Nonlinear Parity Circuits and their Cryptographic Applications

Kenji Koyama and Routo Terada[1]

NTT Research Laboratories
Musashino-shi, Tokyo 180, Japan

ABSTRACT

This paper proposes a new family of nonlinear cryptographic functions called parity circuits. These parity circuits compute a one-to-one Boolean function, and they can be applied to symmetric block ciphers. In this paper, parity circuits are first defined. Next, these circuits are proven to satisfy some of the properties required in cryptography; involution, nonlinearity, the probability of bit complementation, avalanche effect, equivalent keys and computational efficiency. Finally, the speed of parity circuits implemented using the current hardware technology is estimated to show they can achieve 160 Mbps with a 64-bit block size, 8 rounds, and 3.2 K gates.

1. Introduction

Although the Data Encryption Standard (DES) [NBS77] is widely used and standardized today, there is an increasing interest on alternative cryptographic functions. A few DES-type symmetric block ciphers, for example FEAL [MSS88], or Khufu [Me89], or LOKI [BPS90], have been proposed. In order to achieve systematic design and exact evaluation of symmetric ciphers, cryptographic functions may have to be mathematically simple and cryptographically secure, as well as asymmetric (public key) cryptosystems like RSA.

This paper proposes a new family of nonlinear cryptographic functions called parity circuits. The proposed functions have a simple structure. These parity circuits compute a one-to-one function from $\{0,1\}^n$ to $\{0,1\}^n$, and they can be applied to symmetric block ciphers. This paper first defines parity circuits. Next, their cryptographic properties; involution, nonlinearity, randomness, the probability of bit complementation, avalanche effect, equivalent keys are clarified. Based on an analysis of these, design criteria are shown for parity circuit parameters that can keep cryptosystem secure and rapid. Finally, the speed of parity circuits implemented using the current hardware technology is estimated.

2. Parity Layers and Circuits

First, some basic concepts need to be defined that will be used in later sections.

[1]On a special leave of absence from the University of S. Paulo, Brazil, E-mail address: roterada@brusp.bitnet
This author was partially supported by grant FAPESP 89/2983-0

A.J. Menezes and S.A. Vanstone (Eds.): Advances in Cryptology - CRYPTO '90, LNCS 537, pp. 582-600, 1991.

Some examples of $L(n)$ parity circuit layers for $n = 10$ are as follows:

Table 1. $L(n)$

Example 1. $T = 1$ (odd event)

Input	1	0	1	1	0	0	1	0	0	1
Key	−	0	1	−	+	+	1	1	−	+
Output	0	0	0	0	0	0	0	1	1	1

Example 2. $T = 0$ (even event)

Input	1	0	0	1	0	0	1	0	0	1
Key	−	0	1	−	+	+	1	1	−	+
Output	1	0	1	1	1	1	0	1	0	0

2.2 Parity Circuits C(n, d)

To obtain the so-called cascade effect, we compose the parity circuit layers as follows.

Definition 3 A *parity circuit of width n and depth d*, or simply a $C(n,\ d)$ *circuit* , is a matrix of d $L(n)$ circuit layers with keys denoted by $\mathbf{K} = K_1 \parallel K_2 \parallel \cdots \parallel K_d$ 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 whose d lines contain the circuit layer keys. □

In other words, given n-bit input A and key $\mathbf{K}$, the $C(n,\ d)$ circuit computes function $F(\mathbf{K}, A)$ from $\{0,1\}^n$ to $\{0,1\}^n$ and defines it as:

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

where each $f(K_i, .)$ is computed by the i-th circuit layer.
An example of $C(n,\ d)$ parity circuit is shown below.

Table 2. $C(n,\ d)$ when $n = 10$ and $d = 3$

Input	1	0	1	1	0	0	1	0	0	1
K_1	−	0	1	−	+	+	1	1	−	+
Output	0	0	0	0	0	0	0	1	1	1
K_2	+	1	0	1	1	+	0	−	+	−
Output	1	1	0	1	1	1	0	1	0	1
K_3	−	0	1	+	+	0	−	+	+	-
Output	1	1	1	0	0	1	0	0	1	1

3. Basic Properties of C(n, d) Circuits and Inversion

We must first define the inverse circuit layer in order to decrypt the output of an $L(n)$ circuit layer. The inverse layer operates exactly the same as the $L(n)$ layer, except that exclusive-or using the tester cells is performed before the even or odd parity event is computed.

