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 space 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. The expected value of $M \ll 2^d$. 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 [7] 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. In [4],[5] the VDS was incorporated into DES in order to make it stronger against differential and linear cryptanalysis. 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*

D.R. Stinson and S. Tavares (Eds.): SAC 2000, LNCS 2012, pp. 29–38, 2001.

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

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

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

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

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* $K = k_1 k_2 \cdots k_n \in \{0, 1, +, -\}^n$ *is the relation from an* n*-bit input sequence* $A = a_1 a_2 \cdots a_n \in \{0, 1\}^n$ *to an* n*-bit sequence* $B = b_1 b_2 \cdots b_n \in \{0, 1\}^n$ *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 $B = b_1 b_2 \cdots b_n$ *of the circuit layer is then*

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

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.*

Table 1. $C_+(n,d)$ 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	-	+	-	
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

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

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

By showing that, for any fixed key, the $C(n,d)$ circuit can be constructed using XOR gates only, Youssef and Tavares [7] showed that $F(K,A)$ above is affine over $GF(2)$.

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 $2 \le i \le d$. 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*

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

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

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

Since the $C(n,d)$ circuit is affine [7], the $C_+(n,d)$ circuit can be viewed as a composition of key-dependent affine transformations and the VDS layer (see Figure 1). Thus the security of the $C_+(n,d)$ relies heavily on the cryptographic strength of the VDS layer.

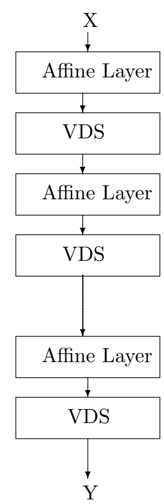

Fig. 1. The $C_+(n, d)$ viewed as a composition of affine and VDS layers

Observation 01 *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*

$$Y = A_i(k)X \oplus b_i(k), \tag{8}$$

where $i = 1, 2, \cdots, M$, $A_i(k)$ 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. I.e., $VDS_i = 0$ if the parity of the input to the VDS layer at round d is even and $VDS_i = 1$ if this parity is odd. 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 and hence the d VDS layers can be modeled by fixed bit permutation layers. 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 01 above. The following observation illustrates how the "swap control" function in this figure operates. By noting that VDS_i is a linear function of the input to layer i, then we have

Observation 02 *Inputs that belong to the same set in observation 01 above must satisfy a set of d linear equations.*

