# Cryptanalysis of the "Augmented Family of Cryptographic Parity Circuits" proposed at ISW'97

A.M. Youssef

Center for Applied Cryptographic Research
Department of Combinatorics and Optimization
University of Waterloo, Waterloo, ON N2L 3G1
{a2youssef@cacr.math.uwaterloo.ca

**Abstract.** At Crypto'90, Koyama and Terada proposed a family of cryptographic functions for application to symmetric block ciphers. Youssef and Tavares showed that this family is affine and hence it is completely insecure. In response to this, Koyama and Terada modified their design, by including a data dependent operation between layers. The modified family of circuits was presented in the first international security workshop (ISW'97). In this paper, we show that the modified circuit can be easily broken by a differential-like attack. More explicitly, we show that after $d$ rounds, and for any specific key $k$, the input $X$ can be partitioned into $M \leq 2^d$ sets such that the ciphertext $Y$ of each set is related to the plaintext $X$ by an affine relation. In the average case, $M << 2^d$. Our attack enables us to explicitly recover these linear relations. We were able to break an 8−round 64−bit version of this family in few minutes on a workstation using less than $2^{20}$ chosen plaintext-ciphertext pairs.

Keywords: Block cipher, cryptanalysis, augmented parity circuits

## 1 Introduction and definitions

Koyama and Terada [2] proposed a family of cryptographic functions called "non-linear" parity circuits. Youssef and Tavares [4] showed that this family of functions is affine over $GF(2)$ and hence it is completely insecure. In [3], Koyama and Terada introduced a random involution called Value-Dependent-Swapping (VDS). In the VDS, the left half and the right half of a sequence of bits are swapped if its parity is odd. By including this VDS in the parity circuits proposed in [2], Koyama and Terada obtained what they called an augmented version of their cryptographic functions family.
The following definitions are given in [3].

**Definition 1.** Let $x = L||R$ be a sequence of $2k$, $k > 0$ bits where $L$ stands for left half of $x$ and $R$ stands for right, $length(L) = length(R) = k$. A value dependent swapping, or $V(x)$, is defined to be

$$V(x) = \begin{cases} R||L \text{ if } h(x) = 0, \\ L||R \text{ if } h(x) = 1, \end{cases} \tag{1}$$

where $h(x) \in 0, 1$.

**Definition 2.** Let $x = L||R$ be a sequence of $2k$, $k > 0$ bits where $L$ stands for left half of $x$ and $R$ stands for right, $length(L) = length(R) = k$. A VDS, which is an involution value-dependent-swapping based on the parity of the weight of $x$, is defined to be

$$V(x) = \begin{cases} R||L \text{ if } weight(x) \text{ is odd}, \\ L||R \text{ if } weight(x) \text{ is even}, \end{cases} \tag{2}$$

where $weight(x)$ is the number of 1's in the bit sequence $x$.

**Definition 3.** A parity layer with length $n$, or simply an $L(n)$ circuit layer, is a Boolean device with an $n$-bit input and $n$-bit output, characterized by a key that is a sequence of $n$ symbols from $0, 1, +, -$.

**Definition 4.** A function $B = f(K, A)$ computed by an $L(n)$ circuit layer with key $K = k_1k_2 \cdots k_n \in \{0, 1, +, -\}^n$ is the relation from an $n$-bit input sequence $A = a_1a_2 \cdots a_n \in 0, 1^n$ to an $n$-bit sequence $B = b_1b_2 \cdots b_n \in \{0, 1\}^n$ defined below. An $L(n)$ circuit layer computes first the variable $T$ modulo 2 such that

$$T = \bigoplus_{j=1}^{n} t_j \tag{3}$$

where

$$t_j = \begin{cases} 1 \text{ if } (k_j = 0 \text{ and } a_j = 0) \text{ or } (k_j = 1 \text{ and } a_j = 1) \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{Otherwise.} \end{cases} \tag{4}$$