Definition 4 Function $B = f^{-1}(K,\ A)$, as computed by an $L^{-1}(n)$ *inverse* circuit layer with key

$$K = k_1\ k_2\ \cdots\ k_n$$

is the relation from $\{0,1\}^n$ to $\{0,1\}^n$ defined below. Given input

$$A = a_1\ a_2\ \cdots\ a_n,$$

it first computes intermediate output:

$$A' = a'_1\ a'_2\ \cdots\ a'_n,$$

defined by:

$$a'_j = \begin{cases} \overline{a}_j & \text{if } k_j = 1 \\ a_j & \text{otherwise.} \end{cases}$$

Then, variable T is computed for A' as in Definition 2, and output $B = b_1 b_2 \cdots b_n$ of the circuit layer is then:

$$b_j = \begin{cases} \overline{a}'_j & \text{if} \begin{cases} k_j = - & \text{and } T = 1 \\ k_j = + & \text{and } T = 0 \end{cases} \\ a'_j & \text{otherwise.} \end{cases}$$

□

The correctness of this definition is established by the following lemma.

Lemma 1 *Every $L(n)$ circuit layer that computes f has an inverse layer, $L^{-1}(n)$, to compute f^{-1} (as in Definition 4 above); i.e. , $f^{-1}(K, f(K, A)) = A$, for any n-bit input A and any key K.*

Proof. This lemma is an immediate consequence of Definitions 2 and 4. Notice that after the intermediate output A' is computed as in Definition 4, the entries in A' affecting parity value T (in $L^{-1}(n)$) are the same as for the input to the $L(n)$ layer, and so the $L^{-1}(n)$ layer will compute the same T value, and $L^{-1}(n)$ will again complement the input bits complemented by $L(n)$ (if any). □

Lemma 2 *Let $F(.)$ be the function from $\{0,\ 1\}^n$ to $\{0,\ 1\}^n$ computed by a $C(n,\ d)$ circuit with key $K_1 \parallel K_2 \parallel \cdots \parallel K_d$. Inverse function $F^{-1}(.)$ is computed by the "inverted" circuit, $C^{-1}(n,\ d)$, with key:*

$$K_d \parallel K_{d-1} \parallel \cdots \parallel K_1.$$

Proof. This lemma immediately follows from consecutive application of Lemma 1 until concatenation of the two circuits characterized by:

$$\underbrace{K_1 \parallel K_2 \parallel \cdots \parallel K_d} \parallel \underbrace{K_d \parallel K_{d-1} \parallel \cdots \parallel K_1}.$$

□

It can be concluded that:

Theorem 1 *Every $C(n,\ d)$ circuit computes a one-to-one function from $\{0,1\}^n$ to $\{0,1\}^n$.*

Proof. By Lemma 2, the $C(n,\ d)$ circuit computes function $F(.)$, which always admits an inverse $F^{-1}(.)$, so this lemma follows. □

Next, we show a basic property of $L(n)$ ciurcuit layer. If the $L(n)$ circuit layers re randomly generated, with uniform distribution of symbols $\{0, 1, -, +\}$, an average of $n/4$ symbols for each type will occur in the key, and thus, about $n/2$ cells will be testers. According to this hypothesis, it can be seen that around half of all the possible input values imply an even event (*i.e.* , variable T in Definition 2 will be 0), and the other half imply an odd one. More precisely, we can prove the following:

Theorem 2 *If the $L(n)$ circuit layers are uniformly generated, then*

$$Prob\{\text{even event}\} = \frac{1}{2} + \frac{1}{2^{n+1}}, \qquad Prob\{\text{odd event}\} = \frac{1}{2} - \frac{1}{2^{n+1}}.$$

Proof. First, assume that each $L(n)$ circuit layer contains at least one tester cell. Let k_j be one of the tester cells. By uniform distribution of the keys, we have

$$Prob\{k_j = 0\} = 1/2, \qquad Prob\{k_j = 1\} = 1/2,$$

so that

$$Prob\{t_j = 0\} = 1/2, \qquad Prob\{t_j = 1\} = 1/2.$$

This conclusion independently holds for any tester cell in the key. Thus, by summing t_j modulo 2 over all the tester key positions, we have

$$Prob\{T = 0 \text{ (even event)}\}=1/2, \qquad Prob\{T =1 \text{ (odd event)}\}=1/2.$$

That is, the probability of an even as well as odd event is 1/2 for the layer, if it contains at least one tester cell.

