6 + x_0.x_2.x_6 + x_0.x_1.x_2.x_6 + x_0.x_1.x_3.x_6 + x_2.x_3.x_6 + x_0.x_1.x_2.x_3.x_6 + x_4.x_6 + x_0.x_4.x_6 + x_0.x_1.x_4.x_6 + x_1.x_2.x_4.x_6 + x_3.x_4.x_6 + x_0.x_3.x_4.x_6 + x_0.x_1.x_3.x_4.x_6 + x_0.x_2.x_3.x_4.x_6 + x_1.x_2.x_3.x_4.x_6 + x_0.x_1.$

$x_5.x_6 + x_2.x_5.x_6 + x_3.x_5.x_6 + x_0.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_0.x_2.x_3.x_5.x_6$ $+ x_1.x_2.x_3.x_5.x_6 + x_0.x_1.x_2.x_3.x_5.x_6 + x_0.x_4.x_5.x_6 + x_0.x_1.x_4.x_5.x_6 +$ $x_1.x_2.x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_0.x_1.x_3.x_4.x_5.x_6$ $+ x_2.x_3.x_4.x_5.x_6 + x_1.x_2.x_3.x_4.x_5.x_6 + x_0.x_7 + x_3.x_7 + x_0.x_1.x_3.x_7$ $+ x_0.x_1.x_2.x_3.x_7 + x_0.x_4.x_7 + x_2.x_4.x_7 + x_3.x_4.x_7 + x_0.x_3.x_4.x_7 +$ $x_1.x_3.x_4.x_7 + x_2.x_3.x_4.x_7 + x_0.x_1.x_2.x_3.x_4.x_7 + x_5.x_7 + x_1.x_5.x_7 +$ $x_0.x_1.x_5.x_7 + x_0.x_2.x_5.x_7 + x_1.x_2.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_3.x_5.x_7 +$ $x_0.x_3.x_5.x_7 + x_0.x_1.x_3.x_5.x_7 + x_2.x_3.x_5.x_7 + x_0.x_2.x_3.x_5.x_7 + x_1.x_2.x_3.$ $x_5.x_7 + x_4.x_5.x_7 + x_1.x_2.x_4.x_5.x_7 + x_3.x_4.x_5.x_7 + x_1.x_3.x_4.x_5.x_7 +$ $x_2.x_3.x_4.x_5.x_7 + x_0.x_2.x_3.x_4.x_5.x_7 + x_1.x_2.x_3.x_4.x_5.x_7 + x_0.x_1.x_2.x_3.x_4.$ $x_5.x_7 + x_6.x_7 + x_0.x_6.x_7 + x_1.x_6.x_7 + x_0.x_1.x_6.x_7 + x_2.x_6.x_7 +$ $x_1.x_2.x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_3.x_6.x_7 + x_0.x_3.x_6.x_7 + x_0.x_1.x_3.x_6.x_7$ $+ x_4.x_6.x_7 + x_0.x_4.x_6.x_7 + x_0.x_2.x_4.x_6.x_7 + x_3.x_4.x_6.x_7 + x_0.x_3.x_4.x_6.x_7$ $+ x_2.x_3.x_4.x_6.x_7 + x_1.x_2.x_3.x_4.x_6.x_7 + x_0.x_1.x_2.x_3.x_4.x_6.x_7 + x_5.x_6.x_7$ $+ x_0.x_1.x_5.x_6.x_7 + x_1.x_2.x_5.x_6.x_7 + x_3.x_5.x_6.x_7 + x_0.x_3.x_5.x_6.x_7 +$ $x_1.x_3.x_5.x_6.x_7 + x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_1.x_4.x_5.x_6.x_7 +$ $x_2.x_4.x_5.x_6.x_7 + x_3.x_4.x_5.x_6.x_7 + x_0.x_3.x_4.x_5.x_6.x_7 + x_0.x_2.x_3.x_4.x_5.x_6.x_7,$

- $y_3 = x_0 + x_1.x_2 + x_0.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3$ $+ x_0.x_1.x_2.x_3 + x_4 + x_0.x_1.x_4 + x_0.x_2.x_4 + x_0.x_1.x_2.x_4 + x_0.x_3.x_4 +$ $x_1.x_3.x_4 + x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_0.x_1.x_5 + x_1.x_2.x_5 + x_0.x_1.x_2.x_5$ $+ x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_1.x_2.x_3.x_5 + x_4.x_5 + x_0.x_4.x_5 + x_2.x_4.x_5$ $+ x_1.x_2.x_4.x_5 + x_1.x_3.x_4.x_5 + x_0.x_1.x_3.x_4.x_5 + x_2.x_3.x_4.x_5 + x_0.x_2.x_3.$ $x_4.x_5 + x_6 + x_0.x_1.x_6 + x_0.x_2.x_6 + x_1.x_2.x_6 + x_3.x_6 + x_0.x_3.x_6 +$ $x_0.x_1.x_3.x_6 + x_1.x_2.x_3.x_6 + x_0.x_4.x_6 + x_0.x_1.x_4.x_6 + x_0.x_2.x_4.x_6 +$ $x_0.x_3.x_4.x_6 + x_1.x_3.x_4.x_6 + x_0.x_1.x_3.x_4.x_6 + x_0.x_1.x_2.x_3.x_4.x_6 + x_5.x_6$ $+ x_1.x_5.x_6 + x_0.x_1.x_5.x_6 + x_0.x_2.x_5.x_6 + x_0.x_1.x_2.x_5.x_6 + x_3.x_5.x_6$ $+ x_1.x_3.x_5.x_6 + x_0.x_1.x_3.x_5.x_6 + x_0.x_2.x_3.x_5.x_6 + x_1.x_2.x_3.x_5.x_6 +$ $x_0.x_1.x_2.x_3.x_5.x_6 + x_0.x_4.x_5.x_6 + x_1.x_4.x_5.x_6 + x_2.x_4.x_5.x_6 + x_0.x_2.x_4.$ $x_5.x_6 + x_1.x_2.x_4.x_5.x_6 + x_3.x_4.x_5.x_6 + x_0.x_3.x_4.x_5.x_6 + x_2.x_3.x_4.x_5.x_6$ $+ x_1.x_2.x_3.x_4.x_5.x_6 + x_0.x_1.x_2.x_3.x_4.x_5.x_6 + x_7 + x_0.x_7 + x_1.x_7 +$ $x_0.x_1.x_7 + x_0.x_2.x_7 + x_1.x_2.x_7 + x_3.x_7 + x_0.x_3.x_7 + x_2.x_3.x_7 +$ $x_0.x_2.x_3.x_7 + x_1.x_2.x_3.x_7 + x_0.x_1.x_2.x_3.x_7 + x_0.x_2.x_4.x_7 + x_1.x_2.x_4.x_7$ $+ x_0.x_1.x_2.x_4.x_7 + x_3.x_4.x_7 + x_1.x_3.x_4.x_7 + x_0.x_1.x_3.x_4.x_7 + x_0.x_1.x_2.$ $x_3.x_4.x_7 + x_5.x_7 + x_2.x_5.x_7 + x_1.x_2.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_3.x_5.x_7$ $+ x_0.x_2.x_3.x_5.x_7 + x_0.x_1.x_2.x_3.x_5.x_7 + x_1.x_4.x_5.x_7 + x_0.x_1.x_4.x_5.x_7$ $+ x_2.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_0.x_1.x_2.x_4.x_5.x_7 + x_0.x_3.x_4.x_5.x_7$ $+ x_0.x_1.x_3.x_4.x_5.x_7 + x_2.x_3.x_4.x_5.x_7 + x_1.x_2.x_3.x_4.x_5.x_7 + x_6.x_7 +$ $x_0.x_1.x_6.x_7 + x_0.x_2.x_6.x_7 + x_1.x_2.x_6.x_7 + x_0.x_3.x_6.x_7 + x_0.x_2.x_3.x_6.x_7$ $+ x_0.x_1.x_2.x_3.x_6.x_7 + x_1.x_4.x_6.x_7 + x_2.x_4.x_6.x_7 + x_0.x_1.x_2.x_4.x_6.x_7$ $+ x_3.x_4.x_6.x_7 + x_0.x_3.x_4.x_6.x_7 + x_1.x_3.x_4.x_6.x_7 + x_0.x_2.x_3.x_4.x_6.x_7 +$ $x_1.x_2.x_3.x_4.x_6.x_7 + x_5.x_6.x_7 + x_0.x_5.x_6.x_7 + x_1.x_5.x_6.x_7 + x_0.x_1.x_5.x_6.x_7$ $+ x_0.x_2.x_5.x_6.x_7 + x_1.x_2.x_5.x_6.x_7 + x_3.x_5.x_6.x_7 + x_0.x_3.x_5.x_6.x_7 +$ $x_1.x_3.x_5.x_6.x_7 + x_0.x_2.x_3.x_5.x_6.x_7 + x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_2.x_3.x_5.$ $x_6.x_7 + x_4.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_2.x_4.x_5.x_6.x_7 + x_1.x_2.x_4.x_5.x_6.x_7$

$+ x_3.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7 + x_2.x_3.x_4.x_5.x_6.x_7 + x_0.x_2.x_3.x_4.x_5.x_6.x_7$,

- $y_2 = x_0 + x_1 + x_0.x_2 + x_0.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_0.x_4 + x_1.x_4 + x_0.x_1.x_4 + x_0.x_2.x_4 + x_1.x_2.x_4 + x_3.x_4 + x_0.x_3.x_4 + x_1.x_3.x_4 + x_0.x_1.x_3.x_4 + x_2.x_3.x_4 + x_0.x_2.x_3.x_4 + x_1.x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_5 + x_0.x_5 + x_0.x_1.x_5 + x_0.x_2.x_5 + x_1.x_2.x_5 + x_0.x_1.x_2.x_5 + x_0.x_3.x_5 + x_1.x_3.x_5 + x_0.x_1.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_1.x_2.x_3.x_5 + x_0.x_4.x_5 + x_1.x_4.x_5 + x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_3.x_4.x_5 + x_1.x_2.x_3.x_4.x_5 + x_0.x_1.x_2.x_3.x_4.x_5 + x_0.x_6 + x_0.x_1.x_6 + x_1.x_2.x_6 + x_0.x_3.x_6 + x_0.x_1.x_3.x_6 + x_2.x_3.x_6 + x_1.x_2.x_3.x_6 + x_0.x_1.x_2.x_3.x_6 + x_0.x_4.x_6 + x_0.x_1.x_4.x_6 + x_0.x_2.x_4.x_6 + x_3.x_4.x_6 + x_0.x_3.x_4.x_6 + x_0.x_1.x_3.x_4.x_6 + x_2.x_3.x_4.x_6 + x_1.x_2.x_3.x_4.x_6 + x_0.x_1.x_2.x_3.x_4.x_6 + x_5.x_6 + x_1.x_5.x_6 + x_0.x_1.x_5.x_6 + x_3.x_5.x_6 + x_0.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_0.x_1.x_2.x_3.x_5.x_6 + x_2.x_4.x_5.x_6 + x_0.x_2.x_4.x_5.x_6 + x_1.x_2.x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_3.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_0.x_2.x_3.x_4.x_5.x_6 + x_7 + x_0.x_7 + x_0.x_1.x_7 + x_0.x_2.x_7 + x_1.x_2.x_7 + x_0.x_1.x_2.x_7 + x_0.x_3.x_7 + x_1.x_3.x_7 + x_2.x_3.x_7 + x_0.x_2.x_3.x_7 + x_0.x_1.x_2.x_3.x_7 + x_4.x_7 + x_0.x_4.x_7 + x_1.x_4.x_7 + x_0.x_2.x_4.x_7 + x_1.x_2.x_4.x_7 + x_0.x_1.x_2.x_4.x_7 + x_0.x_3.x_4.x_7 + x_0.x_1.x_3.x_4.x_7 + x_2.x_3.x_4.x_7 + x_0.x_2.x_3.x_4.x_7 + x_0.x_1.x_2.x_3.x_4.x_7 + x_0.x_5.x_7 + x_1.x_5.x_7 + x_1.x_2.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_0.x_3.x_5.x_7 + x_1.x_3.x_5.x_7 + x_0.x_1.x_3.x_5.x_7 + x_0.x_2.x_3.x_5.x_7 + x_1.x_4.x_5.x_7 + x_2.x_4.x_5.x_7 + x_1.x_2.x_4.x_5.x_7 + x_0.x_1.x_2.x_4.x_5.x_7 + x_0.x_3.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_1.x_2.x_3.x_4.x_5.x_7 + x_6.x_7 + x_0.x_6.x_7 + x_2.x_6.x_7 + x_1.x_2.x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_1.x_3.x_6.x_7 + x_0.x_1.x_3.x_6.x_7 + x_1.x_2.x_3.x_6.x_7 + x_4.x_6.x_7 + x_0.x_1.x_4.x_6.x_7 + x_2.x_4.x_6.x_7 + x_1.x_2.x_4.x_6.x_7 + x_1.x_3.x_4.x_6.x_7 + x_0.x_1.x_3.x_4.x_6.x_7 + x_2.x_3.x_4.x_6.x_7 + x_0.x_2.x_3.x_4.x_6.x_7 + x_1.x_2.x_3.x_4.x_6.x_7 + x_0.x_1.x_2.x_3.x_4.x_6.x_7 + x_1.x_5.x_6.x_7 + x_2.x_5.x_6.x_7 + x_0.x_2.x_5.x_6.x_7 + x_3.x_5.x_6.x_7 + x_0.x_3.x_5.x_6.x_7 + x_1.x_3.x_5.x_6.x_7 + x_0.x_1.x_3.x_5.x_6.x_7 + x_2.x_3.x_5.x_6.x_7 + x_0.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_2.x_3.x_5.x_6.x_7 + x_4.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_1.x_4.x_5.x_6.x_7 + x_0.x_1.x_4.x_5.x_6.x_7 + x_0.x_2.x_4.x_5.x_6.x_7 + x_3.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7 + x_2.x_3.x_4.x_5.x_6.x_7 + x_0.x_2.x_3.x_4.x_5.x_6.x_7 + x_1.x_2.x_3.x_4.x_5.x_6.x_7$,
- $y_1 = 1 + x_0 + x_0.x_1 + x_0.x_2 + x_3 + x_0.x_3 + x_1.x_3 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3 + x_0.x_4 + x_1.x_4 + x_0.x_1.x_4 + x_1.x_2.x_4 + x_0.x_3.x_4 + x_1.x_3.x_4 + x_0.x_1.x_3.x_4 + x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_1.x_3.x_5 + x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_4.x_5 + x_0.x_4.x_5 + x_0.x_1.x_4.x_5 + x_0.x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_0.x_3.x_4.x_5 + x_1.x_3.x_4.x_5 + x_2.x_3.x_4.x_5 + x_0.x_2.x_3.x_4.x_5 + x_6 + x_0.x_1.x_6 + x_2.x_6 + x_0.x_1.x_2.x_6 + x_1.x_3.x_6 + x_0.x_1.x_3.x_6 + x_2.x_3.x_6 + x_0.x_2.x_3.x_6 + x_0.x_1.x_2.x_3.x_6 + x_4.x_6 + x_0.x_4.x_6 + x_0.x_1.x_2.x_4.x_6 + x_3.x_4.x_6 + x_0.x_3.x_4.x_6 + x_0.x_1.x_3.x_4.x_6 + x_1.x_2.x_3.x_4.x_6 + x_0.x_1.x_2.x_3.x_4.x_6 + x_1.x_5.x_6 + x_0.x_1.x_5.x_6 + x_0.x_2.x_5.x_6 + x_1.x_2.x_5.x_6 + x_3.x_5.x_6 + x_1.x_3.x_5.x_6 + x_0.x_1.x_3.x_5.x_6 + x_1.x_2.x_3.x_5.x_6 + x_0.x_1.x_2.x_3.x_5.x_6 + x_0.x_4.x_5.x_6 + x_0.x_1.x_4.x_5.x_6 + x_0.x_2.x_4.x_5.x_6 + x_1.x_2.x_4.x_5.x_6 + x_3.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_0.x_1.x_3.x_4.x_5.x_6 + x_0.x_2.x_3.x_4.x_5.x_6 + x_1.x_2.x_3.x_4.x_5.x_6 + x_7 + x_0.x_7 + x_1.x_7 + x_2.x_7$

$+ x_0.x_2.x_7 + x_3.x_7 + x_1.x_3.x_7 + x_0.x_1.x_2.x_3.x_7 + x_0.x_4.x_7 + x_1.x_4.x_7$
$+ x_0.x_1.x_4.x_7 + x_1.x_2.x_4.x_7 + x_0.x_1.x_2.x_4.x_7 + x_3.x_4.x_7 + x_1.x_3.x_4.x_7$
$+ x_0.x_1.x_3.x_4.x_7 + x_2.x_3.x_4.x_7 + x_0.x_2.x_3.x_4.x_7 + x_0.x_1.x_2.x_3.x_4.x_7 +$
$x_0.x_5.x_7 + x_2.x_5.x_7 + x_1.x_2.x_5.x_7 + x_3.x_5.x_7 + x_1.x_3.x_5.x_7 + x_0.x_1.x_3.$
$x_5.x_7 + x_2.x_3.x_5.x_7 + x_0.x_2.x_3.x_5.x_7 + x_0.x_4.x_5.x_7 + x_1.x_4.x_5.x_7 +$
$x_2.x_4.x_5.x_7 + x_0.x_1.x_2.x_4.x_5.x_7 + x_3.x_4.x_5.x_7 + x_0.x_3.x_4.x_5.x_7 + x_1.x_3.$
$x_4.x_5.x_7 + x_2.x_3.x_4.x_5.x_7 + x_0.x_2.x_3.x_4.x_5.x_7 + x_0.x_6.x_7 + x_2.x_6.x_7 +$
$x_1.x_2.x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_3.x_6.x_7 + x_0.x_1.x_3.x_6.x_7 + x_0.x_2.x_3.x_6.x_7$
$+ x_0.x_4.x_6.x_7 + x_1.x_4.x_6.x_7 + x_2.x_4.x_6.x_7 + x_0.x_1.x_2.x_4.x_6.x_7 + x_3.x_4.$
$x_6.x_7 + x_0.x_1.x_3.x_4.x_6.x_7 + x_2.x_3.x_4.x_6.x_7 + x_0.x_2.x_3.x_4.x_6.x_7 + x_1.x_2.$
$x_3.x_4.x_6.x_7 + x_0.x_1.x_2.x_3.x_4.x_6.x_7 + x_5.x_6.x_7 + x_0.x_1.x_5.x_6.x_7 + x_0.x_2.$
$x_5.x_6.x_7 + x_0.x_1.x_2.x_5.x_6.x_7 + x_0.x_3.x_5.x_6.x_7 + x_0.x_2.x_3.x_5.x_6.x_7 +$
$x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_2.x_4.x_5.x_6.x_7$
$+ x_0.x_1.x_2.x_4.x_5.x_6.x_7 + x_1.x_3.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7,$