The output $B = b_1b_2 \cdots b_n$ of the circuit layer is then

$$b_j = \begin{cases} \overline{a_j} \text{ if } \begin{cases} k_j = - \text{ and } T = 1 \\ or \\ k_j = + \text{ and } T = 0 \\ or \\ k_j = 1 \end{cases} \\ a_j \qquad\qquad \text{Otherwise.} \end{cases} \tag{5}$$

**Definition 5.** A parity circuit of width $n$ and depth $d$, or simply $C(n, d)$ circuit, is a matrix of $d$ $L(n)$ circuit layers with keys denoted by $K = K_1||K_2 \cdots K_d$ for which the $n$ output bits of the $(i-1)$-th circuit layer are the $n$ input bits for the $i$-th circuit layer, for $2 \leq i \leq d$. The key for the $C(n, d)$ circuit is a $d \times n$ matrix with its $d$ lines containing circuit layer keys.

Let $F$ be the function from $\{0,1\}^n$ to $\{0,1\}^n$ computed by a circuit $C(n,d)$ with key $= K||K_2 \cdots K_d$. That is $F(K,A)$ is defined as

$$F(K,A) = f(K_d, f(K_{d-1}, \cdots, f(K_1, A) \cdots). \tag{6}$$

**Definition 6.** A function $B = f_+(K,A)$ computed by an augmented $L(n)$ circuit layer with key $K$, or simply $L_+(n)$ layer, is the function $V(f(K,A))$, where $V$ is the VDS function as in Definition 2, and $f$ is the function computed by an $L(n)$ circuit layer.

**Definition 7.** A augmented parity circuit of width $n$ and depth $d$, or simply $C_+(n,d)$ circuit, is a matrix of $d$ $L_+(n)$ circuit layers with keys denoted by $K = K_1||K_2 \cdots K_d$ for which the $n$ output bits of the $(i-1)$-th circuit layer are the $n$ input bits for the $i$-th circuit layer, for $2 \leq i \leq d$. The key for the $C_+(n,d)$ circuit is a $d \times n$ matrix with its $d$ lines containing circuit layer keys. A $F_+$ function from $\{0,1\}^n$ to $\{0,1\}^n$ computed by a circuit $C(n,d)$ with key $= K||K_2 \cdots K_d$ as

$$F_+(K,A) = f_+(K_d, f_+(K_{d-1}, \cdots, f_+(K_1, A) \cdots). \tag{7}$$

Table 1 shows the example given in [3] for a $C_+(n,d)$ circuit with $n = 10$ and $d = 3$

| *Input* | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | Swap |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $K_1$ | - | 0 | 1 | - | + | + | 1 | 1 | - | + | |
| *Output* | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | yes |
| $K_2$ | + | 1 | 0 | 1 | 1 | + | 0 | - | + | 1 | |
| *Output* | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | no |
| $K_3$ | - | 0 | 1 | + | + | 0 | - | + | + | - | |
| *Output* | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | yes |

**Table 1.** $C_+(n,d)$ with $n = 10$ and $d = 3$

## 2 Cryptanalysis of the $C_+(n,d)$ circuit

Since the $C(n,d)$ circuit is affine [4], the $C_+(n,d)$ circuit can be viewed as a composition of key-dependent affine transformations and the VDS layer (see Figure 2). Thus the security of the $C_+(n,d)$ relies heavily on the cryptographic strength of the VDS layer.