There are 4^n keys, but 2^n of these keys have no tester. So, there are $(4^n - 2^n)$ keys implying an even event with a probability of 1/2. Additionally, there are 2^n keys without a tester which always implies an even event. Therefore, we have the following probability of even event:

$$Prob\{\text{even event}\} = \frac{(4^n - 2^n)}{2 \times 4^n} + \frac{2^n}{4^n} = \frac{1}{2} + \frac{1}{2^{n+1}},$$

and the case for $T = 0$ is proved.

The case for $T = 1$ is complementary. There are $(4^n - 2^n)$ keys that imply an odd event with a probability of 1/2. Therefore, we have the following probability of odd event:

$$Prob\{\text{odd event}\} = \frac{(4^n - 2^n)}{2 \times 4^n} = \frac{1}{2} - \frac{1}{2^{n+1}}. \quad \square$$

4. Cryptographic Properties

We will now consider certain properties of the $C(n,\ d)$ circuits that are relevant to their use as cryptographic devices. It will be shown how n and d affect nonlinearity, the probability of bit complementation, avalanche effect, output randomness, and the existence of equivalent keys. Furthermore, the n and d values can be increased as necessary to properly secure a cryptosystem.

4.1 Nonlinearity

Strictly speaking, *exclusive or* operations are *nonlinear* in the sense that

$$f(K,\ I_1 + I_2) \neq f(K,\ I_1) + f(K,\ I_2).$$

Thus, almost all of the $L(n)$ circuit layers are *nonlinear* except for particular cases where a key contains only $-$ or only 0 symbols. These particular keys equalize the input and output, and the occurrence probability is given by $1-(1-(1/4)^n)^2 = 2^{1-2n} - 2^{-4n}$. Note that the parity circuit has *non-affine* transformation. Whenever a key contains at least one tester cell and one inverter cell, the function computed by its $L(n)$ circuit layer is a *non-homomorphism* in the sense that

$$f(K,\ I_1 \circ I_2) \neq f(K,\ I_1) \circ f(K,\ I_2).$$

If a key contains only $\{-,+\}$ symbols or only $\{0,1\}$ symbols, then we have

$$f(K,\ I_1 \oplus I_2) = f(K,\ I_1) \oplus f(K,\ I_2).$$

This case is the so-called Vernam cipher, and the occurrence probability is given by $1-(1-(1/2)^n)^2 = 2^{1-n} - 2^{-2n}$. Properties such as strict nonlinearity, non-affineness and non-homomorphism are called nonlinearity in this paper. When randomly chosen $L(n)$ circuit layers are combined in circuits $C(n,\ d)$, nonlinear behavior is preserved with a high probability. This nonlinear characteristic is a desirable attribute in any cryptographic function [Ru86, MS89], since it increases the difficulty of breaking the cipher.

The *order* of Boolean canonical form of a nonlinear function, defined to be the maximum of the order of its product terms, is often applied as a measure of nonlinearity [Ru86]. For instance, the Boolean expressions for $L(n)$ when $n = 2$ and $k_1 \in \{0,1\}$ are

$$b_2 = \overline{a}_1\overline{a}_2\ k_1 + \overline{a}_1 a_2 \overline{k}_1 + a_1\overline{a}_2\overline{k}_1 + a_1 a_2 k_1 \quad \text{if } k_2 = +,$$
$$b_2 = \overline{a}_1\overline{a}_2\ \overline{k}_1 + \overline{a}_1\ a_2\ k_1 + a_1\overline{a}_2\ k_1 + a_1 a_2 \overline{k}_1 \quad \text{if } k_2 = -.$$

The order of the $C(n,\ d)$ circuit increases exponentially as n or d increases. It would be practically infeasible to cryptoanalyze $C(n,\ d)$ using its Boolean expression if $n \geq 64$ and $d \geq 8$.

4.2 Probability of Complementation

Now we are going to prove a complementation property regarding the influence of the parameters d and n on the behavior of a $C(n,\ d)$ circuit.

Lemma 3 *If a $C(n,\ d)$ circuit is uniformly generated, then we have the following formulas for the times that one input element a_j $(1 \leq j \leq n)$ will be complemented by the d circuit layers. Probability that one or more complementations occur is:*

$$1 - \Big(\frac{147}{256} - \frac{3}{2^{2n+8}}\Big)^d \approx 1 - (0.57)^d.$$

The average of complementation times is:

$$d\Big(1 - \frac{147}{256} + \frac{3}{2^{2n+8}}\Big) \approx 0.43d.$$

The variance of complementation times is:

$$d\Big(1-\frac{147}{256}+\frac{3}{2^{2n+8}}\Big)\Big(\frac{147}{256}-\frac{3}{2^{2n+8}}\Big)\approx 0.25d.$$