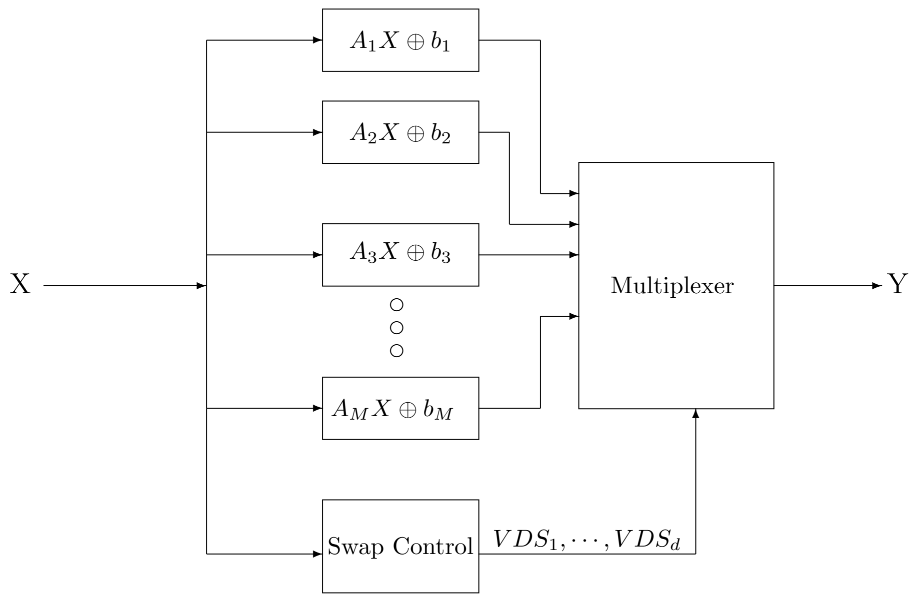

Fig. 2. Equivalent circuit of the $C_+(n, d)$ according to observation 01

For a given known key, these 2^d (d linear relations) can be derived by calculating the parity of the input to the d VDS layers in terms of the input X. If some of these linear relations don't have a solution, then M will be less than 2^d. Figure 4 shows the linear relations corresponding to Example 1 in [3]. Note that for this particular example, we have more than one possible solution for A_is and b_is. Figure 4 shows only one of these possible solutions. While observations 01 and 02 are enough to cause uneasy feeling when using the $C_+(n, d)$ for most practical values of d, we extend our attack to find these linear relations. The main idea is to develop an algorithm that can be used to group the 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 makes use of the following observation

Observation 03 *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

$$CR_1 = b, CR_2 = b, CR_3 = b,$$

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

$$CR_4 = CR_3 \oplus CR_2 \oplus CR_2 = b$$

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

Fig. 3. Basic steps in the attack

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

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

$$F_+(K, R_1) \oplus F_+(K, R_2) \oplus F_+(K, R_3) \oplus F_+(K, (R_3 \oplus (R_1 \oplus R_2))) = 0 \qquad (10)$$

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 ($Trials$ in Figure 3). We decide that R_1 and R_2 are in the same set if the condition is satisfied for a large number of times ($Threshold$ in Figure 3). 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 $Trials$ and make the value of $Threshold$ very close to $Trials$. However, this may increase the number of plaintext-ciphertext pairs required to break the algorithm. Throughout these experiments, the value of $Threshold$ was set based on the statistics of the *pass* variable (see Figure 3). We set $Threshold$ close to the maximum value of *pass*.

3 Analysis of the Algorithm and Experimental 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.

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

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

The maximum value for M is $min(2^d, 2^n)$. 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 intensive use of bit oriented operations in the $C_+(n, d)$ circuits puts an upper-bound on d, and consequently M, for any efficient software implementation. The average number of partitions for $n = 10$ is shown in Table 2. Each point represents an average over 100 $C_+(n, d)$ circuits 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. It is also easy to prove that if the key K is restricted to the set $\{0, 1\}$ instead of $\{0, 1, +, -\}$, then $M \leq 2$ for all $d \geq 1$. Note that because we don't know M in advance, it is hard to optimize the choice of $Trials$ and $Threshold$ to minimize the number of plaintext-ciphertext pairs required for the attack. Moreover, our experiments 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 predict the probability 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.

Remark 1. The non-affineness defined in [3] doesn't provide a useful measure of resistance against linear attacks. The nonlinearity of a function f is defined as the minimum distance between the set of affine functions and all the non-zero linear combinations of the output coordinates of f [6]. Our experiments shows that for practical values of d, the average nonlinearity of the $C_+(n, d)$ circuits is very poor compared to the expected nonlinearity of randomly selected functions of the same size n. Thus it is conceivable that the $C_+(n, d)$ circuit be broken using a variant of linear cryptanalysis [6].

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.

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. (continued)

References

1. Eli Biham and 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. T. Kaneko, K. Koyama and R. Terada, *Dynamic swapping schemes and differential cryptanalysis* IEICE Transactions on Fundamentals, vol. E77-A, pp. 1328-1336, 1994.
5. Y. Nakao, K. Koyama and R. Terada, *The security of an RDES cryptosystem against linear cryptanalysis* IEICE Transactions on Fundamentals, vol. E79-A, pp. 12-19, 1996.
6. M. Matsui, *Linear Cryptanalysis method for DES cipher* Advances in Cryptology, Proceedings of Eurocrypt'93, LNCS 765, pp. 386-397, Springer-Verlag, 1994.
7. 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.