- $y_0 = 1 + x_0 + x_0.x_1 + x_2 + x_1.x_2 + x_3 + x_1.x_3 + x_2.x_3 + x_1.x_2.x_3$
$+ x_0.x_1.x_2.x_3 + x_4 + x_0.x_4 + x_1.x_4 + x_0.x_1.x_4 + x_2.x_4 + x_0.x_2.x_4$
$+ x_1.x_2.x_4 + x_0.x_3.x_4 + x_1.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_0.x_5 + x_0.x_2.x_5$
$+ x_0.x_1.x_2.x_5 + x_0.x_3.x_5 + x_2.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 +$
$x_0.x_1.x_4.x_5 + x_0.x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_0.x_2.x_3.x_4.x_5 + x_0.x_6$
$+ x_1.x_6 + x_0.x_1.x_6 + x_2.x_6 + x_0.x_2.x_6 + x_1.x_2.x_6 + x_0.x_1.x_2.x_6 +$
$x_0.x_3.x_6 + x_1.x_2.x_3.x_6 + x_0.x_1.x_2.x_3.x_6 + x_4.x_6 + x_0.x_4.x_6 + x_1.x_4.x_6 +$
$x_1.x_2.x_4.x_6 + x_0.x_3.x_4.x_6 + x_1.x_3.x_4.x_6 + x_0.x_1.x_3.x_4.x_6 + x_0.x_2.x_3.x_4.x_6$
$+ x_0.x_1.x_2.x_3.x_4.x_6 + x_5.x_6 + x_1.x_5.x_6 + x_2.x_5.x_6 + x_0.x_2.x_5.x_6 +$
$x_0.x_3.x_5.x_6 + x_0.x_1.x_3.x_5.x_6 + x_1.x_2.x_3.x_5.x_6 + x_4.x_5.x_6 + x_0.x_4.x_5.x_6$
$+ x_1.x_4.x_5.x_6 + x_0.x_1.x_4.x_5.x_6 + x_2.x_4.x_5.x_6 + x_1.x_2.x_4.x_5.x_6 + x_0.x_3.$
$x_4.x_5.x_6 + x_2.x_3.x_4.x_5.x_6 + x_0.x_2.x_3.x_4.x_5.x_6 + x_0.x_1.x_7 + x_2.x_7 +$
$x_0.x_2.x_7 + x_0.x_1.x_2.x_7 + x_1.x_3.x_7 + x_2.x_3.x_7 + x_0.x_2.x_3.x_7 + x_1.x_2.x_3.x_7$
$+ x_0.x_1.x_2.x_3.x_7 + x_0.x_4.x_7 + x_0.x_1.x_4.x_7 + x_2.x_4.x_7 + x_0.x_2.x_4.x_7 +$
$x_1.x_2.x_4.x_7 + x_0.x_1.x_2.x_4.x_7 + x_3.x_4.x_7 + x_0.x_1.x_2.x_3.x_4.x_7 + x_5.x_7 +$
$x_2.x_5.x_7 + x_0.x_2.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_3.x_5.x_7 + x_0.x_1.x_3.x_5.x_7$
$+ x_2.x_3.x_5.x_7 + x_1.x_2.x_3.x_5.x_7 + x_0.x_1.x_2.x_3.x_5.x_7 + x_0.x_4.x_5.x_7 +$
$x_1.x_4.x_5.x_7 + x_2.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_0.x_1.x_2.x_4.x_5.x_7 + x_0.x_3.$
$x_4.x_5.x_7 + x_1.x_3.x_4.x_5.x_7 + x_2.x_3.x_4.x_5.x_7 + x_0.x_2.x_3.x_4.x_5.x_7 + x_6.x_7$
$+ x_2.x_6.x_7 + x_1.x_2.x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_3.x_6.x_7 + x_1.x_3.x_6.x_7$
$+ x_0.x_1.x_3.x_6.x_7 + x_2.x_3.x_6.x_7 + x_0.x_1.x_2.x_3.x_6.x_7 + x_0.x_4.x_6.x_7 +$
$x_0.x_2.x_4.x_6.x_7 + x_1.x_2.x_4.x_6.x_7 + x_0.x_3.x_4.x_6.x_7 + x_0.x_1.x_3.x_4.x_6.x_7$
$+ x_1.x_2.x_3.x_4.x_6.x_7 + x_0.x_1.x_2.x_3.x_4.x_6.x_7 + x_5.x_6.x_7 + x_1.x_5.x_6.x_7$
$+ x_0.x_1.x_5.x_6.x_7 + x_1.x_2.x_5.x_6.x_7 + x_0.x_1.x_2.x_5.x_6.x_7 + x_3.x_5.x_6.x_7 +$
$x_0.x_3.x_5.x_6.x_7 + x_0.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_4.x_5.x_6.x_7 + x_2.x_4.x_5.x_6.x_7$
$+ x_1.x_2.x_4.x_5.x_6.x_7 + x_0.x_1.x_2.x_4.x_5.x_6.x_7 + x_1.x_3.x_4.x_5.x_6.x_7 + x_2.x_3.$
$x_4.x_5.x_6.x_7 + x_0.x_2.x_3.x_4.x_5.x_6.x_7.$

For the ANFs of the SKIPJACK S-box [69], the input is denoted (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) and the output is (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7):

- $y_7 = 1 + x_1 + x_0.x_1 + x_0.x_2 + x_1.x_2 + x_0.x_3 + x_0.x_1.x_2.x_3 + x_0.x_4 + x_0.x_1.x_4 + x_0.x_2.x_4 + x_3.x_4 + x_0.x_3.x_4 + x_0.x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_5 + x_0.x_5 + x_1.x_5 + x_0.x_1.x_5 + x_0.x_2.x_5 + x_1.x_2.x_5 + x_0.x_1.x_2.x_5 + x_0.x_3.x_5 + x_2.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_4.x_5 + x_1.x_4.x_5 + x_1.x_2.x_4.x_5 + x_0.x_2.x_3.x_4.x_5 + x_6 + x_0.x_6 + x_1.x_6 + x_2.x_6 + x_1.x_2.x_6 + x_3.x_6 + x_1.x_3.x_6 + x_2.x_3.x_6 + x_0.x_2.x_3.x_6 + x_0.x_1.x_2.x_3.x_6 + x_0.x_4.x_6 + x_1.x_4.x_6 + x_0.x_1.x_4.x_6 + x_2.x_4.x_6 + x_0.x_2.x_4.x_6 + x_0.x_1.x_2.x_4.x_6 + x_0.x_3.x_4.x_6 + x_1.x_3.x_4.x_6 + x_2.x_3.x_4.x_6 + x_0.x_2.x_3.x_4.x_6 + x_0.x_1.x_2.x_3.x_4.x_6 + x_5.x_6 + x_0.x_5.x_6 + x_0.x_1.x_5.x_6 + x_2.x_5.x_6 + x_0.x_2.x_5.x_6 + x_3.x_5.x_6 + x_0.x_3.x_5.x_6 + x_1.x_3.x_5.x_6 + x_0.x_4.x_5.x_6 + x_1.x_4.x_5.x_6 + x_0.x_1.x_4.x_5.x_6 + x_2.x_4.x_5.x_6 + x_0.x_2.x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_3.x_4.x_5.x_6 + x_0.x_3.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_2.x_3.x_4.x_5.x_6 + x_7 + x_0.x_7 + x_0.x_1.x_7 + x_1.x_2.x_7 + x_0.x_1.x_2.x_7 + x_0.x_3.x_7 + x_1.x_3.x_7 + x_2.x_3.x_7 + x_1.x_2.x_3.x_7 + x_4.x_7 + x_0.x_1.x_4.x_7 + x_1.x_2.x_4.x_7 + x_0.x_1.x_2.x_4.x_7 + x_3.x_4.x_7 + x_0.x_1.x_3.x_4.x_7 + x_5.x_7 + x_0.x_5.x_7 + x_2.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_0.x_3.x_5.x_7 + x_1.x_3.x_5.x_7 + x_2.x_3.x_5.x_7 + x_1.x_2.x_3.x_5.x_7 + x_0.x_1.x_2.x_3.x_5.x_7 + x_1.x_4.x_5.x_7 + x_0.x_1.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_1.x_2.x_4.x_5.x_7 + x_0.x_1.x_2.x_4.x_5.x_7 + x_3.x_4.x_5.x_7 + x_0.x_3.x_4.x_5.x_7 + x_1.x_3.x_4.x_5.x_7 + x_1.x_2.x_3.x_4.x_5.x_7 + x_0.x_1.x_2.x_3.x_4.x_5.x_7 + x_0.x_6.x_7 + x_1.x_6.x_7 + x_0.x_3.x_6.x_7 + x_1.x_3.x_6.x_7 + x_1.x_2.x_3.x_6.x_7 + x_1.x_4.x_6.x_7 + x_0.x_1.x_4.x_6.x_7 + x_1.x_2.x_4.x_6.x_7 + x_1.x_3.x_4.x_6.x_7 + x_2.x_3.x_4.x_6.x_7 + x_0.x_1.x_2.x_3.x_4.x_6.x_7 + x_0.x_1.x_5.x_6.x_7 + x_1.x_2.x_5.x_6.x_7 + x_0.x_1.x_3.x_5.x_6.x_7 + x_0.x_2.x_3.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_2.x_4.x_5.x_6.x_7 + x_0.x_2.x_4.x_5.x_6.x_7 + x_1.x_2.x_4.x_5.x_6.x_7 + x_3.x_4.x_5.x_6.x_7 + x_1.x_3.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7,$
- $y_6 = x_0 + x_0.x_1 + x_2 + x_0.x_2 + x_0.x_1.x_2 + x_0.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_0.x_1.x_2.x_3 + x_4 + x_0.x_1.x_4 + x_2.x_4 + x_0.x_3.x_4 + x_2.x_3.x_4 + x_2.x_5 + x_0.x_2.x_5 + x_1.x_2.x_5 + x_0.x_1.x_2.x_5 + x_1.x_3.x_5 + x_0.x_1.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_4.x_5 + x_0.x_4.x_5 + x_1.x_4.x_5 + x_0.x_1.x_4.x_5 + x_1.x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_3.x_4.x_5 + x_0.x_3.x_4.x_5 + x_0.x_1.x_3.x_4.x_5 + x_2.x_3.x_4.x_5 + x_1.x_2.x_3.x_4.x_5 + x_0.x_1.x_2.x_3.x_4.x_5 + x_0.x_6 + x_1.x_6 + x_0.x_1.x_2.x_6 + x_3.x_6 + x_0.x_1.x_3.x_6 + x_0.x_4.x_6 + x_1.x_4.x_6 + x_2.x_4.x_6 + x_0.x_2.x_4.x_6 + x_3.x_4.x_6 + x_0.x_3.x_4.x_6 + x_1.x_3.x_4.x_6 + x_2.x_3.x_4.x_6 + x_1.x_2.x_3.x_4.x_6 + x_0.x_5.x_6 + x_0.x_1.x_5.x_6 + x_0.x_2.x_5.x_6 + x_1.x_2.x_5.x_6 + x_0.x_1.x_2.x_5.x_6 + x_0.x_3.x_5.x_6 + x_1.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_0.x_2.x_3.x_5.x_6 + x_1.x_2.x_3.x_5.x_6 + x_0.x_1.x_2.x_3.x_5.x_6 + x_1.x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_0.x_1.x_3.x_4.x_5.x_6 + x_2.x_3.x_4.x_5.x_6 + x_0.x_2.x_3.x_4.x_5.x_6 + x_1.x_2.x_3.x_4.x_5.x_6 + x_7 + x_0.x_7 + x_1.x_7 + x_2.x_7 + x_1.x_2.x_7 + x_0.x_1.x_2.x_7 + x_3.x_7 + x_1.x_3.x_7 + x_0.x_1.x_3.x_7 + x_0.x_2.x_3.x_7 + x_1.x_2.x_3.x_7 + x_0.x_4.x_7 + x_1.x_4.x_7 + x_0.x_1.x_4.x_7 + x_2.x_4.x_7 + x_0.x_2.x_4.x_7 + x_1.x_2.x_4.x_7 + x_0.x_3.x_4.x_7 + x_1.x_3.x_4.x_7 + x_0.x_1.x_2.x_3.x_4.x_7 + x_0.x_5.x_7 + x_1.x_5.x_7 + x_0.x_1.x_5.x_7 + x_0.x_2.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_3.x_5.x_7 + x_0.x_3.x_5.x_7 + x_0.x_1.x_3.x_5.x_7 + x_2.x_3.x_5.x_7 + x_4.x_5.x_7 + x_0.x_1.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_1.x_3.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_2.x_3.x_4.x_5.x_7 + x_0.x_2.x_3.x_4.x_5.x_7 + x_1.x_2.x_3.x_4.x_5.x_7 + x_0.x_1.x_2.x_3.x_4.x_5.x_7 + x_6.x_7 + x_0.x_6.x_7 + x_0.x_1.x_6.$

$x_7 + x_1.x_2.\ x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_0.x_1.x_3.x_6.x_7 + x_0.x_2.x_3.x_6.x_7 + x_1.x_2.x_3.x_6.x_7 + x_4.x_6.x_7 + x_0.x_4.x_6.x_7 + x_0.x_3.x_4.x_6.x_7 + x_2.x_3.x_4.x_6.x_7 + x_1.x_2.x_3.x_4.x_6.x_7 + x_0.x_5.x_6.x_7 + x_1.x_5.x_6.x_7 + x_0.x_1.x_5.x_6.x_7 + x_2.x_5.x_6.x_7 + x_3.x_5.x_6.x_7 + x_1.x_3.x_5.x_6.x_7 + x_0.x_1.x_3.x_5.x_6.x_7 + x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_2.x_3.x_5.x_6.x_7 + x_4.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_1.x_4.x_5.x_6.x_7 + x_0.x_1.x_4.x_5.x_6.x_7 + x_2.x_4.x_5.x_6.x_7 + x_0.x_1.x_2.x_4.x_5.x_6.x_7 + x_3.x_4.x_5.x_6.x_7 + x_1.x_3.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7 + x_2.x_3.x_4.x_5.x_6.x_7 + x_0.x_2.x_3.x_4.x_5.x_6.x_7,$

- $y_5 = 1 + x_0 + x_1 + x_0.x_1 + x_1.x_2 + x_0.x_3 + x_2.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3 + x_0.x_4 + x_0.x_1.x_4 + x_2.x_4 + x_0.x_1.x_2.x_4 + x_3.x_4 + x_0.x_3.x_4 + x_1.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_5 + x_0.x_5 + x_1.x_5 + x_0.x_2.x_5 + x_1.x_3.x_5 + x_0.x_1.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_1.x_2.x_3.x_5 + x_0.x_4.x_5 + x_1.x_4.x_5 + x_0.x_2.x_4.x_5 + x_1.x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_1.x_3.x_4.x_5 + x_0.x_1.x_3.x_4.x_5 + x_0.x_2.x_3.x_4.x_5 + x_0.x_1.x_2.x_3.x_4.x_5 + x_0.x_6 + x_1.x_6 + x_2.x_6 + x_1.x_2.x_6 + x_0.x_1.x_2.x_6 + x_0.x_3.x_6 + x_1.x_3.x_6 + x_2.x_3.x_6 + x_0.x_2.x_3.x_6 + x_1.x_2.x_3.x_6 + x_4.x_6 + x_0.x_1.x_4.x_6 + x_2.x_4.x_6 + x_3.x_4.x_6 + x_0.x_1.x_3.x_4.x_6 + x_0.x_2.x_3.x_4.x_6 + x_1.x_2.x_3.x_4.x_6 + x_5.x_6 + x_0.x_1.x_5.x_6 + x_2.x_5.x_6 + x_0.x_2.x_5.x_6 + x_1.x_2.x_5.x_6 + x_3.x_5.x_6 + x_0.x_3.x_5.x_6 + x_1.x_3.x_5.x_6 + x_0.x_1.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_1.x_2.x_3.x_5.x_6 + x_1.x_4.x_5.x_6 + x_2.x_4.x_5.x_6 + x_0.x_3.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_0.x_1.x_3.x_4.x_5.x_6 + x_2.x_3.x_4.x_5.x_6 + x_0.x_1.x_2.x_3.x_4.x_5.x_6 + x_7 + x_1.x_7 + x_0.x_1.x_7 + x_1.x_2.x_7 + x_0.x_1.x_2.x_7 + x_3.x_7 + x_0.x_3.x_7 + x_1.x_3.x_7 + x_0.x_1.x_3.x_7 + x_1.x_2.x_3.x_7 + x_0.x_1.x_2.x_3.x_7 + x_1.x_4.x_7 + x_2.x_4.x_7 + x_1.x_2.x_4.x_7 + x_1.x_3.x_4.x_7 + x_0.x_1.x_3.x_4.x_7 + x_2.x_3.x_4.x_7 + x_0.x_1.x_2.x_3.x_4.x_7 + x_0.x_5.x_7 + x_1.x_5.x_7 + x_0.x_1.x_5.x_7 + x_2.x_5.x_7 + x_1.x_2.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_0.x_3.x_5.x_7 + x_2.x_3.x_5.x_7 + x_0.x_2.x_3.x_5.x_7 + x_1.x_2.x_3.x_5.x_7 + x_0.x_1.x_2.x_3.x_5.x_7 + x_1.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_1.x_2.x_4.x_5.x_7 + x_3.x_4.x_5.x_7 + x_1.x_3.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_2.x_3.x_4.x_5.x_7 + x_6.x_7 + x_0.x_6.x_7 + x_1.x_6.x_7 + x_0.x_2.x_6.x_7 + x_3.x_6.x_7 + x_0.x_1.x_3.x_6.x_7 + x_2.x_3.x_6.x_7 + x_1.x_2.x_3.x_6.x_7 + x_0.x_1.x_2.x_3.x_6.x_7 + x_0.x_4.x_6.x_7 + x_0.x_1.x_4.x_6.x_7 + x_2.x_4.x_6.x_7 + x_1.x_2.x_4.x_6.x_7 + x_3.x_4.x_6.x_7 + x_0.x_1.x_3.x_4.x_6.x_7 + x_0.x_2.x_3.x_4.x_6.x_7 + x_1.x_2.x_3.x_4.x_6.x_7 + x_5.x_6.x_7 + x_2.x_5.x_6.x_7 + x_1.x_3.x_5.x_6.x_7 + x_2.x_3.x_5.x_6.x_7 + x_0.x_2.x_3.x_5.x_6.x_7 + x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_2.x_3.x_5.x_6.x_7 + x_4.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_1.x_4.x_5.x_6.x_7 + x_0.x_1.x_4.x_5.x_6.x_7 + x_2.x_4.x_5.x_6.x_7 + x_0.x_2.x_4.x_5.x_6.x_7 + x_1.x_2.x_4.x_5.x_6.x_7 + x_1.x_3.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7 + x_1.x_2.x_3.x_4.x_5.x_6.x_7,$
- $y_4 = x_0 + x_0.x_1 + x_2 + x_3 + x_2.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3 + x_0.x_4 + x_0.x_1.x_4 + x_2.x_4 + x_1.x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_0.x_2.x_5 + x_1.x_2.x_5 + x_0.x_1.x_2.x_5 + x_0.x_3.x_5 + x_2.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_1.x_2.x_3.x_5 + x_4.x_5 + x_0.x_4.x_5 + x_1.x_4.x_5 + x_0.x_2.x_4.x_5 + x_3.x_4.x_5 + x_0.x_3.x_4.x_5 + x_0.x_2.x_3.x_4.x_5 + x_0.x_1.x_2.x_3.x_4.x_5 + x_6 + x_0.x_6 + x_0.x_2.x_6 + x_0.x_1.x_2.x_6 + x_0.x_3.x_6 + x_1.x_3.x_6 + x_0.x_1.x_3.x_6 + x_2.x_3.x_6 + x_0.x_2.x_3.x_6 + x_1.x_2.x_3.x_6 + x_0.x_4.x_6 + x_0.x_1.x_4.x_6 + x_3.x_4.x_6 + x_0.x_3.x_4.x_6 + x_0.x_1.x_3.x_4.x_6 + x_2.x_3.x_4.x_6 + x_0.x_2.x_3.x_4.x_6 + x_1.x_2.x_3.x_4.x_6 + x_0.x_1.x_2.x_3.x_4.x_6 + x_1.x_5.x_6 + x_0.x_1.x_5.x_6 + x_2.x_5.x_6 + x_0.x_2.$