Proof: First consider any uniformly generated $L(n)$ layer of a $C(n,\ d)$ circuit. For fixed position j, $1\le j\le n$, we have

$$Prob\{k_j=0\}=1/4,\quad Prob\{k_j=1\}=1/4,$$
$$Prob\{k_j=-\}=1/4,\quad Prob\{k_j=+\}=1/4.$$

Let X be event "$k_j=+$ and $T=0$" in the $L(n)$ layer, and a_j will be complemented by the *even inverter*. For this compound event, by Theorem 2, we have

$$Prob\{X\}=\frac{1}{4}\times(\frac{1}{2}+\frac{1}{2^{n+1}})=\frac{1}{8}+\frac{1}{2^{n+3}},$$

and

$$Prob\{not\ X\}=1-(\frac{1}{8}+\frac{1}{2^{n+3}})=\frac{7}{8}-\frac{1}{2^{n+3}}.$$

Similarly, let Y be event "$k_j=-$ and $T=1$"; *i.e.*, a_j will be complemented by the *odd inverter*. Again, by Theorem 2, we have

$$Prob\{Y\}=\frac{1}{8}-\frac{1}{2^{n+3}},\quad Prob\{not\ Y\}=\frac{7}{8}+\frac{1}{2^{n+3}}.$$

Let Z be event "$k_j=1$"; *i.e.*, a_j will be complemented by the tester.

$$Prob\{Z\}=1/4,\quad Prob\{not\ Z\}=3/4.$$

By combining these three bounds, we have

$$\begin{aligned}Prob\{not\ (X\ or\ Y\ or\ Z)\}&=(\frac{7}{8}-\frac{1}{2^{n+3}})\times(\frac{7}{8}+\frac{1}{2^{n+3}})\times\frac{3}{4}\\&=\frac{147}{256}-\frac{3}{2^{2n+8}}\end{aligned}$$

Now, considering that d circuit layers are uniformly and independently generated, we have

$$Prob\{a_j \text{ be complemented one or more times in } d \text{ circuit layers}\}$$
$$=1-\Big(\frac{147}{256}-\frac{3}{2^{2n+8}}\Big)^d\approx 1-(0.57)^d.$$

This probability quickly converges to one as d increases. Let p be the probability that the input element is complemented in one layer:

$$p=1-\frac{147}{256}+\frac{3}{2^{2n+8}}$$

Considering now d layers, we have:

$$Prob\{i \text{ complementations in } d \text{ layers}\} = \binom{d}{i} p^i (1-p)^{(d-i)}.$$

Using well known results of binomial distribution, the average of complementation times in d layers is dp, and their variance is $dp(1-p)$.

□

The probability of any input bit being complemented an odd or an even number of times by a circuit is established as follows.

Theorem 3 *If a $C(n,\ d)$ circuit is uniformly generated, the element a_j, for any $1 \leq j \leq n$ of the input sequence, is complemented an odd number of times with a probability asymptotic to 0.5; and an even number of times with the same probability.*

Proof. As in the Lemma 3 proof, we have:

$$Prob\{i \text{ complementations in } d \text{ layers}\} = \binom{d}{i} p^i (1-p)^{(d-i)}.$$

Let P_C be the probability that input element a_j is complemented in d layers of $C(n,\ d)$:

$$P_C = Prob\{b_j = \overline{a}_j\}.$$

This P_C is computable by summing the above expression over all the odd $i's$, $0 \leq i \leq d$:

$$P_C = Prob\{\text{odd number of complementations in } d \text{ layers}\}$$

$$= \sum_{i \text{ is odd},\ i=0}^{d} \binom{d}{i} p^i (1-p)^{(d-i)}$$

Note that the probability of non-complementation is given by $1 - P_C$, which is the one summed over all the even i's for $0 \leq i \leq d$. Both of these probabilities very quickly converge to $1/2$ as d increases. □

For example, if $n = 10$, we have

$$P_C = 0.425781253 \quad \text{if } d = 1$$

$$P_C = 0.499757258 \quad \text{if } d = 4$$

$$P_C = 0.499999862 \quad \text{if } d = 8$$

$$P_C = 0.500000000 \quad \text{if } d = 16$$

4.3 Avalanche Effect and Output Randomness

It is desirable that a cryptographic function exhibits the so-called *avalanche effect*; i.e., a small change in the plaintext or the key gives rise to a large change in the ciphertext [Fe73, Ko81]. This avalanche effect will be analized for our proposed function F.