**Observation 1.** *For any specific key $k$, the ciphertext $Y$ of the $C_+(n,d)$ circuit is related to the plaintext $X$ by one of the affine relations*

$$Y = A_i(k)X \oplus b_i(k), i = 1, 2, \cdots, M \tag{8}$$

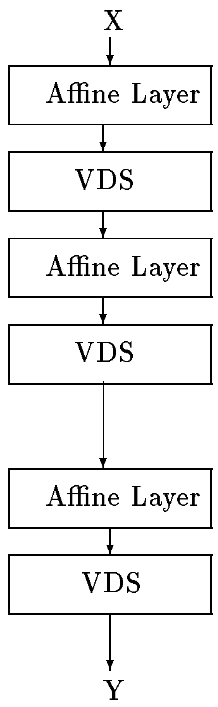


**Fig. 1.** The $C_+(n,d)$ viewed as a composition of affine and VDS layers

*where $A_i(k)$ is a key-dependent non singular binary matrix, $b_i(k)$ is a key-dependent $n \times 1$ binary vector and $M \leq 2^d$*

Proof: Let $VDS_i$ denote the swap variable at round $i$. Thus $VDS_i \in \{0,1\}$ and hence for a $C_+(n,d)$ circuit, $VDS_1, \cdots, VDS_d \in \{0,1\}^d$. Thus the input space of the $C_+(n,d)$ circuit can be partitioned into $2^d$ sets

$$S_1, S_2 \cdots, S_{2^d}, \tag{9}$$

where for any fixed $1 \leq i \leq 2^d$, $VDS_1, \cdots, VDS_d$ is fixed. From Figure 2, the output $Y$ corresponding to the input $X \in S_i$ can be obtained by a composition of fixed affine relations and hence $Y$ is related to $X$ by a fixed affine relation for all $X \in S_i$. Since there is no guarantee that all the $2^d$ possible values of $VDS_1, \cdots, VDS_{2^d}$ will appear, then $M \leq 2^d$. □

Figure 2 illustrates the $C_+(n,d)$ equivalent circuit according to observation 1 above. By noting that $VDS_i$ is a linear function of the input to the layer $i$, then we have

**Observation 2.** *Inputs that belong to the same set in observation 1 above must satisfy a set of $d$ linear equations.*

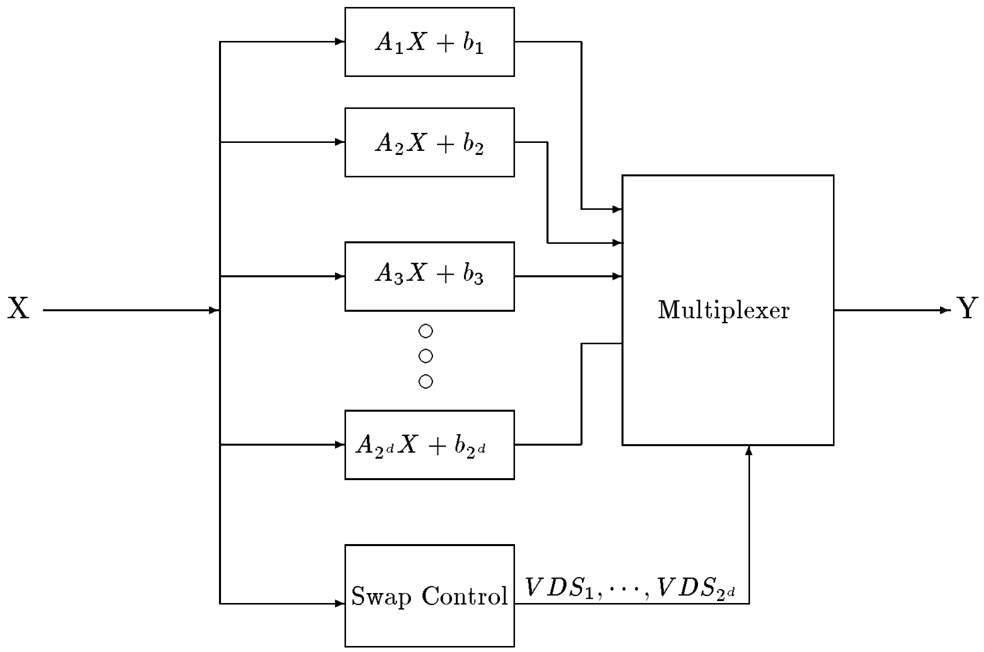


**Fig. 2.** Equivalent circuit of the $C_+(n, d)$ according to observation 1

Figure 4 shows the linear relations corresponding to Example 1 in [3]. While observations 1 and 2 are enough to cause uneasy feeling when using the $C_+(n, d)$ for most practical values of $d$, we extended our attack to find these linear relations. The main idea is to develop an algorithm that can be used to group input/output pairs that belong to the same set $S_i$ and then solve a set of linear equations to find the Matrix $A_i$ and the vector $b_i$. The attack is based on the following observation