$x_5.x_6 + x_1.x_2.x_5.x_6 + x_0.x_1.x_2.x_5.x_6 + x_1.x_3.x_5.x_6 + x_0.x_1.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_0.x_2.x_3.x_5.x_6 + x_1.x_2.x_3.x_5.x_6 + x_0.x_1.x_2.x_3.x_5.x_6 + x_4.x_5.x_6 + x_0.x_4.x_5.x_6 + x_0.x_1.x_4.x_5.x_6 + x_2.x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_0.x_1.x_3.x_4.x_5.x_6 + x_0.x_2.x_3.x_4.x_5.x_6 + x_0.x_7 + x_1.x_7 + x_2.x_7 + x_0.x_2.x_7 + x_1.x_2.x_7 + x_3.x_7 + x_0.x_3.x_7 + x_1.x_3.x_7 + x_0.x_1.x_3.x_7 + x_1.x_2.x_3.x_7 + x_0.x_4.x_7 + x_2.x_4.x_7 + x_0.x_2.x_4.x_7 + x_1.x_3.x_4.x_7 + x_0.x_1.x_3.x_4.x_7 + x_1.x_2.x_3.x_4.x_7 + x_5.x_7 + x_0.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_2.x_3.x_5.x_7 + x_1.x_2.x_3.x_5.x_7 + x_4.x_5.x_7 + x_0.x_4.x_5.x_7 + x_0.x_1.x_4.x_5.x_7 + x_0.x_3.x_4.x_5.x_7 + x_1.x_3.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_2.x_3.x_4.x_5.x_7 + x_0.x_2.x_3.x_4.x_5.x_7 + x_0.x_1.x_2.x_3.x_4.x_5.x_7 + x_6.x_7 + x_0.x_6.x_7 + x_1.x_6.x_7 + x_2.x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_0.x_3.x_6.x_7 + x_1.x_3.x_6.x_7 + x_0.x_4.x_6.x_7 + x_1.x_4.x_6.x_7 + x_0.x_1.x_4.x_6.x_7 + x_3.x_4.x_6.x_7 + x_0.x_3.x_4.x_6.x_7 + x_1.x_3.x_4.x_6.x_7 + x_0.x_1.x_3.x_4.x_6.x_7 + x_0.x_2.x_3.x_4.x_6.x_7 + x_5.x_6.x_7 + x_0.x_1.x_5.x_6.x_7 + x_2.x_5.x_6.x_7 + x_1.x_2.x_5.x_6.x_7 + x_0.x_1.x_2.x_5.x_6.x_7 + x_3.x_5.x_6.x_7 + x_0.x_3.x_5.x_6.x_7 + x_1.x_3.x_5.x_6.x_7 + x_0.x_1.x_3.x_5.x_6.x_7 + x_0.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_1.x_2.x_4.x_5.x_6.x_7 + x_0.x_1.x_2.x_4.x_5.x_6.x_7 + x_0.x_3.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7,$

- $y_3 = x_1 + x_0.x_1 + x_2 + x_0.x_1.x_2 + x_1.x_3 + x_0.x_2.x_3 + x_0.x_1.x_2.x_3 + x_0.x_4 + x_0.x_2.x_4 + x_1.x_2.x_4 + x_0.x_3.x_4 + x_1.x_3.x_4 + x_0.x_1.x_3.x_4 + x_1.x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 + x_5 + x_0.x_1.x_5 + x_1.x_2.x_5 + x_3.x_5 + x_1.x_3.x_5 + x_0.x_1.x_3.x_5 + x_0.x_2.x_3.x_5 + x_4.x_5 + x_0.x_4.x_5 + x_2.x_4.x_5 + x_0.x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_0.x_3.x_4.x_5 + x_1.x_3.x_4.x_5 + x_2.x_3.x_4.x_5 + x_1.x_2.x_3.x_4.x_5 + x_0.x_1.x_2.x_3.x_4.x_5 + x_6 + x_0.x_6 + x_1.x_6 + x_0.x_2.x_6 + x_0.x_1.x_2.x_6 + x_1.x_3.x_6 + x_0.x_2.x_3.x_6 + x_1.x_2.x_3.x_6 + x_4.x_6 + x_0.x_4.x_6 + x_1.x_4.x_6 + x_0.x_1.x_4.x_6 + x_3.x_4.x_6 + x_1.x_3.x_4.x_6 + x_0.x_2.x_3.x_4.x_6 + x_1.x_2.x_3.x_4.x_6 + x_0.x_1.x_2.x_3.x_4.x_6 + x_0.x_5.x_6 + x_2.x_5.x_6 + x_0.x_2.x_5.x_6 + x_1.x_2.x_5.x_6 + x_0.x_1.x_2.x_5.x_6 + x_3.x_5.x_6 + x_1.x_3.x_5.x_6 + x_0.x_1.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_0.x_2.x_3.x_5.x_6 + x_1.x_2.x_3.x_5.x_6 + x_0.x_1.x_2.x_3.x_5.x_6 + x_1.x_4.x_5.x_6 + x_2.x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_3.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_0.x_1.x_3.x_4.x_5.x_6 + x_2.x_3.x_4.x_5.x_6 + x_0.x_2.x_3.x_4.x_5.x_6 + x_0.x_7 + x_0.x_2.x_7 + x_1.x_2.x_7 + x_0.x_1.x_2.x_7 + x_0.x_3.x_7 + x_0.x_1.x_3.x_7 + x_0.x_2.x_3.x_7 + x_0.x_1.x_2.x_3.x_7 + x_4.x_7 + x_2.x_4.x_7 + x_0.x_2.x_4.x_7 + x_1.x_2.x_4.x_7 + x_0.x_3.x_4.x_7 + x_1.x_3.x_4.x_7 + x_1.x_2.x_3.x_4.x_7 + x_5.x_7 + x_1.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_3.x_5.x_7 + x_2.x_3.x_5.x_7 + x_0.x_2.x_3.x_5.x_7 + x_1.x_4.x_5.x_7 + x_2.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_3.x_4.x_5.x_7 + x_0.x_3.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_2.x_3.x_4.x_5.x_7 + x_0.x_6.x_7 + x_1.x_6.x_7 + x_0.x_1.x_6.x_7 + x_0.x_2.x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_2.x_3.x_6.x_7 + x_1.x_2.x_3.x_6.x_7 + x_0.x_4.x_6.x_7 + x_2.x_4.x_6.x_7 + x_1.x_2.x_4.x_6.x_7 + x_3.x_4.x_6.x_7 + x_1.x_3.x_4.x_6.x_7 + x_0.x_1.x_3.x_4.x_6.x_7 + x_0.x_2.x_3.x_4.x_6.x_7 + x_0.x_5.x_6.x_7 + x_1.x_5.x_6.x_7 + x_0.x_1.x_5.x_6.x_7 + x_2.x_5.x_6.x_7 + x_0.x_1.x_2.x_5.x_6.x_7 + x_0.x_1.x_3.x_5.x_6.x_7 + x_4.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_2.x_4.x_5.x_6.x_7 + x_3.x_4.x_5.x_6.x_7 + x_0.x_3.x_4.x_5.x_6.x_7,$
- $y_2 = x_0 + x_0.x_1 + x_0.x_2 + x_1.x_2 + x_0.x_1.x_2 + x_0.x_3 + x_1.x_3 + x_0.x_2.x_3 + x_1.x_2.x_3 + x_0.x_1.x_2.x_3 + x_4 + x_0.x_4 + x_0.x_2.x_4 + x_0.x_1.x_2.x_4 + x_3.x_4 + x_1.x_3.x_4 + x_0.x_1.x_3.x_4 + x_2.x_3.x_4 + x_0.x_2.x_3.x_4 + x_0.x_1.x_2.x_3.x_4 +$

$x_2.x_5 + x_0.x_2.x_5 + x_0.x_1.x_2.x_5 + x_0.x_3.x_5 + x_1.x_3.x_5 + x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_1.x_2.x_3.x_5 + x_1.x_4.x_5 + x_0.x_1.x_4.x_5 + x_0.x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_3.x_4.x_5 + x_0.x_1.x_3.x_4.x_5 + x_2.x_3.x_4.x_5 + x_0.x_1.x_2.x_3.x_4.x_5 + x_0.x_1.x_6 + x_1.x_2.x_6 + x_0.x_1.x_2.x_6 + x_3.x_6 + x_2.x_3.x_6 + x_1.x_2.x_3.x_6 + x_0.x_4.x_6 + x_1.x_4.x_6 + x_1.x_2.x_4.x_6 + x_0.x_1.x_2.x_4.x_6 + x_3.x_4.x_6 + x_1.x_3.x_4.x_6 + x_2.x_3.x_4.x_6 + x_0.x_2.x_3.x_4.x_6 + x_5.x_6 + x_0.x_5.x_6 + x_1.x_5.x_6 + x_0.x_1.x_5.x_6 + x_0.x_2.x_5.x_6 + x_3.x_5.x_6 + x_0.x_3.x_5.x_6 + x_1.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_0.x_3.x_4.x_5.x_6 + x_1.x_3.x_4.x_5.x_6 + x_1.x_7 + x_0.x_2.x_7 + x_1.x_2.x_7 + x_3.x_7 + x_0.x_3.x_7 + x_1.x_2.x_3.x_7 + x_4.x_7 + x_1.x_4.x_7 + x_2.x_4.x_7 + x_0.x_2.x_4.x_7 + x_0.x_3.x_4.x_7 + x_1.x_3.x_4.x_7 + x_2.x_3.x_4.x_7 + x_0.x_2.x_3.x_4.x_7 + x_0.x_1.x_2.x_3.x_4.x_7 + x_0.x_5.x_7 + x_1.x_5.x_7 + x_0.x_2.x_5.x_7 + x_1.x_2.x_5.x_7 + x_3.x_5.x_7 + x_0.x_3.x_5.x_7 + x_1.x_3.x_5.x_7 + x_0.x_1.x_3.x_5.x_7 + x_2.x_3.x_5.x_7 + x_0.x_2.x_3.x_5.x_7 + x_4.x_5.x_7 + x_2.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_0.x_3.x_4.x_5.x_7 + x_1.x_3.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_0.x_2.x_3.x_4.x_5.x_7 + x_1.x_2.x_3.x_4.x_5.x_7 + x_6.x_7 + x_0.x_6.x_7 + x_0.x_1.x_6.x_7 + x_0.x_2.x_6.x_7 + x_1.x_2.x_6.x_7 + x_3.x_6.x_7 + x_1.x_2.x_3.x_6.x_7 + x_0.x_1.x_2.x_3.x_6.x_7 + x_1.x_4.x_6.x_7 + x_2.x_4.x_6.x_7 + x_0.x_2.x_4.x_6.x_7 + x_1.x_2.x_4.x_6.x_7 + x_3.x_4.x_6.x_7 + x_2.x_3.x_4.x_6.x_7 + x_0.x_1.x_2.x_3.x_4.x_6.x_7 + x_0.x_5.x_6.x_7 + x_1.x_5.x_6.x_7 + x_2.x_5.x_6.x_7 + x_3.x_5.x_6.x_7 + x_0.x_3.x_5.x_6.x_7 + x_1.x_3.x_5.x_6.x_7 + x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_2.x_3.x_5.x_6.x_7 + x_4.x_5.x_6.x_7 + x_0.x_4.x_5.x_6.x_7 + x_1.x_4.x_5.x_6.x_7 + x_0.x_1.x_4.x_5.x_6.x_7 + x_2.x_4.x_5.x_6.x_7 + x_0.x_2.x_4.x_5.x_6.x_7 + x_0.x_1.x_3.x_4.x_5.x_6.x_7 + x_2.x_3.x_4.x_5.x_6.x_7,$

- $y_1 = 1 + x_1 + x_0.x_1 + x_2 + x_0.x_3 + x_0.x_2.x_3 + x_0.x_4 + x_0.x_1.x_4 + x_2.x_4 + x_1.x_2.x_4 + x_0.x_1.x_2.x_4 + x_0.x_3.x_4 + x_0.x_2.x_3.x_4 + x_1.x_2.x_3.x_4 + x_1.x_5 + x_2.x_5 + x_0.x_2.x_5 + x_1.x_2.x_5 + x_0.x_3.x_5 + x_0.x_1.x_3.x_5 + x_2.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_1.x_4.x_5 + x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_3.x_4.x_5 + x_0.x_3.x_4.x_5 + x_0.x_1.x_3.x_4.x_5 + x_0.x_2.x_3.x_4.x_5 + x_0.x_1.x_2.x_3.x_4.x_5 + x_6 + x_0.x_6 + x_2.x_6 + x_1.x_2.x_6 + x_0.x_1.x_2.x_6 + x_3.x_6 + x_0.x_3.x_6 + x_1.x_3.x_6 + x_2.x_3.x_6 + x_0.x_1.x_2.x_3.x_6 + x_0.x_1.x_4.x_6 + x_0.x_2.x_4.x_6 + x_1.x_2.x_4.x_6 + x_0.x_1.x_2.x_4.x_6 + x_1.x_3.x_4.x_6 + x_0.x_1.x_3.x_4.x_6 + x_0.x_5.x_6 + x_0.x_2.x_5.x_6 + x_1.x_2.x_5.x_6 + x_3.x_5.x_6 + x_0.x_3.x_5.x_6 + x_1.x_3.x_5.x_6 + x_0.x_1.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_1.x_2.x_3.x_5.x_6 + x_0.x_1.x_2.x_3.x_5.x_6 + x_4.x_5.x_6 + x_0.x_1.x_4.x_5.x_6 + x_0.x_2.x_4.x_5.x_6 + x_0.x_1.x_2.x_4.x_5.x_6 + x_0.x_1.x_3.x_4.x_5.x_6 + x_0.x_7 + x_1.x_7 + x_0.x_2.x_7 + x_2.x_3.x_7 + x_4.x_7 + x_0.x_4.x_7 + x_0.x_2.x_4.x_7 + x_0.x_3.x_4.x_7 + x_2.x_3.x_4.x_7 + x_0.x_1.x_2.x_3.x_4.x_7 + x_5.x_7 + x_0.x_5.x_7 + x_1.x_5.x_7 + x_0.x_1.x_5.x_7 + x_2.x_5.x_7 + x_0.x_2.x_5.x_7 + x_1.x_2.x_5.x_7 + x_0.x_1.x_2.x_5.x_7 + x_0.x_3.x_5.x_7 + x_0.x_2.x_3.x_5.x_7 + x_1.x_2.x_3.x_5.x_7 + x_4.x_5.x_7 + x_0.x_4.x_5.x_7 + x_2.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_1.x_2.x_4.x_5.x_7 + x_3.x_4.x_5.x_7 + x_1.x_3.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_0.x_2.x_3.x_4.x_5.x_7 + x_1.x_2.x_3.x_4.x_5.x_7 + x_1.x_6.x_7 + x_0.x_2.x_6.x_7 + x_1.x_2.x_6.x_7 + x_0.x_1.x_2.x_6.x_7 + x_3.x_6.x_7 + x_2.x_3.x_6.x_7 + x_0.x_1.x_2.x_3.x_6.x_7 + x_4.x_6.x_7 + x_0.x_4.x_6.x_7 + x_1.x_4.x_6.x_7 + x_0.x_1.x_4.x_6.x_7 + x_2.x_4.x_6.x_7 + x_1.x_2.x_4.x_6.x_7 + x_1.x_3.x_4.x_6.x_7 + x_2.x_3.x_4.x_6.x_7 + x_0.x_2.x_3.x_4.x_6.x_7 + x_5.x_6.x_7 + x_0.x_5.x_6.x_7 + x_1.x_5.x_6.x_7 + x_2.x_5.x_6.x_7 + x_0.x_1.x_2.x_5.x_6.x_7 + x_1.x_3.x_5.x_6.x_7 + x_0.x_1.x_3.x_5.x_6.x_7 + x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_2.x_3.x_5.x_6.x_7 +$

$x_4.x_5.x_6.x_7 + x_0.x_1.x_4.x_5.x_6.x_7 + x_2.x_4.x_5.x_6.x_7 + x_0.x_2.x_4.x_5.x_6.x_7 + x_1.x_2.x_4.x_5.x_6.x_7 + x_0.x_3.x_4.x_5.x_6.x_7 + x_2.x_3.x_4.x_5.x_6.x_7 + x_0.x_2.x_3.x_4.x_5.x_6.x_7 + x_1.x_2.x_3.x_4.x_5.x_6.x_7$,

- $y_0 = 1 + x_2 + x_0.x_1.x_3 + x_2.x_3 + x_0.x_1.x_2.x_3 + x_0.x_1.x_4 + x_0.x_1.x_2.x_4 + x_3.x_4 + x_0.x_3.x_4 + x_1.x_3.x_4 + x_0.x_2.x_3.x_4 + x_1.x_2.x_3.x_4 + x_5 + x_0.x_5 + x_0.x_1.x_5 + x_0.x_2.x_5 + x_1.x_2.x_5 + x_0.x_1.x_2.x_5 + x_2.x_3.x_5 + x_0.x_2.x_3.x_5 + x_1.x_2.x_3.x_5 + x_0.x_1.x_2.x_3.x_5 + x_0.x_4.x_5 + x_1.x_4.x_5 + x_0.x_1.x_4.x_5 + x_2.x_4.x_5 + x_0.x_1.x_2.x_4.x_5 + x_3.x_4.x_5 + x_0.x_3.x_4.x_5 + x_1.x_3.x_4.x_5 + x_2.x_3.x_4.x_5 + x_1.x_2.x_3.x_4.x_5 + x_0.x_6 + x_0.x_1.x_6 + x_2.x_6 + x_0.x_2.x_6 + x_0.x_1.x_2.x_6 + x_3.x_6 + x_0.x_3.x_6 + x_2.x_3.x_6 + x_1.x_2.x_3.x_6 + x_0.x_4.x_6 + x_1.x_4.x_6 + x_0.x_1.x_4.x_6 + x_0.x_2.x_4.x_6 + x_1.x_2.x_4.x_6 + x_3.x_4.x_6 + x_0.x_1.x_3.x_4.x_6 + x_1.x_2.x_3.x_4.x_6 + x_0.x_1.x_2.x_3.x_4.x_6 + x_5.x_6 + x_1.x_5.x_6 + x_0.x_1.x_5.x_6 + x_0.x_2.x_5.x_6 + x_3.x_5.x_6 + x_0.x_1.x_3.x_5.x_6 + x_2.x_3.x_5.x_6 + x_0.x_1.x_2.x_3.x_5.x_6 + x_0.x_4.x_5.x_6 + x_0.x_1.x_4.x_5.x_6 + x_0.x_2.x_4.x_5.x_6 + x_3.x_4.x_5.x_6 + x_0.x_2.x_3.x_4.x_5.x_6 + x_1.x_2.x_3.x_4.x_5.x_6 + x_0.x_1.x_2.x_3.x_4.x_5.x_6 + x_7 + x_0.x_7 + x_0.x_1.x_7 + x_3.x_7 + x_2.x_3.x_7 + x_1.x_2.x_3.x_7 + x_0.x_1.x_2.x_3.x_7 + x_4.x_7 + x_0.x_4.x_7 + x_1.x_4.x_7 + x_0.x_1.x_4.x_7 + x_2.x_4.x_7 + x_0.x_2.x_4.x_7 + x_1.x_2.x_4.x_7 + x_3.x_4.x_7 + x_0.x_3.x_4.x_7 + x_0.x_1.x_3.x_4.x_7 + x_2.x_3.x_4.x_7 + x_0.x_2.x_3.x_4.x_7 + x_1.x_2.x_3.x_4.x_7 + x_5.x_7 + x_1.x_5.x_7 + x_0.x_1.x_5.x_7 + x_0.x_2.x_5.x_7 + x_3.x_5.x_7 + x_1.x_3.x_5.x_7 + x_4.x_5.x_7 + x_0.x_4.x_5.x_7 + x_1.x_4.x_5.x_7 + x_0.x_2.x_4.x_5.x_7 + x_3.x_4.x_5.x_7 + x_0.x_3.x_4.x_5.x_7 + x_0.x_1.x_3.x_4.x_5.x_7 + x_1.x_2.x_3.x_4.x_5.x_7 + x_0.x_1.x_2.x_3.x_4.x_5.x_7 + x_6.x_7 + x_0.x_6.x_7 + x_2.x_6.x_7 + x_0.x_2.x_6.x_7 + x_0.x_2.x_3.x_6.x_7 + x_1.x_2.x_3.x_6.x_7 + x_0.x_1.x_2.x_3.x_6.x_7 + x_2.x_4.x_6.x_7 + x_0.x_2.x_4.x_6.x_7 + x_0.x_1.x_2.x_4.x_6.x_7 + x_3.x_4.x_6.x_7 + x_0.x_3.x_4.x_6.x_7 + x_1.x_3.x_4.x_6.x_7 + x_0.x_1.x_3.x_4.x_6.x_7 + x_1.x_2.x_3.x_4.x_6.x_7 + x_0.x_1.x_2.x_3.x_4.x_6.x_7 + x_0.x_5.x_6.x_7 + x_1.x_5.x_6.x_7 + x_0.x_1.x_5.x_6.x_7 + x_2.x_5.x_6.x_7 + x_0.x_2.x_5.x_6.x_7 + x_1.x_3.x_5.x_6.x_7 + x_0.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_2.x_3.x_5.x_6.x_7 + x_0.x_1.x_4.x_5.x_6.x_7 + x_0.x_2.x_4.x_5.x_6.x_7 + x_0.x_1.x_2.x_4.x_5.x_6.x_7 + x_0.x_3.x_4.x_5.x_6.x_7 + x_1.x_3.x_4.x_5.x_6.x_7 + x_0.x_2.x_3.x_4.x_5.x_6.x_7$.