4.3.1 Avalanche Effect between Plaintext and Ciphertext

First, the avalanche effect between the input (plaintext) and output (ciphertext) is analyzed. Given a $C(n,\ d)$ circuit and input pair $(A_1,\ A_2)$ with Hamming distance of i $(1 \le i \le n)$, the *average Hamming distance of output pairs* $(F(\mathbf{K},\ A_1),\ F(\mathbf{K},\ A_2))$ is denoted by $H_I(n,\ d,\ i)$. This average $H_I(n,\ d,\ i)$ is defined over all inputs and keys as; $\binom{n}{i}$ input pairs and 4nd keys. When $d = 1$, average output distance $H_I(n,\ 1,\ i)$ for any n can be directly derived as follows.

Lemma 4 *If a $L(n)$ circuit layer is uniformly distributed, then $H_I(n,\ 1,\ i)$ is explicitly expressed by*

$$H_I(n,\ 1,\ 1) = \frac{n}{4} + \frac{3}{4},$$

$$H_I(n,\ 1,\ i) = \frac{n}{4} + \frac{i}{2} \quad \text{if } i \ge 2.$$

Proof. Let $\mathcal{K}$ be a set of keys corresponding to changes in an input pair. Let $\overline{\mathcal{K}}$ be $K - \mathcal{K}$. Note $\mathcal{K}$ and $\overline{\mathcal{K}}$ have cardinalities i and $n - i$, respectively. Consequently, we have two possible cases.

(1) $\mathcal{K}$ has an *odd* number of tester keys.

In this case, the i-bit change in the input implies the conversion of a parity event between odd and even. Thus, only the following output bits are complemented by the change in the input.

(i) Output bits at the inverter keys in $\overline{\mathcal{K}}$ whose average number is $(n - i)/2$.

(ii) Output bits at the tester keys in $\mathcal{K}$ whose average number is: 1 if $i = 1$, $i/2$ if $i \ge 2$.

Thus, the average Hamming distance of the output pair in this case is:

$$\frac{n-1}{2} + 1 = \frac{n}{2} + \frac{1}{2} \text{ if } i = 1, \qquad \frac{n-i}{2} + \frac{i}{2} = \frac{n}{2} \text{ if } i \ge 2.$$

(2) $\mathcal{K}$ has an *even* number of tester keys.

In this case, the i-bit change in the input *never* converts the parity event between odd and even. Thus, only the following output bits are complemented by the change in the input.

(i) Output bits at the inverter keys in $\mathcal{K}$ whose average number is: 1 if $i = 1$, $i/2$ if $i \ge 2$.

(ii) Output bits at the tester keys in $\mathcal{K}$ whose average number is: 0 if $i = 1$, $i/2$ if $i \ge 2$.

Thus, the average Hamming distance of the output pair in this case is:

$$1 + 0 = 1 \text{ if } i = 1, \qquad \frac{i}{2} + \frac{i}{2} = i \text{ if } i \ge 2.$$

For the above cases, we have

$$Prob\{\mathcal{K} \text{ has } odd \text{ number of tester keys}\} = 1/2,$$

$$Prob\{\mathcal{K} \text{ has } even \text{ number of tester keys}\} = 1/2.$$

In summation, the average Hamming distance of the output pair is:

$$H_I(n,\ 1,\ i) = \frac{1}{2}(\frac{n}{2} + \frac{1}{2}) + \frac{1}{2} \times 1 = \frac{n}{4} + \frac{3}{4} \quad \text{if } i = 1,$$

$$H_I(n,\ 1,\ i) = \frac{1}{2} \times \frac{n}{2} + \frac{1}{2} \times i = \frac{n}{4} + \frac{i}{2} \quad \text{if } i \geq 2.$$

Thus, the lemma has been proved. □

Examples of $H_I(n,\ 1,\ i)$ are shown in Table 3.

Table 3. Average output distances $H_I(n,\ 1,\ i)$

	n=1	n=2	n=3	n=4	n=5	n=6
$i = 1$	1.00	1.25	1.50	1.75	2.00	2.25
$i = 2$	—	1.50	1.75	2.00	2.25	2.50
$i = 3$	—	—	2.25	2.50	2.75	3.00
$i = 4$	—	—	—	3.00	3.25	3.50
$i = 5$	—	—	—	—	3.75	4.00
$i = 6$	—	—	—	—	—	4.50

If only the average distances of the intermediate values are used without analysis of the exact distribution of the distances, then $H_I(n,\ d,\ i)$ when $d \geq 2$ can be approximately estimated as follows:

Lemma 5 *If a $C(n,\ d)$ circuit is uniformly distributed, then $H_I(n,\ d,\ i)$ is approximately expressed by*

$$\frac{n}{2} + (\frac{1}{2})^d(i - \frac{n}{2}) \leq H_I(n,\ d,\ i) \leq \frac{n+1}{2} + (\frac{1}{2})^d(i - \frac{n+1}{2}).$$

Proof. Since we only consider the average distances of the intermediate values, then $H_I(n,\ d,\ i)$ when $d \geq 2$ can be approximately estimated by using $H_I(n,\ 1,\ i)$ as

$$H_I(n,\ d,\ i) = \overbrace{H_I(n,\ 1,\ H_I(n,\ 1,\ H_I(\ldots,\ H_I(n,\ 1,\ i))))}^{d \text{ times}}.$$

From Lemma 4, $H_I(n,\ 1,\ i)$ is generally expressed by

$$\frac{n}{4} + \frac{i}{2} \leq H_I(n,\ 1,\ i) \leq \frac{n}{4} + \frac{1}{4} + \frac{i}{2} \quad \text{for all } i.$$

Thus, we have

$$H_I(n,\ d,\ i) \geq \frac{n}{4}(1 + \frac{1}{2} + (\frac{1}{2})^2 + \cdots + (\frac{1}{2})^{d-1}) + i(\frac{1}{2})^d,$$

$$H_I(n,\ d,\ i) \leq \frac{n+1}{4}(1 + \frac{1}{2} + (\frac{1}{2})^2 + \cdots + (\frac{1}{2})^{d-1}) + i(\frac{1}{2})^d.$$

By rewriting a finite geometric sum for $H_I(n,\ d,\ i)$, we have

$$\frac{n}{2} + (\frac{1}{2})^d(i - \frac{n}{2}) \leq H_I(n,\ d,\ i) \leq \frac{n+1}{2} + (\frac{1}{2})^d(i - \frac{n+1}{2}). \quad \square$$

Note that $H_I(n,\ d,\ i)$ converges to a value between $n/2$ and $(n+1)/2$ as $d \to \infty$ regardless of the i value. Using the formula in Lemma 5, we can observe the dependency of n, d and i for the avalanche effect.

To obtain the exact $H_I(n,\ d,\ i)$ when $d \geq 2$, an analysis is needed that includes intermediate distance distribution. Thus, we introduce a transition probability based on the Markov chain theory. Given input pair $(A_1,\ A_2)$ with Hamming distance of i $(1 \leq i \leq n)$, then *the probability that the Hamming distance of output pairs* $(f(K,\ A_1),\ f(K,\ A_2))$ *is* j $(0 \leq j \leq n)$ is denoted by $P_n(i,\ j)$. This transition probability, $P_n(i,\ j)$, is defined over all the inputs and keys as; $\binom{n}{i}$ input pairs and 4^n keys, and it satisfies

$$0 \leq P_n(i,\ j) \leq 1, \quad \sum_{j=0}^{n} P_n(i,\ j) = 1.$$

Since our proposed function f is a one-to-one mapping, note that

$$P_n(i,\ 0) = 0 \quad \text{if } i > 0.$$

We have obtained some of the $P_n(i,\ j)$ values through computer simulation. Some examples of $[P_n(i,\ j)]$ matrices such that $2 \leq n \leq 6$ are as follows.

$$[P_2(i,\ j)] = \begin{pmatrix} 3/4 & 1/4 \\ 1/2 & 1/2 \end{pmatrix}, \quad [P_3(i,\ j)] = \begin{pmatrix} 5/8 & 2/8 & 1/8 \\ 1/4 & 3/4 & 0 \\ 3/8 & 0 & 5/8 \end{pmatrix},$$

$$[P_4(i,\ j)] = \begin{pmatrix} 9/16 & 3/16 & 3/16 & 1/16 \\ 1/8 & 6/8 & 1/8 & 0 \\ 3/16 & 3/16 & 9/16 & 1/16 \\ 1/4 & 0 & 1/4 & 2/4 \end{pmatrix}, \quad [P_5(i,\ j)] = \begin{pmatrix} 17/32 & 4/32 & 6/32 & 4/32 & 1/32 \\ 1/16 & 11/16 & 3/16 & 1/16 & 0 \\ 3/32 & 6/32 & 20/32 & 2/32 & 1/32 \\ 1/8 & 1/8 & 1/8 & 5/8 & 0 \\ 5/32 & 0 & 10/32 & 0 & 17/32 \end{pmatrix},$$

$$[P_6(i,\ j)] = \begin{pmatrix} 33/64 & 5/64 & 10/64 & 10/64 & 5/64 & 1/64 \\ 1/32 & 20/32 & 6/32 & 4/32 & 1/32 & 0 \\ 3/64 & 9/64 & 42/64 & 6/64 & 3/64 & 1/64 \\ 1/16 & 2/16 & 2/16 & 10/16 & 1/16 & 0 \\ 5/64 & 5/64 & 10/64 & 10/64 & 33/64 & 1/64 \\ 3/32 & 0 & 10/32 & 0 & 3/32 & 16/32 \end{pmatrix}.$$