**Observation 3.** *For the $C_+(n, d)$, if the input $R_1, R_2$ and $R_3$ belong to the set $S_i$, then*

$$R_4 = R_3 \oplus (R_1 \oplus R_2)$$

*belongs to the same set $S_i$.*

Proof: If $R_1, R_2$ and $R_3 \in S_i$ then they must satisfy a set of $d$ linear equations in the form

$$CR_1 = d, CR_2 = d, CR_3 = d,$$

where $C$ is an $d \times n$ matrix and $d$ is a $d \times 1$ vector. The observation is proved by noting that

$$CR_4 = CR_3 \oplus CR_2 \oplus CR_2 = d$$

and hence $R_4$ also satisfy this set of equation. Thus $R_4$ must belong to the same set $S_i$. $\square$

Note that if $R_1, R_2$, and $R_3 \in S_i$ then for any key $K$

$$F_{+}(K, R_1) \oplus F_{+}(K, R_2) \oplus F_{+}(K, R_3) F_{+}(K, (R_3 \oplus (R_2 \oplus R_3))) = 0 \tag{10}$$

In our attack, we pick random triples $R_1, R_2$ and $R_3$ and test for the condition in equation (10). Since there is no guarantee that $R_3$ will belong to $S_i$ even if $R_1$ and $R_2$ do, we repeat the test for different values of $R_3$ ($Threshold_1$ in Figure 2). We decide that $R_1$ and $R_2$ are in the same set if the condition is satisfied for a large number of times ($Threshold_2$ in Figure 2). Wrong decisions by the algorithm (i.e., if the algorithm declares that $R_2$ and $R_1$ are in the same set while they are not) can be filtered out by collecting more than $n+1$ pairs (e.g., $P = 2n$ pairs) because with high probability the resulting set of equations we will try to solve will be inconsistent if the algorithm accepts wrong pairs. Another method to prevent the algorithm from accepting wrong pairs is to increase the value of $Threshold_1$ and make the value of $Threshold_2$ very close to $Threshold_1$. However, this may increase the number of plaintext-ciphertext pairs required to break the algorithm.

$R_1 = Random()$<br>
*do*<br>
{<br>
$pass = 0$<br>
$R_2 = Random()$<br>
$\delta_x = R_1 \oplus R_2$<br>
for $i = 1$to $i = Threshold_1$<br>
{<br>
$R_3 = Random()$<br>
$R_4 = R_3 \oplus \delta_x$<br>
$delta_y = F_{+}(R_1) \oplus F_{+}(R_2) \oplus F_{+}(R_3) \oplus F_{+}(R_4)$<br>
if $(\delta_y = 0)$ increment pass<br>
}<br>
$mboxif(pass \geq Threshold_2)$ Declare $R_1$ and $R_2 \in$ same set<br>
}while number of collected pairs $\leq P$

**Fig. 3.** Basic Steps in the Attack

## 3 Analysis of the algorithm and experemental results

Assuming that the size of the input sets are equal, then the probability that $R_1, R_2$ and $R_3$ are in the same set is $\frac{1}{M^3}$ where $M$ is the number of partitions.