C.2.1 Examples of Real S-boxes

This section lists several examples of S-boxes used in cryptographic primitives. The S-boxes are depicted in tabular form. Most of the S-boxes are bijective mappings. The inverse S-boxes are not always presented but can be easily computed: for an S-box $S : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, if $S[x] = y$ then $S^{-1}[y] = x$ for all $x, y \in \mathbb{Z}_2^n$.

Table C.5 The 3 × 3 S-boxes of CTC, 3WAY (BASEKING), PANAMA (StepRightUp and RadioGatún), PRINTcipher.

Cipher	x							
	0	1	2	3	4	5	6	7
CTC[x]	7	6	0	4	2	5	1	3
3WAY[x]	7	2	4	5	1	6	3	0
PANAMA[x]	7	4	1	6	2	3	5	0
PRINTcipher[x]	0	1	3	6	7	4	5	2

Table C.6 The 4×4 S-boxes of HAMSI (S-box S_2 of SERPENT), PRESENT (and LED), JH (two S-boxes), NOEKEON and SC2000.

Cipher	x															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
HAMSI[x]	8	6	7	9	3	12	10	15	13	1	14	4	0	11	5	2
PRESENT[x]	12	5	6	11	9	0	10	13	3	14	15	8	4	7	1	2
$JH_1[x]$	9	0	4	11	13	12	3	15	1	10	2	6	7	5	8	14
$JH_2[x]$	3	12	6	13	5	7	1	9	15	2	0	4	11	10	14	8
NOEKEON[x]	7	10	2	12	4	8	15	0	5	9	1	14	3	13	11	6
SC2000[x]	2	5	10	12	7	15	1	11	13	6	0	9	4	8	3	14

Table C.7 The eight 4 × 4 SERPENT S-boxes.

S-box	x															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_0[x]$	3	8	15	1	10	6	5	11	14	13	4	2	7	0	9	12
$S_1[x]$	15	12	2	7	9	0	5	10	1	11	14	8	6	13	3	4
$S_2[x]$	8	6	7	9	3	12	10	15	13	1	14	4	0	11	5	2
$S_3[x]$	0	15	11	8	12	9	6	3	13	1	2	4	10	7	5	14
$S_4[x]$	1	15	8	3	12	0	11	6	2	5	4	10	9	14	7	13
$S_5[x]$	15	5	2	11	4	10	9	12	0	3	14	8	13	6	7	1
$S_6[x]$	7	2	12	5	8	4	6	11	14	9	1	15	13	3	10	0
$S_7[x]$	1	13	15	0	14	8	2	11	7	4	12	10	9	3	5	6

Table C.8 The 5 × 5 KECCAK S-box and its inverse.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S[i]	0	5	10	11	20	17	22	23	9	12	3	2	13	8	15	14
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
S[i]	18	21	24	27	6	1	4	7	26	29	16	19	30	25	28	31
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S^{-1}[i]$	0	21	11	10	22	1	20	23	13	8	2	3	9	12	15	14
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$S^{-1}[i]$	26	5	16	27	4	17	6	7	18	29	24	19	30	25	28	31

Table C.9 The 7×7 MISTY1 S-box. For $0 \le i \le 7, 0 \le j \le 15$, the input is $(i \ll 3)|j \in \mathbb{Z}_2^7$ and the output is $S[(i \ll 3)|j] \in \mathbb{Z}_2^7$. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	$1b_x$	32_x	33_x	$5a_x$	$3b_x$	10_x	17_x	54_x	$5b_x$	$1a_x$	72_x	73_x	$6b_x$	$2c_x$	66_x	49_x
1	$1f_x$	24_x	13_x	$6c_x$	37_x	$2e_x$	$3f_x$	$4a_x$	$5d_x$	$0f_x$	40_x	56_x	25_x	51_x	$1c_x$	04_x
2	$0b_x$	46_x	20_x	$0d_x$	$7b_x$	35_x	44_x	42_x	$2b_x$	$1e_x$	41_x	14_x	$4b_x$	79_x	15_x	$6f_x$
3	$0e_x$	55_x	09_x	36_x	74_x	$0c_x$	67_x	53_x	28_x	$0a_x$	$7e_x$	38_x	02_x	07_x	60_x	29_x
4	19_x	12_x	65_x	$2f_x$	30_x	39_x	08_x	68_x	$5f_x$	78_x	$2a_x$	$4c_x$	64_x	45_x	75_x	$3d_x$
5	59_x	48_x	03_x	57_x	$7c_x$	$4f_x$	62_x	$3c_x$	$1d_x$	21_x	$5e_x$	27_x	$6a_x$	70_x	$4d_x$	$3a_x$
6	01_x	$6d_x$	$6e_x$	63_x	18_x	77_x	23_x	05_x	26_x	76_x	00_x	31_x	$2d_x$	$7a_x$	$7f_x$	61_x
7	50_x	22_x	11_x	06_x	47_x	16_x	52_x	$4e_x$	71_x	$3e_x$	69_x	43_x	34_x	$5c_x$	58_x	$7d_x$

Table C.10 The 9 × 9 MISTY1 S-box. For $0 \le i \le 31$, $0 \le j \le 15$, the input is $(i \ll 5)|j \in \mathbb{Z}_2^9$ and the output is $S[(i \ll 5)|j] \in \mathbb{Z}_2^9$. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	$1c3_x$	$0cb_x$	153_x	$19f_x$	$1e3_x$	$0e9_x$	$0fb_x$	035_x	181_x	$0b9_x$	117_x	$1eb_x$	133_x	009_x	$02d_x$	$0d3_x$
1	$0c7_x$	$14a_x$	037_x	$07e_x$	$0eb_x$	164_x	193_x	$1d8_x$	$0a3_x$	$11e_x$	055_x	$02c_x$	$01d_x$	$1a2_x$	163_x	118_x
2	$14b_x$	152_x	$1d2_x$	$00f_x$	$02b_x$	030_x	$13a_x$	$0e5_x$	111_x	138_x	$18e_x$	063_x	$0e3_x$	$0c8_x$	$1f4_x$	$01b_x$
3	001_x	$09d_x$	$0f8_x$	$1a0_x$	$16d_x$	$1f3_x$	$01c_x$	146_x	$07d_x$	$0d1_x$	082_x	$1ea_x$	183_x	$12d_x$	$0f4_x$	$19e_x$
4	$1d3_x$	$0dd_x$	$1e2_x$	128_x	$1e0_x$	$0ec_x$	059_x	091_x	011_x	$12f_x$	026_x	$0dc_x$	$0b0_x$	$18c_x$	$10f_x$	$1f7_x$
5	$0e7_x$	$16c_x$	$0b6_x$	$0f9_x$	$0d8_x$	151_x	101_x	$14c_x$	103_x	$0b8_x$	154_x	$12b_x$	$1ae_x$	017_x	071_x	$00c_x$
6	047_x	058_x	$07f_x$	$1a4_x$	134_x	129_x	084_x	$15d_x$	$19d_x$	$1b2_x$	$1a3_x$	048_x	$07c_x$	051_x	$1ca_x$	023_x
7	$13d_x$	$1a7_x$	165_x	$03b_x$	042_x	$0da_x$	192_x	$0ce_x$	$0c1_x$	$06b_x$	$09f_x$	$1f1_x$	$12c_x$	184_x	$0fa_x$	196_x
8	$1e1_x$	169_x	$17d_x$	031_x	180_x	$10a_x$	094_x	$1da_x$	186_x	$13e_x$	$11c_x$	060_x	175_x	$1cf_x$	067_x	119_x
9	065_x	068_x	099_x	150_x	008_x	007_x	$17c_x$	$0b7_x$	024_x	019_x	$0de_x$	127_x	$0db_x$	$0e4_x$	$1a9_x$	052_x
10	109_x	090_x	$19c_x$	$1c1_x$	028_x	$1b3_x$	135_x	$16a_x$	176_x	$0df_x$	$1e5_x$	188_x	$0c5_x$	$16e_x$	$1de_x$	$1b1_x$
11	$0c3_x$	$1df_x$	036_x	$0ee_x$	$1ee_x$	$0f0_x$	093_x	049_x	$09a_x$	$1b6_x$	069_x	081_x	125_x	$00b_x$	$05e_x$	$0b4_x$
12	149_x	$1c7_x$	174_x	$03e_x$	$13b_x$	$1b7_x$	$08e_x$	$1c6_x$	$0ae_x$	010_x	095_x	$1ef_x$	$04e_x$	$0f2_x$	$1fd_x$	085_x
13	$0fd_x$	$0f6_x$	$0a0_x$	$16f_x$	083_x	$08a_x$	156_x	$09b_x$	$13c_x$	107_x	167_x	098_x	$1d0_x$	$1e9_x$	003_x	$1fe_x$
14	$0bd_x$	122_x	089_x	$0d2_x$	$18f_x$	012_x	033_x	$06a_x$	142_x	$0ed_x$	170_x	$11b_x$	$0e2_x$	$14f_x$	158_x	131_x
15	147_x	$05d_x$	113_x	$1cd_x$	079_x	161_x	$1a5_x$	179_x	$09e_x$	$1b4_x$	$0cc_x$	022_x	132_x	$01a_x$	$0e8_x$	004_x
16	187_x	$1ed_x$	197_x	039_x	$1bf_x$	$1d7_x$	027_x	$18b_x$	$0c6_x$	$09c_x$	$0d0_x$	$14e_x$	$06c_x$	034_x	$1f2_x$	$06e_x$
17	$0ca_x$	025_x	$0ba_x$	191_x	$0fe_x$	013_x	106_x	$02f_x$	$1ad_x$	172_x	$1db_x$	$0c0_x$	$10b_x$	$1d6_x$	$0f5_x$	$1ec_x$
18	$10d_x$	076_x	114_x	$1ab_x$	075_x	$10c_x$	$1e4_x$	159_x	054_x	$11f_x$	$04b_x$	$0c4_x$	$1be_x$	$0f7_x$	029_x	$0a4_x$
19	$00e_x$	$1f0_x$	077_x	$04d_x$	$17a_x$	086_x	$08b_x$	$0b3_x$	171_x	$0bf_x$	$10e_x$	104_x	097_x	$15b_x$	160_x	168_x
20	$0d7_x$	$0bb_x$	066_x	$1ce_x$	$0fc_x$	092_x	$1c5_x$	$06f_x$	016_x	$04a_x$	$0a1_x$	139_x	$0af_x$	$0f1_x$	190_x	$00a_x$
21	$1aa_x$	143_x	$17b_x$	056_x	$18d_x$	166_x	$0d4_x$	$1fb_x$	$14d_x$	194_x	$19a_x$	087_x	$1f8_x$	123_x	$0a7_x$	$1b8_x$
22	141_x	$03c_x$	$1f9_x$	140_x	$02a_x$	155_x	$11a_x$	$1a1_x$	198_x	$0d5_x$	126_x	$1af_x$	061_x	$12e_x$	157_x	$1dc_x$
23	072_x	$18a_x$	$0aa_x$	096_x	115_x	$0ef_x$	045_x	$07b_x$	$08d_x$	145_x	053_x	$05f_x$	178_x	$0b2_x$	$02e_x$	020_x
24	$1d5_x$	$03f_x$	$1c9_x$	$1e7_x$	$1ac_x$	044_x	038_x	014_x	$0b1_x$	$16b_x$	$0ab_x$	$0b5_x$	$05a_x$	182_x	$1c8_x$	$1d4_x$
25	018_x	177_x	064_x	$0cf_x$	$06d_x$	100_x	199_x	130_x	$15a_x$	005_x	120_x	$1bb_x$	$1bd_x$	$0e0_x$	$04f_x$	$0d6_x$
26	$13f_x$	$1c4_x$	$12a_x$	015_x	006_x	$0ff_x$	$19b_x$	$0a6_x$	043_x	088_x	050_x	$15f_x$	$1e8_x$	121_x	073_x	$17e_x$
27	$0bc_x$	$0c2_x$	$0c9_x$	173_x	189_x	$1f5_x$	074_x	$1cc_x$	$1e6_x$	$1a8_x$	195_x	$01f_x$	041_x	$00d_x$	$1ba_x$	032_x
28	$03d_x$	$1d1_x$	080_x	$0a8_x$	057_x	$1b9_x$	162_x	148_x	$0d9_x$	105_x	062_x	$07a_x$	021_x	$1ff_x$	112_x	108_x
29	$1c0_x$	$0a9_x$	$11d_x$	$1b0_x$	$1a6_x$	$0cd_x$	$0f3_x$	$05c_x$	102_x	$05b_x$	$1d9_x$	144_x	$1f6_x$	$0ad_x$	$0a5_x$	$03a_x$
30	$1cb_x$	136_x	$17f_x$	046_x	$0e1_x$	$01e_x$	$1dd_x$	$0e6_x$	137_x	$1fa_x$	185_x	$08c_x$	$08f_x$	040_x	$1b5_x$	$0be_x$
31	078_x	000_x	$0ac_x$	110_x	$15e_x$	124_x	002_x	$1bc_x$	$0a2_x$	$0ea_x$	070_x	$1fc_x$	116_x	$15c_x$	$04c_x$	$1c2_x$

Table C.11 The eight 6×4 DES S-boxes. The input is denoted $(x_1x_2x_3x_4x_5x_6) \in \mathbb{Z}_2^6$. The output is $S_i[x_1x_6][x_2x_3x_4x_5] \in \mathbb{Z}_2^4$.

		$x_2x_3x_4x_5$															
S_1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	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

Table C.12 The eight 6×4 s^2-DES S-boxes. The input is denoted $(x_1x_2x_3x_4x_5x_6) \in \mathbb{Z}_2^6$. The output is $S_i[x_1x_6][x_2x_3x_4x_5] \in \mathbb{Z}_2^4$.

		$x_2x_3x_4x_5$															
S_1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	12	14	1	15	11	10	8	4	7	9	5	0	3	2	13	6
	1	3	5	4	12	9	14	0	8	2	7	10	1	13	6	15	11
	2	10	7	9	11	15	13	2	5	14	6	1	4	12	3	8	0
	3	6	0	5	8	14	7	1	3	12	4	2	13	11	15	10	9
S_2		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	2	12	4	6	3	0	8	5	10	11	15	7	13	1	14	9
	1	14	8	3	11	9	13	10	2	5	0	1	6	7	12	15	4
	2	4	13	14	3	1	10	5	7	9	15	8	12	11	2	0	6
	3	0	11	1	15	4	5	12	14	7	10	9	13	6	8	2	3
S_3		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	5	10	7	12	13	2	0	4	6	14	11	15	3	1	9	8
	1	3	13	6	14	2	0	15	12	1	5	10	7	4	11	8	9
	2	9	2	11	6	1	13	10	15	14	12	3	5	0	8	4	7
	3	1	8	14	11	5	15	9	3	7	6	4	0	12	10	13	2
S_4		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	13	2	12	11	3	1	5	9	15	6	8	0	14	10	4	7
	1	9	1	14	5	11	7	12	4	8	15	0	6	3	2	13	10
	2	14	5	15	13	7	2	9	6	0	12	10	11	4	8	1	3
	3	6	10	5	3	2	11	14	0	7	4	1	8	9	13	15	12
S_5		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	10	2	9	11	8	7	6	3	5	13	12	15	0	4	14	1
	1	2	1	0	5	11	8	15	7	12	9	14	6	3	13	10	4
	2	0	5	8	6	4	3	13	14	9	1	15	11	2	10	7	12
	3	7	14	11	13	12	2	10	9	1	8	0	4	5	6	3	15
S_6		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	0	11	7	10	12	9	14	6	1	3	5	15	2	4	8	13
	1	3	1	4	5	0	12	8	7	10	2	14	13	6	9	15	11
	2	5	12	15	6	7	11	10	13	0	8	9	14	4	2	1	3
	3	4	8	10	11	6	5	7	1	14	15	12	2	3	13	9	0
S_7		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	5	0	11	14	10	2	9	8	13	3	12	6	4	7	1	15
	1	10	12	13	4	9	1	3	0	6	8	5	15	14	11	2	7
	2	6	11	12	9	0	3	4	14	1	7	8	13	10	2	15	5
	3	9	4	7	0	3	11	2	1	15	5	6	8	12	13	10	14
S_8		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	7	13	4	14	10	15	8	2	11	9	6	3	1	12	5	0
	1	6	14	10	12	5	7	0	1	2	13	11	4	8	9	15	3
	2	10	6	12	1	11	9	14	3	13	15	4	5	0	2	8	7
	3	5	3	15	6	0	1	13	9	4	10	14	8	12	7	11	2

Table C.13 The eight 6×4 s^3-DES S-boxes. The input is denoted $(x_1x_2x_3x_4x_5x_6) \in \mathbb{Z}_2^6$. The output is $S_i[x_1x_6][x_2x_3x_4x_5] \in \mathbb{Z}_2^4$.

		$x_2x_3x_4x_5$															
S_1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	15	8	3	14	4	2	9	5	0	11	10	1	13	7	6	12
	1	6	15	9	5	3	12	10	0	13	8	4	11	14	2	1	7
	2	9	14	5	8	2	4	15	3	10	7	6	13	1	11	12	0
	3	10	5	3	15	12	9	0	6	1	2	8	4	11	14	7	13
S_2		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	13	14	0	3	10	4	7	9	11	8	12	6	1	15	2	5
	1	8	2	11	13	4	1	14	7	5	15	0	3	10	6	9	12
	2	14	9	3	10	0	7	13	4	8	5	6	15	11	12	1	2
	3	1	4	14	7	11	13	8	2	6	3	5	10	12	0	15	9
S_3		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	13	3	11	5	14	8	0	6	4	15	1	12	7	2	10	9
	1	4	13	1	8	7	2	14	11	15	10	12	3	9	5	0	6
	2	6	5	8	11	13	14	3	0	9	2	4	1	10	7	15	12
	3	1	11	7	2	8	13	4	14	6	12	10	15	3	0	9	5
S_4		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	9	0	7	11	12	5	10	6	15	3	1	14	2	8	4	13
	1	5	10	12	6	0	15	3	9	8	13	11	1	7	2	14	4
	2	10	7	9	12	5	0	6	11	3	14	4	2	8	13	15	1
	3	3	9	15	0	6	10	5	12	14	2	1	7	13	4	8	11
S_5		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	5	15	9	10	0	3	14	4	2	12	7	1	13	6	8	11
	1	6	9	3	15	5	12	0	10	8	7	13	4	2	11	14	1
	2	15	0	10	9	3	5	4	14	8	11	1	7	6	12	13	2
	3	12	5	0	6	15	10	9	3	7	2	14	11	8	1	4	13
S_6		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	4	3	7	10	9	0	14	13	15	5	12	6	2	11	1	8
	1	14	13	11	4	2	7	1	8	9	10	5	3	15	0	12	6
	2	13	0	10	9	4	3	7	14	1	15	6	12	8	5	11	2
	3	1	7	4	14	11	8	13	2	10	12	3	5	6	15	0	9
S_7		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	4	10	15	12	2	9	1	6	11	5	0	3	7	14	13	8
	1	10	15	6	0	5	3	12	9	1	8	11	13	14	4	7	2
	2	2	12	9	6	15	10	4	1	5	11	3	0	8	7	14	13
	3	12	6	3	9	0	5	10	15	2	13	4	14	7	11	1	8
S_8		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	13	10	0	7	3	9	14	4	2	15	12	1	5	6	11	8
	1	2	7	13	1	4	14	11	8	15	12	6	10	9	5	0	3
	2	4	13	14	0	9	3	7	10	1	8	2	11	15	5	12	6
	3	8	11	7	14	2	4	13	1	6	5	9	0	12	15	3	10

Table C.14 The eight 6×4 s^5-DES S-boxes. The input is denoted $(x_1x_2x_3x_4x_5x_6) \in \mathbb{Z}_2^6$. The output is $S_i[x_1x_6][x_2x_3x_4x_5] \in \mathbb{Z}_2^4$.