Using the $[P_n(i,\ j)]$ matrices, an exact $H_I(n,\ d,\ i)$ is generally expressed by

$$H_I(n,\ d,\ i) = \sum_{j=1}^{n} j G_{n,d}(i,\ j), \quad \text{where} \quad [G_{n,d}(i,\ j)] = [P_n(i,\ j)]^d.$$

If $d = 1$ and $i \geq 1$, then $H_I(n,\ 1,\ i)$ can be simply expressed as

$$H_I(n,\ 1,\ i) = \sum_{j=1}^{n} j G_{n,1}(i,\ j) = \sum_{j=1}^{n} j P_n(i,\ j),$$

and its explicit formula using n and i can be obtained in Lemma 4.

Since a Markov chain with a transition probability of $[P_n(i,\ j)]$ is ergodic (i.e. irreducible and non-periodic), it has a stable limit distribution, $[G_{n,\infty}(i,\ j)]$, defined by

$$[G_{n,\infty}(i,\ j)] = [P_n(i,\ j)]^{\infty}.$$

Since $[G_{n,\infty}(i,\ j)]$ is rewritten as $[P_n(i,\ j)]^m$ to satisfy

$$[P_n(i,\ j)]^{m+1} = [P_n(i,\ j)]^m,$$

the elements of $G_{n,\infty}(i,\ j)$ are directly derived from $P_n(i,\ j)$ by solving a system of linear equations

$$[G_{n,\infty}(i,\ j)] \times [\mathbf{I} - [P_n(i,\ j)]] = \mathbf{0},$$

$$\sum_{j=1}^{n} G_{n,\infty}(i,\ j) = 1,$$

where $\mathbf{I}$ and $\mathbf{0}$ denote a unit matrix and a zero matrix, respectively.

Table 4 shows average output distances for $H_I(n,\ d,\ i)$ when $i = 1$. The values for $H_I(n,\ d,\ 1)$ when $n \leq 6$ and $d = 2,\ 3,\ 16$, and ∞ are calculated by the transition probability matrices. The analytical results coincide with the ones obtained by exhaustive computer simulation. The values for $H_I(n,\ d,\ 1)$ when $n \geq 8$ in Table 4 are the results of 1 million random samplings.

Table 4. Average output distance $H_I(n,\ d,\ 1)$

	n=1	n=2	n=3	n=4	n=6	n=8	n=16	n=32	n=64
$d = 1$	1	1.2500	1.5000	1.7500	2.2500	2.7500	4.7500	8.7500	16.750
$d = 2$	1	1.3125	1.6563	2.0156	2.7539	3.5469	6.3750	11.750	22.508
$d = 3$	1	1.3281	1.7031	2.1055	2.9502	3.9625	7.4625	14.463	28.463
$d = 16$	1	1.3331	1.7141	2.1331	3.0474	4.0021	8.0000	16.000	32.000
$d = \infty$	1	1.3333	1.7143	2.1333	3.0476	4.0023	8.0000	16.000	32.000

In Table 4, we can observe that the $H_I(n,\ d,\ i)$ estimate in Lemma 5 seems to be a "good" approximation, and that the average output distance $H_I(n,\ d,\ i)$ converges to $n/2$ as both n and d increase.

4.3.2 Avalanche Effect between Key and Ciphertext

Given two sequences, $\mathbf{K} = (k_1,\ k_2,\ \ldots,\ k_m)$ and $\mathbf{K}' = (k'_1,\ k'_2,\ \ldots,\ k'_m)$, of m symbols from $\{0,\ 1,\ -,\ +\}$, *key symbol distance* s is defined as

$$s = \sum_{\ell=1}^{m} c_\ell \quad \text{where} \quad c_\ell = \begin{cases} 0 & \text{if } k_\ell = k'_\ell \\ 1 & \text{otherwise.} \end{cases}$$

Given a $C(n,\ d)$ circuit and a key pair $(\mathbf{K},\ \mathbf{K}')$ whose key symbol distance is i, the *average Hamming distance of output pairs* $(F(\mathbf{K},\ A),\ F(\mathbf{K}',\ A))$ is denoted by $H_K(n,\ d,\ i)$.