The maximum value for $M$ is $max(2^d, 2^n)$. Thus the number of chosen plaintext-ciphertext pairs required for the attack increases with $M^3$. In other words, the success of the attack depends heavily on the number of the input partitions. The average number of partitions for $n = 10$ is shown in Table 3. Each point represents an average over 100 $C_+(n, d)$ with randomly selected keys. It is clear that this number is much less than the optimum value $max(2^d, 2^n)$. Our experimental results shows that this large deviation from the optimum case holds for larger block lengths. Note that because we don't know $M$ in advance, it is hard to optimize the choice of $Threshold_1$ and $Threshold_2$ to minimize the number of plaintext-ciphertext pairs required for the attack. Moreover, our experements shows that the $C_+(n, d)$ circuit fails to behave like a random function for practical values of $d$ and hence it is not easy to predicit the probabilty of wrong pairs satisfying equation (10) based on the random function model. The good point (from the attacker point of view) is that the attack works almost all the time. In many cases, we were able to break an 8−round 64−bit version of this family in few minutes on a workstation using less than $2^{20}$ chosen plaintext-ciphertext pairs. Throughout these experements, the value of $Threshold_2$ was set based on the statistics of the *pass* variable (see Figure 2). We set $Threshold_2$ close to the maximum value of *pass*.

| $d$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Average | 2 | 3 | 4 | 7 | 11 | 15 | 25 | 37 | 57 | 62 | 100 | 143 | 162 | 232 | 325 | 393 |
| $max(2^d, 2^n)$ | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 1024 | 1024 | 1024 | 1024 | 1024 | 1024 |

**Table 2.** Average number of sets versus optimal value for $n = 10$

## 4 Conclusion

The security of the $C_+(n, d)$ circuit relies only on the cryptographic strength of the VDS function because the rest of the circuit is affine. Controlling the swapping based on the parity results in a cryptographically weak function. Thus for practical values of $n$ and $d$, the augmented family of parity circuits $C_+(n, d)$ proposed by Koyama and Terada is insecure.

## References


1. Eli Biham, Adi Shamir, *Differential cryptanalysis of DES-like cryptosystems*, Journal of Cryptology, Vol. 4, No. 1, pp. 3-72. 1991.
2. K. Koyama and R. Terada, *Nonlinear Parity Circuits and their cryptographic applications*, Advances in Cryptology, Proceedings of Crypto'90 LNCS537 , pp. 582-599 Springer-Verlag. 1991.

3. K. Koyama and R. Terada, *An Augmented Family of Cryptographic Parity Circuits*, Proceeding of Information Security Workshop (ISW'97). LNCS1396, pp.198-208. Springer-Verlag. 1998.
4. A.M. Youssef and S.E. Tavares, *Cryptanalysis of the "Non-linear parity circuits" proposed at crypto'90*, IEE Electronics Letters. Vol.33, No. 7, pp. 585-586. 1997.

This article was processed using the LaTeX macro package with LLNCS style

# 5 Appendix

$$\text{if} \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1 \\ 1&0&0&0&0&1&0&1&1&1 \\ 1&1&1&1&1&0&0&0&1&1 \end{bmatrix} X = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \text{ then } Y = \begin{bmatrix} 1&1&1&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&1&0&1&1&0&0&0 \\ 0&0&0&1&1&1&0&1&0&0 \\ 0&1&1&1&1&1&1&1&0&0 \\ 0&0&0&0&0&1&1&0&0&0 \\ 0&1&1&0&0&1&0&0&1&0 \\ 0&1&1&0&0&1&0&0&0&1 \end{bmatrix} X \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix},$$

$$\text{if} \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1 \\ 1&0&1&1&1&1&0&0&0&0 \\ 0&0&0&1&1&1&1&1&1&1 \end{bmatrix} X = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \text{ then } Y = \begin{bmatrix} 1&0&0&0&0&1&1&1&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 1&1&0&0&0&0&0&0&0&1 \\ 1&1&0&0&0&0&0&0&1&0 \\ 1&0&1&0&0&0&0&0&1&1 \\ 1&1&1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&0&1&1&0&0 \\ 1&0&0&0&1&0&1&1&0&0 \end{bmatrix} X \oplus \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix},$$