		$x_2x_3x_4x_5$															
S_1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	9	10	15	1	4	7	2	12	6	5	3	14	8	11	13	0
	1	2	13	8	4	11	1	14	7	12	3	15	9	5	6	0	10
	2	10	12	4	7	9	2	15	1	3	6	13	8	14	5	0	11
	3	4	11	1	13	14	7	8	2	10	0	6	3	9	12	15	5
S_2		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	6	3	5	0	8	14	11	13	9	10	12	7	15	4	2	1
	1	9	6	10	12	15	0	5	3	4	1	7	11	2	13	14	8
	2	5	8	3	14	6	13	0	11	10	15	9	2	12	1	7	4
	3	6	3	15	9	0	10	12	5	13	8	2	4	11	7	1	14
S_3		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	11	5	8	2	6	12	1	15	7	14	13	4	0	9	10	3
	1	7	8	1	14	11	2	13	4	12	3	6	9	5	15	0	10
	2	8	11	1	12	15	6	2	5	4	7	10	9	3	0	13	14
	3	13	2	4	7	1	11	14	8	10	9	15	0	12	6	3	5
S_4		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	13	11	8	14	3	0	6	5	4	7	2	9	15	12	1	10
	1	10	0	3	5	15	6	12	9	1	13	4	14	8	11	2	7
	2	6	5	11	8	0	14	13	3	9	12	7	2	10	1	4	15
	3	9	12	5	15	6	3	0	10	7	11	2	8	13	4	14	1
S_5		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	12	6	2	11	5	8	15	1	3	13	9	14	0	7	10	4
	1	15	0	12	5	3	6	9	10	4	11	2	8	14	1	7	13
	2	1	12	15	5	6	11	8	2	4	7	10	9	13	0	3	14
	3	6	3	10	0	9	12	5	15	13	4	1	14	7	11	8	2
S_6		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	14	8	2	5	9	15	4	3	7	1	12	6	0	10	11	13
	1	1	13	11	8	2	4	7	14	10	6	0	15	5	9	12	3
	2	4	2	9	15	14	8	3	5	10	7	0	12	13	1	6	11
	3	8	11	7	4	13	1	14	2	5	0	9	10	6	15	3	12
S_7		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	4	13	10	3	7	0	9	14	2	1	15	6	12	11	5	8
	1	9	0	15	10	12	6	5	3	14	7	1	13	11	8	2	4
	2	13	10	3	9	0	7	14	4	8	6	5	12	11	1	2	15
	3	10	3	12	6	5	9	0	15	4	8	11	1	14	7	13	2
S_8		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1x_6	0	1	10	2	12	15	9	4	7	14	3	5	0	8	6	11	13
	1	14	13	7	11	2	4	1	8	0	10	9	6	5	15	12	3
	2	10	15	12	1	9	2	7	4	13	0	6	11	3	5	8	14
	3	4	8	1	2	7	11	13	14	10	5	15	12	0	6	3	9

Table C.15 The 8×8 AES S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	63_x	$7c_x$	77_x	$7b_x$	$f2_x$	$6b_x$	$6f_x$	$c5_x$	30_x	01_x	67_x	$2b_x$	fe_x	$d7_x$	ab_x	76_x
1	ca_x	82_x	$c9_x$	$7d_x$	fa_x	59_x	47_x	$f0_x$	ad_x	$d4_x$	$a2_x$	af_x	$9c_x$	$a4_x$	72_x	$c0_x$
2	$b7_x$	fd_x	93_x	26_x	36_x	$3f_x$	$f7_x$	cc_x	34_x	$a5_x$	$e5_x$	$f1_x$	71_x	$d8_x$	31_x	15_x
3	04_x	$c7_x$	23_x	$c3_x$	18_x	96_x	05_x	$9a_x$	07_x	12_x	80_x	$e2_x$	eb_x	27_x	$b2_x$	75_x
4	09_x	83_x	$2c_x$	$1a_x$	$1b_x$	$6e_x$	$5a_x$	$a0_x$	52_x	$3b_x$	$d6_x$	$b3_x$	29_x	$e3_x$	$2f_x$	84_x
5	53_x	$d1_x$	00_x	ed_x	20_x	fc_x	$b1_x$	$5b_x$	$6a_x$	cb_x	be_x	39_x	$4a_x$	$4c_x$	58_x	cf_x
6	$d0_x$	ef_x	aa_x	fb_x	43_x	$4d_x$	33_x	85_x	45_x	$f9_x$	02_x	$7f_x$	50_x	$3c_x$	$9f_x$	$a8_x$
7	51_x	$a3_x$	40_x	$8f_x$	92_x	$9d_x$	38_x	$f5_x$	bc_x	$b6_x$	da_x	21_x	10_x	ff_x	$f3_x$	$d2_x$
8	cd_x	$0c_x$	13_x	ec_x	$5f_x$	97_x	44_x	17_x	$c4_x$	$a7_x$	$7e_x$	$3d_x$	64_x	$5d_x$	19_x	73_x
9	60_x	81_x	$4f_x$	dc_x	22_x	$2a_x$	90_x	88_x	46_x	ee_x	$b8_x$	14_x	de_x	$5e_x$	$0b_x$	db_x
10	$e0_x$	32_x	$3a_x$	$0a_x$	49_x	06_x	24_x	$5c_x$	$c2_x$	$d3_x$	ac_x	62_x	91_x	95_x	$e4_x$	79_x
11	$e7_x$	$c8_x$	37_x	$6d_x$	$8d_x$	$d5_x$	$4e_x$	$a9_x$	$6c_x$	56_x	$f4_x$	ea_x	65_x	$7a_x$	ae_x	08_x
12	ba_x	78_x	25_x	$2e_x$	$1c_x$	$a6_x$	$b4_x$	$c6_x$	$e8_x$	dd_x	74_x	$1f_x$	$4b_x$	bd_x	$8b_x$	$8a_x$
13	70_x	$3e_x$	$b5_x$	66_x	48_x	03_x	$f6_x$	$0e_x$	61_x	35_x	57_x	$b9_x$	86_x	$c1_x$	$1d_x$	$9e_x$
14	$e1_x$	$f8_x$	98_x	11_x	69_x	$d9_x$	$8e_x$	94_x	$9b_x$	$1e_x$	87_x	$e9_x$	ce_x	55_x	28_x	df_x
15	$8c_x$	$a1_x$	89_x	$0d_x$	bf_x	$e6_x$	42_x	68_x	41_x	99_x	$2d_x$	$0f_x$	$b0_x$	54_x	bb_x	16_x

Table C.16 The 8×8 ARIA S-box 1. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	63_x	$7c_x$	77_x	$7b_x$	$f2_x$	$6b_x$	$6f_x$	$c5_x$	30_x	01_x	67_x	$2b_x$	fe_x	$d7_x$	ab_x	76_x
1	ca_x	82_x	$c9_x$	$7d_x$	fa_x	59_x	47_x	$f0_x$	ad_x	$d4_x$	$a2_x$	af_x	$9c_x$	$a4_x$	72_x	$c0_x$
2	$b7_x$	fd_x	93_x	26_x	36_x	$3f_x$	$f7_x$	cc_x	34_x	$a5_x$	$e5_x$	$f1_x$	71_x	$d8_x$	31_x	15_x
3	04_x	$c7_x$	23_x	$c3_x$	18_x	96_x	05_x	$9a_x$	07_x	12_x	80_x	$e2_x$	eb_x	27_x	$b2_x$	75_x
4	09_x	83_x	$2c_x$	$1a_x$	$1b_x$	$6e_x$	$5a_x$	$a0_x$	52_x	$3b_x$	$d6_x$	$b3_x$	29_x	$e3_x$	$2f_x$	84_x
5	53_x	$d1_x$	00_x	ed_x	20_x	fc_x	$b1_x$	$5b_x$	$6a_x$	cb_x	be_x	39_x	$4a_x$	$4c_x$	58_x	cf_x
6	$d0_x$	ef_x	aa_x	fb_x	43_x	$4d_x$	33_x	85_x	45_x	$f9_x$	02_x	$7f_x$	50_x	$3c_x$	$9f_x$	$a8_x$
7	51_x	$a3_x$	40_x	$8f_x$	92_x	$9d_x$	38_x	$f5_x$	bc_x	$b6_x$	da_x	21_x	10_x	ff_x	$f3_x$	$d2_x$
8	cd_x	$0c_x$	13_x	ec_x	$5f_x$	97_x	44_x	17_x	$c4_x$	$a7_x$	$7e_x$	$3d_x$	64_x	$5d_x$	19_x	73_x
9	60_x	81_x	$4f_x$	dc_x	22_x	$2a_x$	90_x	88_x	46_x	ee_x	$b8_x$	14_x	de_x	$5e_x$	$0b_x$	db_x
10	$e0_x$	32_x	$3a_x$	$0a_x$	49_x	06_x	24_x	$5c_x$	$c2_x$	$d3_x$	ac_x	62_x	91_x	95_x	$e4_x$	79_x
11	$e7_x$	$c8_x$	37_x	$6d_x$	$8d_x$	$d5_x$	$4e_x$	$a9_x$	$6c_x$	56_x	$f4_x$	ea_x	65_x	$7a_x$	ae_x	08_x
12	ba_x	78_x	25_x	$2e_x$	$1c_x$	$a6_x$	$b4_x$	$c6_x$	$e8_x$	dd_x	74_x	$1f_x$	$4b_x$	bd_x	$8b_x$	$8a_x$
13	70_x	$3e_x$	$b5_x$	66_x	48_x	03_x	$f6_x$	$0e_x$	61_x	35_x	57_x	$b9_x$	86_x	$c1_x$	$1d_x$	$9e_x$
14	$e1_x$	$f8_x$	98_x	11_x	69_x	$d9_x$	$8e_x$	94_x	$9b_x$	$1e_x$	87_x	$e9_x$	ce_x	55_x	28_x	df_x
15	$8c_x$	$a1_x$	89_x	$0d_x$	bf_x	$e6_x$	42_x	68_x	41_x	99_x	$2d_x$	$0f_x$	$b0_x$	54_x	bb_x	16_x

Table C.17 The 8×8 ARIA S-box 2. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	$e2_x$	$4e_x$	54_x	fc_x	94_x	$c2_x$	$4a_x$	cc_x	62_x	$0d_x$	$6a_x$	46_x	$3c_x$	$4d_x$	$8b_x$	$d1_x$
1	$5e_x$	fa_x	64_x	cb_x	$b4_x$	97_x	be_x	$2b_x$	bc_x	77_x	$2e_x$	03_x	$d3_x$	19_x	59_x	$c1_x$
2	$1d_x$	06_x	41_x	$6b_x$	55_x	$f0_x$	99_x	69_x	ea_x	$9c_x$	18_x	ae_x	63_x	df_x	$e7_x$	bb_x
3	00_x	73_x	66_x	fb_x	96_x	$4c_x$	85_x	$e4_x$	$3a_x$	09_x	45_x	aa_x	$0f_x$	ee_x	10_x	eb_x
4	$2d_x$	$7f_x$	$f4_x$	29_x	ac_x	cf_x	ad_x	91_x	$8d_x$	78_x	$c8_x$	95_x	$f9_x$	$2f_x$	ce_x	cd_x
5	08_x	$7a_x$	88_x	38_x	$5c_x$	83_x	$2a_x$	28_x	47_x	db_x	$b8_x$	$c7_x$	93_x	$a4_x$	12_x	53_x
6	ff_x	87_x	$0e_x$	31_x	36_x	21_x	58_x	48_x	01_x	$8e_x$	37_x	74_x	32_x	ca_x	$e9_x$	$b1_x$
7	$b7_x$	ab_x	$0c_x$	$d7_x$	$c4_x$	56_x	42_x	26_x	07_x	98_x	60_x	$d9_x$	$b6_x$	$b9_x$	11_x	40_x
8	ec_x	20_x	$8c_x$	bd_x	$a0_x$	$c9_x$	84_x	04_x	49_x	23_x	$f1_x$	$4f_x$	50_x	$1f_x$	13_x	dc_x
9	$d8_x$	$c0_x$	$9e_x$	57_x	$e3_x$	$c3_x$	$7b_x$	65_x	$3b_x$	02_x	$8f_x$	$3e_x$	$e8_x$	25_x	92_x	$e5_x$
10	15_x	dd_x	fd_x	17_x	$a9_x$	bf_x	$d4_x$	$9a_x$	$7e_x$	$c5_x$	39_x	67_x	fe_x	76_x	$9d_x$	43_x
11	$a7_x$	$e1_x$	$d0_x$	$f5_x$	68_x	$f2_x$	$1b_x$	34_x	70_x	05_x	$a3_x$	$8a_x$	$d5_x$	79_x	86_x	$a8_x$
12	30_x	$c6_x$	51_x	$4b_x$	$1e_x$	$a6_x$	27_x	$f6_x$	35_x	$d2_x$	$6e_x$	24_x	16_x	82_x	$5f_x$	da_x
13	$e6_x$	75_x	$a2_x$	ef_x	$2c_x$	$b2_x$	$1c_x$	$9f_x$	$5d_x$	$6f_x$	80_x	$0a_x$	72_x	44_x	$9b_x$	$6c_x$
14	90_x	$0b_x$	$5b_x$	33_x	$7d_x$	$5a_x$	52_x	$f3_x$	61_x	$a1_x$	$f7_x$	$b0_x$	$d6_x$	$3f_x$	$7c_x$	$6d_x$
15	ed_x	14_x	$e0_x$	$a5_x$	$3d_x$	22_x	$b3_x$	$f8_x$	89_x	de_x	71_x	$1a_x$	af_x	ba_x	$b5_x$	81_x

Table C.18 The 8×8 CAMELLIA S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	70_x	82_x	$2c_x$	ec_x	$b3_x$	27_x	$c0_x$	$e5_x$	$e4_x$	85_x	57_x	35_x	ea_x	$0c_x$	ae_x	41_x
1	23_x	ef_x	$6b_x$	93_x	45_x	19_x	$a5_x$	21_x	ed_x	$0e_x$	$4f_x$	$4e_x$	$1d_x$	65_x	92_x	bd_x
2	86_x	$b8_x$	af_x	$8f_x$	$7c_x$	eb_x	$1f_x$	ce_x	$3e_x$	30_x	dc_x	$5f_x$	$5e_x$	$c5_x$	$0b_x$	$1a_x$
3	$a6_x$	$e1_x$	39_x	ca_x	$d5_x$	47_x	$5d_x$	$3d_x$	$d9_x$	01_x	$5a_x$	$d6_x$	51_x	56_x	$6c_x$	$4d_x$
4	$8b_x$	$0d_x$	$9a_x$	66_x	fb_x	cc_x	$b0_x$	$2d_x$	74_x	12_x	$2b_x$	20_x	$f0_x$	$b1_x$	84_x	99_x
5	df_x	$4c_x$	cb_x	$c2_x$	34_x	$7e_x$	76_x	05_x	$6d_x$	$b7_x$	$a9_x$	31_x	$d1_x$	17_x	04_x	$d7_x$
6	14_x	58_x	$3a_x$	61_x	de_x	$1b_x$	11_x	$1c_x$	32_x	$0f_x$	$9c_x$	16_x	53_x	18_x	$f2_x$	22_x
7	fe_x	44_x	cf_x	$b2_x$	$c3_x$	$b5_x$	$7a_x$	91_x	24_x	08_x	$e8_x$	$a8_x$	60_x	fc_x	69_x	50_x
8	aa_x	$d0_x$	$a0_x$	$7d_x$	$a1_x$	89_x	62_x	97_x	54_x	$5b_x$	$1e_x$	95_x	$e0_x$	ff_x	64_x	$d2_x$
9	10_x	$c4_x$	00_x	48_x	$a3_x$	$f7_x$	75_x	db_x	$8a_x$	03_x	$e6_x$	da_x	09_x	$3f_x$	dd_x	94_x
10	87_x	$5c_x$	83_x	02_x	cd_x	$4a_x$	90_x	33_x	73_x	67_x	$f6_x$	$f3_x$	$9d_x$	$7f_x$	bf_x	$e2_x$
11	52_x	$9b_x$	$d8_x$	26_x	$c8_x$	37_x	$c6_x$	$3b_x$	81_x	96_x	$6f_x$	$4b_x$	13_x	be_x	63_x	$2e_x$
12	$e9_x$	79_x	$a7_x$	$8c_x$	$9f_x$	$6e_x$	bc_x	$8e_x$	29_x	$f5_x$	$f9_x$	$b6_x$	$2f_x$	fd_x	$b4_x$	59_x
13	78_x	98_x	06_x	$6a_x$	$e7_x$	46_x	71_x	ba_x	$d4_x$	25_x	ab_x	42_x	88_x	$a2_x$	$8d_x$	fa_x
14	72_x	07_x	$b9_x$	55_x	$f8_x$	ee_x	ac_x	$0a_x$	36_x	49_x	$2a_x$	68_x	$3c_x$	38_x	$f1_x$	$a4_x$
15	40_x	28_x	$d3_x$	$7b_x$	bb_x	$c9_x$	43_x	$c1_x$	15_x	$e3_x$	ad_x	$f4_x$	77_x	$c7_x$	80_x	$9e_x$

Table C.19 The 8×8 CS S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	29_x	$0d_x$	61_x	40_x	$9c_x$	eb_x	$9e_x$	$8f_x$	$1f_x$	85_x	$5f_x$	58_x	$5b_x$	01_x	39_x	86_x
1	97_x	$2e_x$	$d7_x$	$d6_x$	35_x	ae_x	17_x	16_x	21_x	$b6_x$	69_x	$4e_x$	$a5_x$	72_x	87_x	08_x
2	$3c_x$	18_x	$e6_x$	$e7_x$	fa_x	ad_x	$b8_x$	89_x	$b7_x$	00_x	$f7_x$	$6f_x$	73_x	84_x	11_x	63_x
3	$3f_x$	96_x	$7f_x$	$6e_x$	bf_x	14_x	$9d_x$	ac_x	$a4_x$	$0e_x$	$7e_x$	$f6_x$	20_x	$4a_x$	62_x	30_x
4	03_x	$c5_x$	$4b_x$	$5a_x$	46_x	$a3_x$	44_x	65_x	$7d_x$	$4d_x$	$3d_x$	42_x	79_x	49_x	$1b_x$	$5c_x$
5	$f5_x$	$6c_x$	$b5_x$	94_x	54_x	ff_x	56_x	57_x	$0b_x$	$f4_x$	43_x	$0c_x$	$4f_x$	70_x	$6d_x$	$0a_x$
6	$e4_x$	02_x	$3e_x$	$2f_x$	$a2_x$	47_x	$e0_x$	$c1_x$	$d5_x$	$1a_x$	95_x	$a7_x$	51_x	$5e_x$	33_x	$2b_x$
7	$5d_x$	$d4_x$	$1d_x$	$2c_x$	ee_x	75_x	ec_x	dd_x	$7c_x$	$4c_x$	$a6_x$	$b4_x$	78_x	48_x	$3a_x$	32_x
8	98_x	af_x	$c0_x$	$e1_x$	$2d_x$	09_x	$0f_x$	$1e_x$	$b9_x$	27_x	$8a_x$	$e9_x$	bd_x	$e3_x$	$9f_x$	07_x
9	$b1_x$	ea_x	92_x	93_x	53_x	$6a_x$	31_x	10_x	80_x	$f2_x$	$d8_x$	$9b_x$	04_x	36_x	06_x	$8e_x$
10	be_x	$a9_x$	64_x	45_x	38_x	$1c_x$	$7a_x$	$6b_x$	$f3_x$	$a1_x$	$f0_x$	cd_x	37_x	25_x	15_x	81_x
11	fb_x	90_x	$e8_x$	$d9_x$	$7b_x$	52_x	19_x	28_x	26_x	88_x	fc_x	$d1_x$	$e2_x$	$8c_x$	$a0_x$	34_x
12	82_x	67_x	da_x	cb_x	$c7_x$	41_x	$e5_x$	$c4_x$	$c8_x$	ef_x	db_x	$c3_x$	cc_x	ab_x	ce_x	ed_x
13	$d0_x$	bb_x	$d3_x$	$d2_x$	71_x	68_x	13_x	12_x	$9a_x$	$b3_x$	$c2_x$	ca_x	de_x	77_x	dc_x	df_x
14	66_x	83_x	bc_x	$8d_x$	60_x	$c6_x$	22_x	23_x	$b2_x$	$8b_x$	91_x	05_x	76_x	cf_x	74_x	$c9_x$
15	aa_x	$f1_x$	99_x	$a8_x$	59_x	50_x	$3b_x$	$2a_x$	fe_x	$f9_x$	24_x	$b0_x$	ba_x	fd_x	$f8_x$	55_x

Table C.20 The 8×8 E2 S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	$e1_x$	42_x	$3e_x$	81_x	$4e_x$	17_x	$9e_x$	fd_x	$b4_x$	$3f_x$	$2c_x$	da_x	31_x	$1e_x$	$e0_x$	41_x
1	cc_x	$f3_x$	82_x	$7d_x$	$7c_x$	12_x	$8e_x$	bb_x	$e4_x$	58_x	15_x	$d5_x$	$6f_x$	$e9_x$	$4c_x$	$4b_x$
2	35_x	$7b_x$	$5a_x$	$9a_x$	90_x	45_x	bc_x	$f8_x$	79_x	$d6_x$	$1b_x$	88_x	02_x	ab_x	cf_x	64_x
3	09_x	$0c_x$	$f0_x$	01_x	$a4_x$	$b0_x$	$f6_x$	93_x	43_x	63_x	86_x	dc_x	11_x	$a5_x$	83_x	$8b_x$
4	$c9_x$	$d0_x$	19_x	95_x	$6a_x$	$a1_x$	$5c_x$	24_x	$6e_x$	50_x	21_x	80_x	$2f_x$	$e7_x$	53_x	$0f_x$
5	91_x	22_x	04_x	ed_x	$a6_x$	48_x	49_x	67_x	ec_x	$f7_x$	$c0_x$	39_x	ce_x	$f2_x$	$2d_x$	be_x
6	$5d_x$	$1c_x$	$e3_x$	87_x	07_x	$0d_x$	$7a_x$	$f4_x$	fb_x	32_x	$f5_x$	$8c_x$	db_x	$8f_x$	25_x	96_x
7	$a8_x$	ea_x	cd_x	33_x	65_x	54_x	06_x	$8d_x$	89_x	$0a_x$	$5e_x$	$d9_x$	16_x	$0e_x$	71_x	$6c_x$
8	$0b_x$	ff_x	60_x	$d2_x$	$2e_x$	$d3_x$	$c8_x$	55_x	$c2_x$	23_x	$b7_x$	74_x	$e2_x$	$9b_x$	df_x	77_x
9	$2b_x$	$b9_x$	$3c_x$	62_x	13_x	$e5_x$	94_x	34_x	$b1_x$	27_x	84_x	$9f_x$	$d7_x$	51_x	00_x	61_x
10	ad_x	85_x	73_x	03_x	08_x	40_x	ef_x	68_x	fe_x	97_x	$1f_x$	de_x	af_x	66_x	$e8_x$	$b8_x$
11	ae_x	bd_x	$b3_x$	eb_x	$c6_x$	$6b_x$	47_x	$a9_x$	$d8_x$	$a7_x$	72_x	ee_x	$1d_x$	$7e_x$	aa_x	$b6_x$
12	75_x	cb_x	$d4_x$	30_x	69_x	20_x	$7f_x$	37_x	$5b_x$	$9d_x$	78_x	$a3_x$	$f1_x$	76_x	fa_x	05_x
13	$3d_x$	$3a_x$	44_x	57_x	$3b_x$	ca_x	$c7_x$	$8a_x$	18_x	46_x	$9c_x$	bf_x	ba_x	38_x	56_x	$1a_x$
14	92_x	$4d_x$	26_x	29_x	$a2_x$	98_x	10_x	99_x	70_x	$a0_x$	$c5_x$	28_x	$c1_x$	$6d_x$	14_x	ac_x
15	$f9_x$	$5f_x$	$4f_x$	$c4_x$	$c3_x$	$d1_x$	fc_x	dd_x	$b2_x$	59_x	$e6_x$	$b5_x$	36_x	52_x	$4a_x$	$2a_x$

Table C.21 The 8×8 FOX S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	$5d_x$	de_x	00_x	$b7_x$	$d3_x$	ca_x	$3c_x$	$0d_x$	$c3_x$	$f8_x$	cb_x	$8d_x$	76_x	89_x	aa_x	12_x
1	88_x	22_x	$4f_x$	db_x	$6d_x$	47_x	$e4_x$	$4c_x$	78_x	$9a_x$	49_x	93_x	$c4_x$	$c0_x$	86_x	13_x
2	$a9_x$	20_x	53_x	$1c_x$	$4e_x$	cf_x	35_x	39_x	$b4_x$	$a1_x$	54_x	64_x	03_x	$c7_x$	85_x	$5c_x$
3	$5b_x$	cd_x	$d8_x$	72_x	96_x	42_x	$b8_x$	$e1_x$	$a2_x$	60_x	ef_x	bd_x	02_x	af_x	$8c_x$	73_x
4	$7c_x$	$7f_x$	$5e_x$	$f9_x$	65_x	$e6_x$	eb_x	ad_x	$5a_x$	$a5_x$	79_x	$8e_x$	15_x	30_x	ec_x	$a4_x$
5	$c2_x$	$3e_x$	$e0_x$	74_x	51_x	fb_x	$2d_x$	$6e_x$	94_x	$4d_x$	55_x	34_x	ae_x	52_x	$7e_x$	$9d_x$
6	$4a_x$	$f7_x$	80_x	$f0_x$	$d0_x$	90_x	$a7_x$	$e8_x$	$9f_x$	50_x	$d5_x$	$d1_x$	98_x	cc_x	$a0_x$	17_x
7	$f4_x$	$b6_x$	$c1_x$	28_x	$5f_x$	26_x	01_x	ab_x	25_x	38_x	82_x	$7d_x$	48_x	fc_x	$1b_x$	ce_x
8	$3f_x$	$6b_x$	$e2_x$	67_x	66_x	43_x	59_x	19_x	84_x	$3d_x$	$f5_x$	$2f_x$	$c9_x$	bc_x	$d9_x$	95_x
9	29_x	41_x	da_x	$1a_x$	$b0_x$	$e9_x$	69_x	$d2_x$	$7b_x$	$d7_x$	11_x	$9b_x$	33_x	$8a_x$	23_x	09_x
10	$d4_x$	71_x	44_x	68_x	$6f_x$	$f2_x$	$0e_x$	df_x	87_x	dc_x	83_x	18_x	$6a_x$	ee_x	99_x	81_x
11	62_x	36_x	$2e_x$	$7a_x$	fe_x	45_x	$9c_x$	75_x	91_x	$0c_x$	$0f_x$	$e7_x$	$f6_x$	14_x	63_x	$1d_x$
12	$0b_x$	$8b_x$	$b3_x$	$f3_x$	$b2_x$	$3b_x$	08_x	$4b_x$	10_x	$a6_x$	32_x	$b9_x$	$a8_x$	92_x	$f1_x$	56_x
13	dd_x	21_x	bf_x	04_x	be_x	$d6_x$	fd_x	77_x	ea_x	$3a_x$	$c8_x$	$8f_x$	57_x	$1e_x$	fa_x	$2b_x$
14	58_x	$c5_x$	27_x	ac_x	$e3_x$	ed_x	97_x	bb_x	46_x	05_x	40_x	31_x	$e5_x$	37_x	$2c_x$	$9e_x$
15	$0a_x$	$b1_x$	$b5_x$	06_x	$6c_x$	$1f_x$	$a3_x$	$2a_x$	70_x	ff_x	ba_x	07_x	24_x	16_x	$c6_x$	61_x

Table C.22 The 8×8 HIEROCRYPT-1/L3 S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	07_x	fc_x	55_x	70_x	98_x	$8e_x$	84_x	$4e_x$	bc_x	75_x	ce_x	18_x	02_x	$e9_x$	$5d_x$	80_x
1	$1c_x$	60_x	78_x	42_x	$9d_x$	$2e_x$	$f5_x$	$e8_x$	$c6_x$	$7a_x$	$2f_x$	$a4_x$	$b2_x$	$5f_x$	19_x	87_x
2	$0b_x$	$9b_x$	$9c_x$	$d3_x$	$c3_x$	77_x	$3d_x$	$6f_x$	$b9_x$	$2d_x$	$4d_x$	$f7_x$	$8c_x$	$a7_x$	ac_x	17_x
3	$3c_x$	$5a_x$	41_x	$c9_x$	29_x	ed_x	de_x	27_x	69_x	30_x	72_x	$a8_x$	95_x	$3e_x$	$f9_x$	$d8_x$
4	21_x	$8b_x$	44_x	$d7_x$	11_x	$0d_x$	48_x	fd_x	$6a_x$	01_x	57_x	$e5_x$	bd_x	85_x	ec_x	$1e_x$
5	37_x	$9f_x$	$b5_x$	$9a_x$	$7c_x$	09_x	$f1_x$	$b1_x$	94_x	81_x	82_x	08_x	fb_x	$c0_x$	51_x	$0f_x$
6	61_x	$7f_x$	$1a_x$	56_x	96_x	13_x	$c1_x$	67_x	99_x	03_x	$5e_x$	$b6_x$	ca_x	fa_x	$9e_x$	df_x
7	$d6_x$	83_x	cc_x	$a2_x$	12_x	23_x	$b7_x$	65_x	$d0_x$	39_x	$7d_x$	$3b_x$	$d5_x$	$b0_x$	af_x	$1f_x$
8	06_x	$c8_x$	34_x	$c5_x$	$1b_x$	79_x	$4b_x$	66_x	bf_x	88_x	$4a_x$	$c4_x$	ef_x	58_x	$3f_x$	$0a_x$
9	$2c_x$	73_x	$d1_x$	$f8_x$	$6b_x$	$e6_x$	20_x	$b8_x$	22_x	43_x	$b3_x$	33_x	$e7_x$	$f0_x$	71_x	$7e_x$
10	52_x	89_x	47_x	63_x	$0e_x$	$6d_x$	$e3_x$	be_x	59_x	64_x	ee_x	$f6_x$	38_x	$5c_x$	$f4_x$	$5b_x$
11	49_x	$d4_x$	$e0_x$	$f3_x$	bb_x	54_x	26_x	$2b_x$	00_x	86_x	90_x	ff_x	fe_x	$a6_x$	$7b_x$	05_x
12	ad_x	68_x	$a1_x$	10_x	eb_x	$c7_x$	$e2_x$	$f2_x$	46_x	$8a_x$	$6c_x$	14_x	$6e_x$	cf_x	35_x	45_x
13	50_x	$d2_x$	92_x	74_x	93_x	$e1_x$	da_x	ae_x	$a9_x$	53_x	$e4_x$	40_x	cd_x	ba_x	97_x	$a3_x$
14	91_x	31_x	25_x	76_x	36_x	32_x	28_x	$3a_x$	24_x	$4c_x$	db_x	$d9_x$	$8d_x$	dc_x	62_x	$2a_x$
15	ea_x	15_x	dd_x	$c2_x$	$a5_x$	$0c_x$	04_x	$1d_x$	$8f_x$	cb_x	$b4_x$	$4f_x$	16_x	ab_x	aa_x	$a0_x$

Table C.23 The 8×8 SAFER K/SK/+/++ S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	01_x	$2d_x$	$e2_x$	93_x	be_x	45_x	15_x	ae_x	78_x	03_x	87_x	$a4_x$	$b8_x$	38_x	cf_x	$3f_x$
1	08_x	67_x	09_x	94_x	eb_x	26_x	$a8_x$	$6b_x$	bd_x	18_x	34_x	$1b_x$	bb_x	bf_x	72_x	$f7_x$
2	40_x	35_x	48_x	$9c_x$	51_x	$2f_x$	$3b_x$	55_x	$e3_x$	$c0_x$	$9f_x$	$d8_x$	$d3_x$	$f3_x$	$8d_x$	$b1_x$
3	ff_x	$a7_x$	$3e_x$	dc_x	86_x	77_x	$d7_x$	$a6_x$	11_x	fb_x	$f4_x$	ba_x	92_x	91_x	64_x	83_x
4	$f1_x$	33_x	ef_x	da_x	$2c_x$	$b5_x$	$b2_x$	$2b_x$	88_x	$d1_x$	99_x	cb_x	$8c_x$	84_x	$1d_x$	14_x
5	81_x	97_x	71_x	ca_x	$5f_x$	$a3_x$	$8b_x$	57_x	$3c_x$	82_x	$c4_x$	52_x	$5c_x$	$1c_x$	$e8_x$	$a0_x$
6	04_x	$b4_x$	85_x	$4a_x$	$f6_x$	13_x	54_x	$b6_x$	df_x	$0c_x$	$1a_x$	$8e_x$	de_x	$e0_x$	39_x	fc_x
7	20_x	$9b_x$	24_x	$4e_x$	$a9_x$	98_x	$9e_x$	ab_x	$f2_x$	60_x	$d0_x$	$6c_x$	ea_x	fa_x	$c7_x$	$d9_x$
8	00_x	$d4_x$	$1f_x$	$6e_x$	43_x	bc_x	ec_x	53_x	89_x	fe_x	$7a_x$	$5d_x$	49_x	$c9_x$	32_x	$c2_x$
9	$f9_x$	$9a_x$	$f8_x$	$6d_x$	16_x	db_x	59_x	96_x	44_x	$e9_x$	cd_x	$e6_x$	46_x	42_x	$8f_x$	$0a_x$
10	$c1_x$	cc_x	$b9_x$	65_x	$b0_x$	$d2_x$	$c6_x$	ac_x	$1e_x$	41_x	62_x	29_x	$2e_x$	$0e_x$	74_x	50_x
11	02_x	$5a_x$	$c3_x$	25_x	$7b_x$	$8a_x$	$2a_x$	$5b_x$	$f0_x$	06_x	$0d_x$	47_x	$6f_x$	70_x	$9d_x$	$7e_x$
12	10_x	ce_x	12_x	27_x	$d5_x$	$4c_x$	$4f_x$	$d6_x$	79_x	30_x	68_x	36_x	75_x	$7d_x$	$e4_x$	ed_x
13	80_x	$6a_x$	90_x	37_x	$a2_x$	$5e_x$	76_x	aa_x	$c5_x$	$7f_x$	$3d_x$	af_x	$a5_x$	$e5_x$	19_x	61_x
14	fd_x	$4d_x$	$7c_x$	$b7_x$	$0b_x$	ee_x	ad_x	$4b_x$	22_x	$f5_x$	$e7_x$	73_x	23_x	21_x	$c8_x$	05_x
15	$e1_x$	66_x	dd_x	$b3_x$	58_x	69_x	63_x	56_x	$0f_x$	$a1_x$	31_x	95_x	17_x	07_x	$3a_x$	28_x

Table C.24 The 8×8 SHARK S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	$b1_x$	ce_x	$c3_x$	95_x	$5a_x$	ad_x	$e7_x$	02_x	$4d_x$	44_x	fb_x	91_x	$0c_x$	87_x	$a1_x$	50_x
1	cb_x	67_x	54_x	dd_x	46_x	$8f_x$	$e1_x$	$4e_x$	$f0_x$	fd_x	fc_x	eb_x	$f9_x$	$c4_x$	$1a_x$	$6e_x$
2	$5e_x$	$f5_x$	cc_x	$8d_x$	$1c_x$	56_x	43_x	fe_x	07_x	61_x	$f8_x$	75_x	59_x	ff_x	03_x	22_x
3	$8a_x$	$d1_x$	13_x	ee_x	88_x	00_x	$0e_x$	34_x	15_x	80_x	94_x	$e3_x$	ed_x	$b5_x$	53_x	23_x
4	$4b_x$	47_x	17_x	$a7_x$	90_x	35_x	ab_x	$d8_x$	$b8_x$	df_x	$4f_x$	57_x	$9a_x$	92_x	db_x	$1b_x$
5	$3c_x$	$c8_x$	99_x	04_x	$8e_x$	$e0_x$	$d7_x$	$7d_x$	85_x	bb_x	40_x	$2c_x$	$3a_x$	45_x	$f1_x$	42_x
6	65_x	20_x	41_x	18_x	72_x	25_x	93_x	70_x	36_x	05_x	$f2_x$	$0b_x$	$a3_x$	79_x	ec_x	08_x
7	27_x	31_x	32_x	$b6_x$	$7c_x$	$b0_x$	$0a_x$	73_x	$5b_x$	$7b_x$	$b7_x$	81_x	$d2_x$	$0d_x$	$6a_x$	26_x
8	$9e_x$	58_x	$9c_x$	83_x	74_x	$b3_x$	ac_x	30_x	$7a_x$	69_x	77_x	$0f_x$	ae_x	21_x	de_x	$d0_x$
9	$2e_x$	97_x	10_x	$a4_x$	98_x	$a8_x$	$d4_x$	68_x	$2d_x$	62_x	29_x	$6d_x$	16_x	49_x	76_x	$c7_x$
10	$e8_x$	$c1_x$	96_x	37_x	$e5_x$	ca_x	$f4_x$	$e9_x$	63_x	12_x	$c2_x$	$a6_x$	14_x	bc_x	$d3_x$	28_x
11	af_x	$2f_x$	$e6_x$	24_x	52_x	$c6_x$	$a0_x$	09_x	bd_x	$8c_x$	cf_x	$5d_x$	11_x	$5f_x$	01_x	$c5_x$
12	$9f_x$	$3d_x$	$a2_x$	$9b_x$	$c9_x$	$3b_x$	be_x	51_x	19_x	$1f_x$	$3f_x$	$5c_x$	$b2_x$	ef_x	$4a_x$	cd_x
13	bf_x	ba_x	$6f_x$	64_x	$d9_x$	$f3_x$	$3e_x$	$b4_x$	aa_x	dc_x	$d5_x$	06_x	$c0_x$	$7e_x$	$f6_x$	66_x
14	$6c_x$	84_x	71_x	38_x	$b9_x$	$1d_x$	$7f_x$	$9d_x$	48_x	$8b_x$	$2a_x$	da_x	$a5_x$	33_x	82_x	39_x
15	$d6_x$	78_x	86_x	fa_x	$e4_x$	$2b_x$	$a9_x$	$1e_x$	89_x	60_x	$6b_x$	ea_x	55_x	$4c_x$	$f7_x$	$e2_x$