$$\text{if} \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1 \\ 1&0&0&0&0&1&0&1&1&1 \\ 0&1&1&0&0&1&0&1&1&1 \end{bmatrix} X = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ then } Y = \begin{bmatrix} 1&1&1&1&1&1&0&1&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&1&1&1&1&0&1&1&0&0 \\ 1&1&1&1&1&0&0&1&1&0 \\ 1&1&1&1&1&0&0&1&0&1 \\ 1&0&0&1&1&0&0&1&0&0 \\ 1&0&1&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&0&1&1&0&0 \\ 1&0&0&0&1&0&1&1&0&0 \end{bmatrix} X \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix},$$

$$\text{if} \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1 \\ 1&0&1&1&1&1&0&0&0&0 \\ 1&0&1&1&1&0&1&1&0&0 \end{bmatrix} X = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{ then } Y = \begin{bmatrix} 1&0&1&0&0&1&1&1&1&1 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&1&1&0&0&0&1&1&1&1 \\ 0&0&1&1&0&1&1&1&1&1 \\ 0&0&1&0&1&1&1&1&1&1 \\ 0&0&1&0&0&1&0&0&1&1 \\ 0&0&0&0&0&1&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&0 \\ 0&1&1&0&0&1&0&0&1&0 \\ 0&1&1&0&0&1&0&0&0&1 \end{bmatrix} X \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix},$$

**Fig. 4.** Linear Relations for Example 1 in [3]

$$\text{if } \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1 \\ 1&0&0&0&0&1&0&1&1&1 \\ 1&1&1&1&1&0&0&0&1&1 \end{bmatrix} X = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \text{ then } Y = \begin{bmatrix} 0&0&0&1&1&1&0&1&0&0 \\ 0&1&1&1&1&1&1&1&0&0 \\ 0&0&0&0&0&1&1&0&0&0 \\ 0&1&1&0&0&1&0&0&1&0 \\ 0&1&1&0&0&1&0&0&0&1 \\ 1&1&1&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&1&0&1&1&0&0&0 \end{bmatrix} X \oplus \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

$$\text{if } \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1 \\ 1&0&1&1&1&1&0&0&0&0 \\ 0&0&0&1&1&1&1&1&1&1 \end{bmatrix} X = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ then } Y = \begin{bmatrix} 1&0&1&0&0&0&0&0&1&1 \\ 1&1&1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&0&1&1&0&0 \\ 1&0&0&0&1&0&1&1&0&0 \\ 1&0&0&0&0&1&1&1&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 1&1&0&0&0&0&0&0&0&1 \\ 1&1&0&0&0&0&0&0&1&0 \end{bmatrix} X \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

$$\text{if } \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1 \\ 1&0&0&0&0&1&0&1&1&1 \\ 0&1&1&0&0&1&0&1&1&1 \end{bmatrix} X = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ then } Y = \begin{bmatrix} 1&0&1&0&0&0&0&0&1&1 \\ 1&1&1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&0&1&1&0&0 \\ 1&0&0&0&1&0&1&1&0&0 \\ 1&0&0&0&0&1&1&1&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 1&1&0&0&0&0&0&0&0&1 \\ 1&1&0&0&0&0&0&0&1&0 \end{bmatrix} X \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

$$\text{if } \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1 \\ 1&0&1&1&1&1&0&0&0&0 \\ 1&0&1&1&1&0&1&1&0&0 \end{bmatrix} X = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \text{ then } Y = \begin{bmatrix} 1&0&1&0&0&0&0&0&1&1 \\ 1&1&1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&0&1&1&0&0 \\ 1&0&0&0&1&0&1&1&0&0 \\ 1&0&0&0&0&1&1&1&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 1&1&0&0&0&0&0&0&0&1 \\ 1&1&0&0&0&0&0&0&1&0 \end{bmatrix} X \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

**Fig. 4.** (Cont.) Linear Relations for Example 1 in [3]