Table C.25 The 8×8 SMS4 S-box. For $0 \le i, j \le 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	$d6_x$	90_x	$e9_x$	fe_x	cc_x	$e1_x$	$3d_x$	$b7_x$	16_x	$b6_x$	14_x	$c2_x$	28_x	fb_x	$2c_x$	05_x
1	$2b_x$	67_x	$9a_x$	76_x	$2a_x$	be_x	04_x	$c3_x$	aa_x	44_x	13_x	26_x	49_x	86_x	06_x	99_x
2	$9c_x$	42_x	50_x	$f4_x$	91_x	ef_x	98_x	$7a_x$	33_x	54_x	$0b_x$	43_x	ed_x	cf_x	ac_x	62_x
3	$e4_x$	$b3_x$	$1c_x$	$a9_x$	$c9_x$	08_x	$e8_x$	95_x	80_x	df_x	94_x	fa_x	75_x	$8f_x$	$3f_x$	$a6_x$
4	47_x	07_x	$a7_x$	fc_x	$f3_x$	73_x	17_x	ba_x	83_x	59_x	$3c_x$	19_x	$e6_x$	85_x	$4f_x$	$a8_x$
5	68_x	$6b_x$	81_x	$b2_x$	71_x	64_x	da_x	$8b_x$	$f8_x$	eb_x	$0f_x$	$4b_x$	70_x	56_x	$9d_x$	35_x
6	$1e_x$	24_x	$0e_x$	$5e_x$	63_x	58_x	$d1_x$	$a2_x$	25_x	22_x	$7c_x$	$3b_x$	01_x	21_x	78_x	87_x
7	$d4_x$	00_x	46_x	57_x	$9f_x$	$d3_x$	27_x	52_x	$4c_x$	36_x	02_x	$e7_x$	$a0_x$	$c4_x$	$c8_x$	$9e_x$
8	ea_x	bf_x	$8a_x$	$d2_x$	40_x	$c7_x$	38_x	$b5_x$	$a3_x$	$f7_x$	$f2_x$	ce_x	$f9_x$	61_x	15_x	$a1_x$
9	$e0_x$	ae_x	$5d_x$	$a4_x$	$9b_x$	34_x	$1a_x$	55_x	ad_x	93_x	32_x	30_x	$f5_x$	$8c_x$	$b1_x$	$e3_x$
10	$1d_x$	$f6_x$	$e2_x$	$2e_x$	82_x	66_x	ca_x	60_x	$c0_x$	29_x	23_x	ab_x	$0d_x$	53_x	$4e_x$	$6f_x$
11	$d5_x$	db_x	37_x	45_x	de_x	fd_x	$8e_x$	$2f_x$	03_x	ff_x	$6a_x$	72_x	$6d_x$	$6c_x$	$5b_x$	51_x
12	$8d_x$	$1b_x$	af_x	92_x	bb_x	dd_x	bc_x	$7f_x$	11_x	$d9_x$	$5c_x$	41_x	$1f_x$	10_x	$5a_x$	$d8_x$
13	$0a_x$	$c1_x$	31_x	88_x	$a5_x$	cd_x	$7b_x$	bd_x	$2d_x$	74_x	$d0_x$	12_x	$b8_x$	$e5_x$	$b4_x$	$b0_x$
14	89_x	69_x	97_x	$4a_x$	$0c_x$	96_x	77_x	$7e_x$	65_x	$b9_x$	$f1_x$	09_x	$c5_x$	$6e_x$	$c6_x$	84_x
15	18_x	$f0_x$	$7d_x$	ec_x	$3a_x$	dc_x	$4d_x$	20_x	79_x	ee_x	$5f_x$	$3e_x$	$d7_x$	cb_x	39_x	48_x

Table C.26 The 8×8 KHAZAD (tweaked) S-box. For $0 \le i, j \le 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	ba_x	54_x	$2f_x$	74_x	53_x	$d3_x$	$d2_x$	$4d_x$	50_x	ac_x	$8d_x$	bf_x	70_x	52_x	$9a_x$	$4c_x$
1	ea_x	$d5_x$	97_x	$d1_x$	33_x	51_x	$5b_x$	$a6_x$	de_x	48_x	$a8_x$	99_x	db_x	32_x	$b7_x$	fc_x
2	$e3_x$	$9e_x$	91_x	$9b_x$	$e2_x$	bb_x	41_x	$6e_x$	$a5_x$	cb_x	$6b_x$	95_x	$a1_x$	$f3_x$	$b1_x$	02_x
3	cc_x	$c4_x$	$1d_x$	14_x	$c3_x$	63_x	da_x	$5d_x$	$5f_x$	dc_x	$7d_x$	cd_x	$7f_x$	$5a_x$	$6c_x$	$5c_x$
4	$f7_x$	26_x	ff_x	ed_x	$e8_x$	$9d_x$	$6f_x$	$8e_x$	19_x	$a0_x$	$f0_x$	89_x	$0f_x$	07_x	af_x	fb_x
5	08_x	15_x	$0d_x$	04_x	01_x	64_x	df_x	76_x	79_x	dd_x	$3d_x$	16_x	$3f_x$	37_x	$6d_x$	38_x
6	$b9_x$	73_x	$e9_x$	35_x	55_x	71_x	$7b_x$	$8c_x$	72_x	88_x	$f6_x$	$2a_x$	$3e_x$	$5e_x$	27_x	46_x
7	$0c_x$	65_x	68_x	61_x	03_x	$c1_x$	57_x	$d6_x$	$d9_x$	58_x	$d8_x$	66_x	$d7_x$	$3a_x$	$c8_x$	$3c_x$
8	fa_x	96_x	$a7_x$	98_x	ec_x	$b8_x$	$c7_x$	ae_x	69_x	$4b_x$	ab_x	$a9_x$	67_x	$0a_x$	47_x	$f2_x$
9	$b5_x$	22_x	$e5_x$	ee_x	be_x	$2b_x$	81_x	12_x	83_x	$1b_x$	$0e_x$	23_x	$f5_x$	45_x	21_x	ce_x
10	49_x	$2c_x$	$f9_x$	$e6_x$	$b6_x$	28_x	17_x	82_x	$1a_x$	$8b_x$	fe_x	$8a_x$	09_x	$c9_x$	87_x	$4e_x$
11	$e1_x$	$2e_x$	$e4_x$	$e0_x$	eb_x	90_x	$a4_x$	$1e_x$	85_x	60_x	00_x	25_x	$f4_x$	$f1_x$	94_x	$0b_x$
12	$e7_x$	75_x	ef_x	34_x	31_x	$d4_x$	$d0_x$	86_x	$7e_x$	ad_x	fd_x	29_x	30_x	$3b_x$	$9f_x$	$f8_x$
13	$c6_x$	13_x	06_x	05_x	$c5_x$	11_x	77_x	$7c_x$	$7a_x$	78_x	36_x	$1c_x$	39_x	59_x	18_x	56_x
14	$b3_x$	$b0_x$	24_x	20_x	$b2_x$	92_x	$a3_x$	$c0_x$	44_x	62_x	10_x	$b4_x$	84_x	43_x	93_x	$c2_x$
15	$4a_x$	bd_x	$8f_x$	$2d_x$	bc_x	$9c_x$	$6a_x$	40_x	cf_x	$a2_x$	80_x	$4f_x$	$1f_x$	ca_x	aa_x	42_x

Table C.27 The 8×8 RAINBOW S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	00_x	$0e_x$	$1c_x$	08_x	38_x	$e5_x$	10_x	19_x	70_x	16_x	cb_x	42_x	20_x	$e7_x$	32_x	$d4_x$
1	$e0_x$	cc_x	$2c_x$	65_x	97_x	$a7_x$	84_x	$1f_x$	40_x	67_x	cf_x	78_x	64_x	$2d_x$	$a9_x$	be_x
2	$c1_x$	$c2_x$	99_x	ec_x	58_x	$d1_x$	ca_x	fb_x	$2f_x$	$8e_x$	$4f_x$	$6d_x$	09_x	50_x	$3e_x$	$2a_x$
3	80_x	56_x	ce_x	11_x	$9f_x$	$0c_x$	$f0_x$	$a4_x$	$c8_x$	df_x	$5a_x$	$b1_x$	53_x	73_x	$7d_x$	$6f_x$
4	83_x	79_x	85_x	$f9_x$	33_x	$e9_x$	$d9_x$	$4b_x$	$b0_x$	74_x	$a3_x$	14_x	95_x	03_x	$f7_x$	dc_x
5	$5e_x$	$7a_x$	$1d_x$	$c0_x$	$9e_x$	55_x	da_x	26_x	12_x	$6b_x$	$a0_x$	$d5_x$	$7c_x$	98_x	54_x	72_x
6	01_x	48_x	ac_x	$0f_x$	$9d_x$	ad_x	22_x	36_x	$3f_x$	82_x	18_x	ba_x	$e1_x$	57_x	49_x	$2e_x$
7	91_x	$f1_x$	bf_x	$4a_x$	$b4_x$	62_x	63_x	ee_x	$a6_x$	51_x	$e6_x$	71_x	fa_x	$c9_x$	de_x	43_x
8	07_x	04_x	$f2_x$	$8c_x$	$0b_x$	21_x	$f3_x$	$6a_x$	66_x	$b2_x$	$d3_x$	$8f_x$	$b3_x$	$3c_x$	96_x	$5f_x$
9	61_x	76_x	$e8_x$	fd_x	47_x	$b6_x$	28_x	15_x	$2b_x$	88_x	06_x	52_x	ef_x	$d8_x$	$b9_x$	$b7_x$
10	bc_x	fc_x	$f4_x$	$a5_x$	$3a_x$	$0a_x$	81_x	$6e_x$	$3d_x$	60_x	aa_x	13_x	$b5_x$	ea_x	$4c_x$	39_x
11	24_x	87_x	$d6_x$	$1b_x$	41_x	$5d_x$	ab_x	17_x	$f8_x$	25_x	31_x	77_x	$a8_x$	$b8_x$	$e4_x$	$a1_x$
12	02_x	46_x	90_x	35_x	59_x	$c7_x$	$1e_x$	af_x	$3b_x$	fe_x	$5b_x$	$8a_x$	44_x	29_x	$6c_x$	db_x
13	$7e_x$	$d2_x$	05_x	37_x	30_x	89_x	75_x	$9c_x$	$c3_x$	$8d_x$	ae_x	$8b_x$	92_x	bb_x	$5c_x$	$d0_x$
14	23_x	$9a_x$	$e3_x$	$d7_x$	$7f_x$	45_x	94_x	ed_x	69_x	$9b_x$	$c4_x$	$4e_x$	$c6_x$	$c5_x$	dd_x	68_x
15	$4d_x$	eb_x	$a2_x$	$f6_x$	cd_x	27_x	$e2_x$	34_x	$f5_x$	$7b_x$	93_x	$1a_x$	bd_x	$0d_x$	86_x	ff_x

Table C.28 The 8×8 WHIRLPOOL S-box. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	18_x	23_x	$c6_x$	$E8_x$	87_x	$B8_x$	01_x	$4F_x$	36_x	$A6_x$	$d2_x$	$F5_x$	79_x	$6F_x$	91_x	52_x
1	60_x	Bc_x	$9B_x$	$8E_x$	$A3_x$	$0c_x$	$7B_x$	35_x	$1d_x$	$E0_x$	$d7_x$	$c2_x$	$2E_x$	$4B_x$	FE_x	57_x
2	15_x	77_x	37_x	$E5_x$	$9F_x$	$F0_x$	$4A_x$	dA_x	58_x	$c9_x$	29_x	$0A_x$	$B1_x$	$A0_x$	$6B_x$	85_x
3	Bd_x	$5d_x$	10_x	$F4_x$	cB_x	$3E_x$	05_x	67_x	$E4_x$	27_x	41_x	$8B_x$	$A7_x$	$7d_x$	95_x	$d8_x$
4	FB_x	EE_x	$7c_x$	66_x	dd_x	17_x	47_x	$9E_x$	cA_x	$2d_x$	BF_x	07_x	Ad_x	$5A_x$	83_x	33_x
5	63_x	02_x	AA_x	71_x	$c8_x$	19_x	49_x	$d9_x$	$F2_x$	$E3_x$	$5B_x$	88_x	$9A_x$	26_x	32_x	$B0_x$
6	$E9_x$	$0F_x$	$d5_x$	80_x	BE_x	cd_x	34_x	48_x	FF_x	$7A_x$	90_x	$5F_x$	20_x	68_x	$1A_x$	AE_x
7	$B4_x$	54_x	93_x	22_x	64_x	$F1_x$	73_x	12_x	40_x	08_x	$c3_x$	Ec_x	dB_x	$A1_x$	$8d_x$	$3d_x$
8	97_x	00_x	cF_x	$2B_x$	76_x	82_x	$d6_x$	$1B_x$	$B5_x$	AF_x	$6A_x$	50_x	45_x	$F3_x$	30_x	EF_x
9	$3F_x$	55_x	$A2_x$	EA_x	65_x	BA_x	$2F_x$	$c0_x$	dE_x	$1c_x$	Fd_x	$4d_x$	92_x	75_x	06_x	$8A_x$
10	$B2_x$	$E6_x$	$0E_x$	$1F_x$	62_x	$d4_x$	$A8_x$	96_x	$F9_x$	$c5_x$	25_x	59_x	84_x	72_x	39_x	$4c_x$
11	$5E_x$	78_x	38_x	$8c_x$	$d1_x$	$A5_x$	$E2_x$	61_x	$B3_x$	21_x	$9c_x$	$1E_x$	43_x	$c7_x$	Fc_x	04_x
12	51_x	99_x	$6d_x$	$0d_x$	FA_x	dF_x	$7E_x$	24_x	$3B_x$	AB_x	cE_x	11_x	$8F_x$	$4E_x$	$B7_x$	EB_x
13	$3c_x$	81_x	94_x	$F7_x$	$B9_x$	13_x	$2c_x$	$d3_x$	$E7_x$	$6E_x$	$c4_x$	03_x	56_x	44_x	$7F_x$	$A9_x$
14	$2A_x$	BB_x	$c1_x$	53_x	dc_x	$0B_x$	$9d_x$	$6c_x$	31_x	74_x	$F6_x$	46_x	Ac_x	89_x	14_x	$E1_x$
15	16_x	$3A_x$	69_x	09_x	70_x	$B6_x$	$d0_x$	Ed_x	cc_x	42_x	98_x	$A4_x$	28_x	$5c_x$	$F8_x$	86_x

Table C.29 The 8×8 ZODIAC S-box 1. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	$2d_x$	$f3_x$	$7c_x$	$6d_x$	$9d_x$	$b5_x$	26_x	74_x	$f2_x$	93_x	53_x	$b0_x$	$f0_x$	11_x	ed_x	83_x
1	78_x	$b6_x$	03_x	16_x	73_x	$3b_x$	$1e_x$	$8e_x$	70_x	bd_x	86_x	$1b_x$	47_x	$7e_x$	24_x	56_x
2	$f1_x$	77_x	88_x	46_x	97_x	$b1_x$	ba_x	$a3_x$	$b7_x$	10_x	$0a_x$	$c5_x$	37_x	$b3_x$	$c9_x$	$5a_x$
3	28_x	ac_x	64_x	$a5_x$	ec_x	ab_x	aa_x	$c6_x$	67_x	95_x	58_x	$0d_x$	$f8_x$	$9a_x$	$f6_x$	$6e_x$
4	66_x	dc_x	05_x	$3d_x$	$d3_x$	$8a_x$	$c3_x$	$d8_x$	89_x	$6a_x$	$e9_x$	36_x	49_x	43_x	bf_x	eb_x
5	$d4_x$	96_x	$9b_x$	68_x	$a0_x$	65_x	$5d_x$	57_x	92_x	$1f_x$	$d5_x$	71_x	$5c_x$	bb_x	22_x	$c1_x$
6	be_x	$7b_x$	bc_x	99_x	63_x	94_x	$5f_x$	$2a_x$	61_x	$b8_x$	34_x	32_x	19_x	fd_x	fb_x	17_x
7	40_x	$e6_x$	51_x	$1d_x$	41_x	44_x	$8f_x$	29_x	dd_x	04_x	80_x	de_x	$e7_x$	31_x	$d6_x$	$7f_x$
8	01_x	$a2_x$	$f7_x$	39_x	da_x	$6f_x$	23_x	ca_x	fe_x	$3a_x$	$d0_x$	$1c_x$	$d1_x$	30_x	$3e_x$	12_x
9	$a1_x$	cd_x	$0f_x$	$e0_x$	$a8_x$	af_x	82_x	59_x	$2c_x$	$f5_x$	$7d_x$	ad_x	$b2_x$	ef_x	$c2_x$	87_x
10	ce_x	75_x	06_x	13_x	02_x	90_x	$4f_x$	$2e_x$	72_x	33_x	85_x	$c0_x$	$8d_x$	cf_x	$a9_x$	81_x
11	$e2_x$	$c4_x$	27_x	$2f_x$	$6c_x$	$7a_x$	$9f_x$	52_x	$e1_x$	15_x	38_x	$2b_x$	fc_x	20_x	42_x	$c7_x$
12	08_x	$e4_x$	09_x	55_x	$5e_x$	$8c_x$	14_x	76_x	60_x	ff_x	df_x	$d7_x$	98_x	fa_x	$0b_x$	21_x
13	00_x	$1a_x$	$f9_x$	$a6_x$	$b9_x$	$e8_x$	$9e_x$	62_x	$4c_x$	$d9_x$	91_x	50_x	$d2_x$	ee_x	18_x	$b4_x$
14	07_x	84_x	ea_x	$5b_x$	$a4_x$	$c8_x$	$0e_x$	cb_x	48_x	69_x	$4b_x$	$4e_x$	$9c_x$	35_x	79_x	45_x
15	$4d_x$	54_x	$e5_x$	25_x	$3c_x$	$0c_x$	$4a_x$	$8b_x$	$3f_x$	cc_x	$a7_x$	db_x	$6b_x$	ae_x	$f4_x$	$e3_x$

Table C.30 The 8×8 ZODIAC S-box 2. For $0 \leq i, j \leq 15$, the $S[(i \ll 4)|j]$ entry is in the i-th row and j-th column. Values in hexadecimal.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	00_x	$4a_x$	ce_x	$e7_x$	$d2_x$	62_x	$0c_x$	$e0_x$	$1f_x$	ef_x	11_x	75_x	78_x	71_x	$a5_x$	$8e_x$
1	76_x	$3d_x$	bd_x	bc_x	86_x	57_x	$0b_x$	28_x	$2f_x$	$a3_x$	da_x	$d4_x$	$e4_x$	$0f_x$	$a9_x$	27_x
2	53_x	04_x	$1b_x$	fc_x	ac_x	$e6_x$	$7a_x$	07_x	ae_x	63_x	$c5_x$	db_x	$e2_x$	ea_x	94_x	$8b_x$
3	$c4_x$	$d5_x$	$9d_x$	$f8_x$	90_x	$6b_x$	$b1_x$	$0d_x$	$d6_x$	eb_x	$c6_x$	$0e_x$	cf_x	ad_x	08_x	$4e_x$
4	$d7_x$	$e3_x$	$5d_x$	50_x	$1e_x$	$b3_x$	$5b_x$	23_x	38_x	34_x	68_x	46_x	03_x	$8c_x$	dd_x	$9c_x$
5	$7d_x$	$a0_x$	cd_x	$1a_x$	41_x	01_x	01_x	$8d_x$	$f6_x$	cb_x	52_x	$7b_x$	$d1_x$	$e8_x$	$4f_x$	29_x
6	$c0_x$	$b0_x$	$e1_x$	$e5_x$	$c7_x$	74_x	$b4_x$	aa_x	$4b_x$	99_x	$2b_x$	60_x	$5f_x$	58_x	$3f_x$	fd_x
7	cc_x	ff_x	40_x	ee_x	$b2_x$	$3a_x$	$6e_x$	$5a_x$	$f1_x$	55_x	$4d_x$	$a8_x$	$c9_x$	$c1_x$	$0a_x$	98_x
8	15_x	30_x	44_x	$a2_x$	$c2_x$	$2c_x$	45_x	92_x	$6c_x$	$f3_x$	39_x	66_x	42_x	$f2_x$	35_x	20_x
9	$6f_x$	77_x	bb_x	59_x	19_x	$1d_x$	fe_x	37_x	67_x	$2d_x$	31_x	$f5_x$	69_x	$a7_x$	64_x	ab_x
10	13_x	54_x	25_x	$e9_x$	09_x	ed_x	$5c_x$	05_x	ca_x	$4c_x$	24_x	87_x	bf_x	18_x	$3e_x$	22_x
11	$f0_x$	51_x	ec_x	61_x	17_x	16_x	$5e_x$	af_x	$d3_x$	49_x	$a6_x$	36_x	43_x	$f4_x$	47_x	91_x
12	df_x	33_x	93_x	21_x	$3b_x$	79_x	$b7_x$	97_x	85_x	10_x	$b5_x$	ba_x	$3c_x$	$b6_x$	70_x	$d0_x$
13	06_x	$a1_x$	fa_x	81_x	82_x	83_x	$7e_x$	$7f_x$	80_x	96_x	73_x	be_x	56_x	$9b_x$	$9e_x$	95_x
14	$d9_x$	$f7_x$	02_x	$b9_x$	$a4_x$	de_x	$6a_x$	32_x	$6d_x$	$d8_x$	$8a_x$	84_x	72_x	$2a_x$	14_x	$9f_x$
15	88_x	$f9_x$	dc_x	89_x	$9a_x$	fb_x	$7c_x$	$2e_x$	$c3_x$	$8f_x$	$b8_x$	65_x	48_x	26_x	$c8_x$	12_x

References

1. 3GPP: 3GPP TS 35.202 V5.0.0 Third Generation Partnership Project. Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms. Document 2: KASUMI Specification (Release 5) (2002)
2. Adams, C.M.: The CAST-256 Encryption Algorithm. First AES Conference, USA (1998)
3. Anderson, R.J., Biham, E., Knudsen, L.R.: Serpent and Smartcards. In: J.J. Quisquater, B. Schneier (eds.) Smart Card Research and Application Conference (CARDIS), LNCS 2000, pp. 246–253. Springer (1998)
4. Barreto, P.S.L.M., Nikov, V., Nikova, S., Rijmen, V., Tischhauser, E.: Whirlwind: a New Cryptographic Hash Function. Designs, Codes and Cryptography **56**, 141–162 (2010)
5. Barreto, P.S.L.M., Rijmen, V.: The ANUBIS Block Cipher. First NESSIE Workshop, Heverlee, Belgium (2000)
6. Barreto, P.S.L.M., Rijmen, V.: The KHAZAD Legacy-Level Block Cipher. First NESSIE Workshop, Heverlee, Belgium (2000)
7. Barreto, P.S.L.M., Rijmen, V.: The Whirlpool Hash Function. Submission to NESSIE (2003)
8. Barreto, P.S.L.M., Simplicio Jr, M.: Curupira, a Block Cipher for Constrained Platforms. 25th Brazilian Symposium on Computer Networks and Distributed Systems (2007)
9. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: RadioGatún, a Belt-and-Mill Hash Function. Second Cryptographic Hash Function Workshop (2006)
10. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak Reference. http://keccak.noekeon.org (2011)
11. Biham, E., Anderson, R.J., Knudsen, L.R.: Serpent: a New Block Cipher Proposal. In: S. Vaudenay (ed.) Fast Software Encryption (FSE), LNCS 1372, pp. 222–238. Springer (1998)
12. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer (1993)
13. Bilgin, B., Nikova, S., Rijmen, V., Nikov, V., Stutz, G.: Threshold Implementations of All 3×3 and 4×4 S-boxes. IACR ePrint archive 2012/300 (2012)
14. Biryukov, A., Shamir, A.: Structural Cryptanalysis of SASAS. In: B. Pfitzmann (ed.) Advances in Cryptology, Eurocrypt, LNCS 2045, pp. 394–405. Springer (2001)
15. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: Present: an Utra-Lightweight Block Cipher. In: P. Paillier, I. Verbauwhede (eds.) Cryptographic Hardware and Embedded Systems (CHES), LNCS 4727, pp. 450–466. Springer (2007)
16. Borst, J.: The Block Cipher: GRAND CRU. First NESSIE Workshop, Heverlee, Belgium (2000)
17. Brown, L., Kwan, M., Pieprzyk, J., Seberry, J.: Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI. In: H. Imai, R.L. Rivest, T. Matsumoto (eds.) Advances in Cryptology, Asiacrypt, LNCS 739, pp. 36–50. Springer (1991)
18. Brown, L., Pieprzyk, J.: Introducing the New LOKI97 Block Cipher. First AES Conference, California, USA (1998)
19. Brown, L., Pieprzyk, J., Seberry, J.: LOKI - a Cryptographic Primitive for Authentication and Secrecy Applications. In: J. Seberry, J. Pieprzyk (eds.) Advances in Cryptology, Auscrypt, LNCS 453, pp. 229–236. Springer (1990)
20. Burwick, C., Coppersmith, D., D'Avignon, E., Genario, R., Halevi, S., Jutla, C., Matyas Jr, S.M., O'Connor, L., Peyravian, M., Safford, D., Zunic, N.: MARS – a Candidate Cipher for AES. First AES Conference, California, USA (1998)

21. Courtois, N., Pieprzyk, J.: Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In: Y. Zheng (ed.) Advances in Cryptology, Asiacrypt, LNCS 2501, pp. 267–287. Springer (2002)
22. Courtois, N.T.: How Fast can be Algebraic Attacks on Block Ciphers? IACR ePrint archive, 2006/168 (2006)
23. Daemen, J.: Cipher and Hash Function Design – Strategies based on Linear and Differential Cryptanalysis. Ph.D. thesis, Dept. Elektrotechniek, ESAT, Katholieke Universiteit Leuven, Belgium (1995)
24. Daemen, J., Govaerts, R., Vandewalle, J.: A New Approach to Block Cipher Design. In: R. Anderson (ed.) Fast Software Encryption (FSE), LNCS 809, pp. 18–32. Springer (1993)
25. Daemen, J., Knudsen, L.R., Rijmen, V.: The Block Cipher SQUARE. In: E. Biham (ed.) Fast Software Encryption (FSE), LNCS 1267, pp. 149–165. Springer (1997)
26. Daemen, J., Peeters, M., Van Assche, G., Rijmen, V.: NESSIE Proposal: NOEKEON. First NESSIE Workshop, Heverlee, Belgium (2000)
27. Daemen, J., Rijmen, V.: The Block Cipher BKSQ. In: J.J. Quisquater, B. Schneier (eds.) Smart Card Research and Applications (CARDIS), LNCS 1820, pp. 236–245. Springer (2000)
28. Davies, D., Murphy, S.: Pairs and Triplets of DES S-boxes. Journal of Cryptology **8**(1), 1–25 (1993)
29. Dawson, E., Chen, K., Henricksen, M., Millan, W., Simpson, L., Lee, H., Moon, S.J.: Dragon: a Fast Word-Based Stream Cipher. Submission to eStream Project (2006)
30. Diffie, W., Ledin, G.: SMS4 Encryption Algorithm for Wireless Networks, version 10.3 (2008)
31. ETSI: SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 and 128-EIA3. Document 1: 128-EEA3 and 128-EIA3 specification. ver. 1.6 (2011)
32. Filho, G.D., Barreto, P.S.L.M., Rijmen, V.: The Maelstrom-0 Hash Function. Proceedings of the 6th Brazilian Symposium on Information and Computer Systems Security (2006)
33. FIPS197: Advanced Encryption Standard (AES). FIPS PUB 197 Federal Information Processing Standard Publication 197, United States Department of Commerce (2001)
34. Gauravaram, P., Knudsen, L.R., Matusiewicz, K., Mendel, F., Rechberger, C., Schlaffer, M., Thomsen, S.: Grostl, a SHA-3 Candidate. Submission to NIST, Secure Hash Standard 3 (2008)
35. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.J.B.: The LED Block Cipher. In: B. Preneel, T. Takagi (eds.) Cryptographic Hardware and Embedded Systems (CHES), LNCS 6917, pp. 326–341. Springer (2011)
36. Gupta, K.C., Ray, I.G.: On Constructions of Involutory MDS Matrices. In: A.M. Youssef, A. Nitaj, A.E. Hassanien (eds.) AfricaCrypt, LNCS 7918, pp. 43–60. Springer (2013)
37. Isobe, T.: A Single-Key Attack on the Full GOST Block Cipher. In: A. Joux (ed.) Fast Software Encryption (FSE), LNCS 6733, pp. 290–305. Springer (2011)
38. Jacobson Jr, M.J., Huber, K.: The MAGENTA Block Cipher Algorithm. First AES Conference, California, USA (1998)
39. Junod, P., Macchetti, M.: Revisiting the IDEA Philosophy. In: O. Dunkelman (ed.) Fast Software Encryption (FSE), LNCS 5665, pp. 277–295. Springer (2009)
40. Junod, P., Vaudenay, S.: FOX: a New Family of Block Ciphers. In: H. Handschuh, M.A. Hasan (eds.) Selected Areas in Cryptography (SAC), LNCS 3357, pp. 114–129. Springer (2004)
41. Junod, P., Vaudenay, S.: Perfect Diffusion Primitives for Block Ciphers Building Efficient MDS Matrices. In: H. Handschuh, M.A. Hasan (eds.) Selected Areas in Cryptography (SAC), LNCS 3357, pp. 84–99. Springer (2004)

42. Kim, K.: Construction of DES-like S-boxes based on Boolean Functions Satisfying the SAC. In: H. Imai, R.L. Rivest, T. Matsumoto (eds.) Advances in Cryptology, Asiacrypt, LNCS 739, pp. 59–72. Springer (1991)
43. Kim, K., Lee, S., Park, S., Lee, D.: How to Strengthen DES Against Two Robust Attacks. Proceedings of 1995 Korea–Japan Joint Workshop on Info. Security and Cryptology, JW-ISC'95 (1995)
44. Kim, K., Park, S., Lee, S.: Reconstruction of s^2-DES S-boxes and Their Immunity to Differential Cryptanalysis. JW-ISC, 1993 Korean-Japan Joint Workshop on Information Security and Cryptology (1993)
45. Knudsen, L.R., Leander, G., Poschmann, A., Robsaw, M.J.B.: PRINTcipher: a Block Cipher for IC-Printing. In: S. Mangard, F.X. Standaert (eds.) Cryptographic Hardware and Embedded Systems (CHES), LNCS 6225, pp. 16–32. Springer (2010)
46. Küçük, O.: The Hash Function Hamsi. Submission to NIST, SHA-3 Competition (2009)
47. Kwon, D., Kim, J., Park, S., Sung, S.H., Sohn, Y., Song, J.H., Yeom, Y., Yoon, E.J., Lee, S., Lee, J., Chee, S., Han, D., Hong, J.: New Block Cipher: ARIA. In: J.I. Lim, D.H. Lee (eds.) Information Security and Cryptology, ICISC, LNCS 2971, pp. 432–445. Springer (2003)
48. Lacan, J., Fimes, J.: Systematic MDS Erasure Codes based on Vandermonde Matrices. IEEE Transactions on Communications Letters **8**(9), 570–572 (2004)
49. Lai, X., Massey, J.L.: A Proposal for a New Block Encryption Standard. In: I.B. Damgård (ed.) Advances in Cryptology, Eurocrypt, LNCS 473, pp. 389–404. Springer (1990)
50. Lai, X., Massey, J.L., Murphy, S.: Markov Ciphers and Differential Cryptanalysis. In: D.W. Davies (ed.) Advances in Cryptology, Eurocrypt, LNCS 547, pp. 17–38. Springer (1991)
51. Lee, C., Jun, K., Jung, M., Park, S., Kim, J.: Zodiac version 1.0 (revised) Architecture and Specification. Standardization Workshop on Information Security Technology, Korean contribution on MP18033, ISO/IEC JTC1/SC27 N2563 (2000)
52. Lee, C.H., Kim, J.S.: The New Block Cipher Rainbow. Samsung Advanced Institute of Technology (1997)
53. Lee, H.J., Lee, S.J., Yoon, J.H., Cheon, D.H., Lee, J.I.: The SEED Encryption Algorithm. RFC 4269 (2005)
54. Lim, C.H.: A Revised Version of Crypton - CRYPTON version 1.0. In: L.R. Knudsen (ed.) Fast Software Encryption (FSE), LNCS 1636, p. 31. Springer (1999)
55. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland Mathematical Library. North-Holland Publishing Co. (1977)
56. Massey, J.L.: SAFER K–64: a Byte-Oriented Block-Ciphering Algorithm. In: R. Anderson (ed.) Fast Software Encryption (FSE), LNCS 809, pp. 1–17. Springer (1994)
57. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: T. Helleseth (ed.) Advances in Cryptology, Eurocrypt, LNCS 765, pp. 386–397. Springer (1993)
58. Matsui, M.: On Correlation between the Order of S-boxes and the Strength of DES. In: A. De Santis (ed.) Advances in Cryptology, Eurocrypt, LNCS 950, pp. 366–375. Springer (1994)
59. Matsui, M.: Block Encryption Algorithm MISTY. Technical report of IEICE, isec96-11, in Japanese, Mitsubishi Co. (1996)
60. Matsui, M.: New Block Encryption Algorithm MISTY. In: E. Biham (ed.) Fast Software Encryption (FSE), LNCS 1267, pp. 54–68. Springer (1997)
61. Matsui, M., Nakajima, J., Moriai, S.: A Description of the Camellia Encryption Algorithm. Request for comments, RFC3713 (2004)
62. Menezes, A.J., Van Oorschot, P., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1997)
63. Merkle, R.C.: Fast Software Encryption Functions. In: A.J. Menezes, S.A. Vanstone (eds.) Advances in Cryptology, Crypto, LNCS 537, pp. 476–501. Springer (1991)

64. Mileva, A.: Multipermutations in Crypto World: Different Faces of the Same Perfect Diffusion Layer. IACR ePrint, 2014/085 (2014)
65. Nakahara Jr, J.: A Linear Analysis of Blowfish and Khufu. In: E. Dawson, S. Duncan (eds.) Information Security Practice and Experience Conference (ISPEC), LNCS 4464, pp. 20–32. Springer (2007)
66. Nakahara Jr, J.: Analysis of Venkaiah et al.'s AES Design. International Journal of Network Security (IJNS) **9**, 285–289 (2009)
67. Nakahara Jr, J., Rijmen, V., Preneel, B., Vandewalle, J.: The MESH Block Ciphers. In: K. Chae, M. Yung (eds.) Information Security Applications (WISA), LNCS 2908, pp. 458–473. Springer (2003)
68. NBS: Data Encryption Standard (DES). FIPS PUB 46–3, Federal Information Processing Standards Publication 46–3 (1999)
69. NIST: Skipjack and KEA Specification, version 2.0. http://csrc.nist.gov/ (1998)
70. NTT: Specification of E2 – a 128-bit Block Cipher. First AES Conference, California, USA (1998)
71. Nyberg, K.: Differentially Uniform Mappings for Cryptography. In: T. Helleseth (ed.) Advances in Cryptology, Eurocrypt, LNCS 765, pp. 55–64. Springer (1993)
72. Pommerening, K.: Fourier Analysis of Boolean Maps – a Tutorial. available at http://www.staff.uni-mainz.de/pommeren/Kryptologie (2005)
73. Preneel, B., Leekwijck, W.V., Linden, L.V., Govaerts, R., Vandewalle, J.: Propagation Characteristics of Boolean Functions. In: I.B. Damgaard (ed.) Advances in Cryptology, Eurocrypt, LNCS 473, pp. 161–173. Springer (1990)
74. Rijmen, V., Daemen, J., Preneel, B., Bosselaers, A., De Win, E.: The Cipher SHARK. In: D. Gollmann (ed.) Fast Software Encryption (FSE), LNCS 1039, pp. 99–112. Springer (1996)
75. Rijmen, V., Preneel, B., De Win, E.: On Weaknesses of Non-Surjective Round Functions. Designs, Codes and Cryptography **12**(3), 253–266 (1997)
76. Saarinen, M.J.O.: Cryptographic Analysis of All 4×4-bit S-boxes. In: A. Miri, S. Vaudenay (eds.) Selected Areas in Cryptography (SAC), LNCS 7118, pp. 118–133. Springer (2011)
77. Sajadieh, M., Dakhilalian, M., Mala, H., Omoomi, B.: On Construction of Involutory MDS Matrices from Vandermonde Matrices in GF(2^q). Designs, Codes and Cryptography **64**(3), 287–308 (2012)
78. Schneier, B.: Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish). In: R. Anderson (ed.) Fast Software Encryption (FSE), LNCS 809, pp. 191–204. Springer (1994)
79. Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., Ferguson, N.: Twofish: a 128-bit Block Cipher. First AES Conference, California, USA (1998)
80. Schnorr, C.P., Vaudenay, S.: Black-Box Cryptanalysis of Hash Networks based on Multipermutations. In: A. De Santis (ed.) Advances in Cryptology, Eurocrypt, LNCS 950, pp. 47–57. Springer (1995)
81. Shannon, C.E.: Communication Theory of Secrecy Systems. Bell System Technical Journal **28**(4), 656–715 (1949)
82. Shimoyama, T., Kaneko, T.: Quadratic Relation of S-box and its Application to the Linear Attack of Full Round DES. In: H. Krawczyk (ed.) Advances in Cryptology, Crypto, LNCS 1462, pp. 200–211. Springer (1998)
83. Shimoyama, T., Yanami, H., Yokoyama, K., Takenaka, M., Itoh, K., Yajima, J., Torii, N., Tanaka, H.: The Block Cipher SC2000. In: M. Matsui (ed.) Fast Software Encryption (FSE), LNCS 2355, pp. 312–327. Springer (2002)
84. Shirai, T., Shibutani, K.: On the Diffusion Matrix Employed in the Whirlpool Hashing Function. The NESSIE project (2003)
85. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-bit Blockcipher CLEFIA (extended abstract). In: A. Biryukov (ed.) Fast Software Encryption (FSE), LNCS 4593, pp. 181–195. Springer (2007)

86. Sony: CLEFIA. https://www.sony.net/Products/cryptography/clefia (2007)
87. Stern, J., Vaudenay, S.: CS-Cipher. In: S. Vaudenay (ed.) Fast Software Encryption (FSE), LNCS 1372, pp. 189–205. Springer (1998)
88. Toshiba: Specification of Hierocrypt-3. First NESSIE Workshop, Heverlee, Belgium (2000)
89. Toshiba: Specification of Hierocrypt-L1. First NESSIE Workshop, Heverlee, Belgium (2000)
90. Tsunoo, Y., Kubo, H., Miyauchi, H., Nakamura, K.: A Secure Cipher Evaluated by Statistical Methods. SCIS'98-4.2.B, The 1998 Symposium on Cryptography and Information Security, p.28–31, The Institute of Electronics, Information and Communication Engineers (1998)
91. Tsunoo, Y., Kubo, H., Miyauchi, H., Nakamura, K.: A New 128-bit Block Cipher CIPHERUNICORN-A. The Institute of Electronics, Information and Communication Engineers, Technical Report of IEICE, ISEC2000-5 (2000)
92. Vaudenay, S.: On the Need for Multipermutations: Cryptanalysis of MD4 and SAFER. In: B. Preneel (ed.) Fast Software Encryption (FSE), LNCS 1008, pp. 286–297. Springer (1995)
93. Vaudenay, S.: Provable Security for Block Cipher by Decorrelation. In: M. Morvan, C. Meinel, D. Krob (eds.) Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science (STACS), LNCS 1373, pp. 249–275. Springer (1998)
94. Watanabe, D., Furuya, S., Yoshida, H., Takaragi, K., Preneel, B.: A New Keystream Generator MUGI. In: J. Daemen, V. Rijmen (eds.) Fast Software Encryption (FSE), LNCS 2365, pp. 179–194. Springer (2002)
95. Webster, A.F., Tavares, S.E.: On the Design of S-boxes. In: H.C. Williams (ed.) Advances in Cryptology, Crypto, LNCS 219, pp. 523–534. Springer (1985)
96. Wu, H.: The Hash Function JH. Submission to NIST, SHA-3 competition (2011)
97. Youssef, A.M., Mister, S., Tavares, S.E.: On the Design of Linear Transformations for Substitution Permutation Encryption Networks. In: C. Adams, M. Just (eds.) Selected Areas in Cryptography (SAC), pp. 40–48. Springer (1997)
98. Zabotin, I.A., Glazkov, G.P., Isaeva, V.B.: GOST 28147-89, Cryptographic Protection for Data Processing Systems, Cryptographic Transformation Algorithm. Government Standard of the U.S.S.R., Inv. No. 3583, UDC 681.325.6:006.354 (1989)

Index

J. Nakahara Jr., *Lai-Massey Cipher Designs*, https://doi.org/10.1007/978-3-319-68273-0