When $d = 1$, average output distance $H_K(n,\ d,\ i)$ for any n can be directly derived as follows.

Lemma 6 *If an input (plaintext) to an $L(n)$ circuit layer is uniformly distributed and fixed, and the keys of the key pair $(K,\ K')$ are also uniformly distributed, then $H_K(n,\ 1,\ i)$ is explicitly expressed by*

$$H_K(n,\ 1,\ 1) = \frac{n}{4} + \frac{5}{12},$$

$$H_K(n,\ 1,\ i) = \frac{n}{4} + \frac{i}{3} \quad \text{if } i \geq 2.$$

Proof. (Proof is similar to the analysis of Lemma 4 proof, and will be given in the full paper.)

If we only consider key changes in the first layer of $C(n,d)$ circuit, then $H_K(n,\ d,\ i)$ when $d \geq 2$ can be approximately estimated as follows.

Lemma 7 *If an input (plaintext) to a $C(n,\ d)$ circuit is uniformly distributed and fixed, and the keys of the key pair $(\mathbf{K},\ \mathbf{K}')$ are also uniformly distributed where $K_1 \neq K_1'$, $K_\ell = K_\ell'$ $(2 \leq \ell \leq d)$, then $H_K(n,\ d,\ i)$ is approximately expressed by*

$$\frac{n}{2} + \left(\frac{1}{2}\right)^{d-1}\left(\frac{i}{3} - \frac{n}{4}\right) \leq H_K(n,\ d,\ i) \leq \frac{n}{2} + \left(\frac{1}{2}\right)^{d-1}\left(\frac{1}{12} + \frac{i}{3} - \frac{n}{4}\right).$$

Proof. By combining the results of Lemmas 4 and 6, $H_K(n,\ d,\ i)$ can be estimated similarly to Lemma 5. □

In this subsection, we discussed the avalanche effect between an *internal key* and the output of one $L(n)$ layer. From a practical viewpoint, it is necessary to clarify the avalanche effect between an *external key* and an output. The avalanche effect depends on a *key generation scheme* or *key schedule calculation scheme*, which will be described in Section 5.1. The results obtained in Section 4.3 are useful to design an optimal scheme.

4.3.3 Completeness and Avalanche Effect

The notion of avalanche effect has a close relationship with *completeness* defined as:

Definition 5 (Completeness)
A function is complete if and only if each output bit depends on all of the input bits. □

Kam and Davida [KD79] showed a method of designing substitution-permutation encryption scheme to meet the *completeness* condition. As for function F based on a $C(n,\ d)$ circuit, the completeness condition is expressed as follows:

Lemma 8 *Let $\mathbf{K} = K_1K_2\cdots K_d$ and $K_\ell = k_{\ell 1}k_{\ell 2}\cdots k_{\ell n}$ $(1 \leq \ell \leq d)$. Function $F(\mathbf{K},\ .)$ based on a $C(n,\ d)$ circuit is complete if and only if*

$$\Big(\ k_{\ell i} = \{\text{inverter}\},\ k_{\ell j} = \{\text{tester}\}\ \ 1 \leq i,\ j \leq n,\ \ i \neq j,\ \ \exists \ell\ \Big)\ \ \forall i,\ \forall j.$$

Proof. (Sketch) If $k_{\ell i} = \{\text{inverter}\}$, and $k_{\ell j} = \{\text{tester}\}$ in the same layer, then i-th output bit depends on j-th input bit. If this relation satisfies in any one layer of d layers for all i and for all j $(1 \leq i,\ j \leq n)$, then function F is complete, vice versa. □

From Lemma 8 , we get a necessary and sufficient condition of completeness as follows:

Lemma 9 *There exists a complete function F based on a $C(n,\ d)$ circuit if and only if $d \geq n$.*

Proof. If $d < n$, it is impossible to construct a parity circuit satisfying complete condition described in Lemma 8. If $d \geq n$, we have an instance of complete function such that

$$k_{\ell\ell} = \{\text{inverter}\},\quad k_{\ell j} = \{\text{tester}\},\quad \ell \neq j,\quad 1 \leq \ell,\ j \leq n. \qquad \square$$

Webster and Tavares [WT86] introduced the *strict avalanche criterion* in order to combine the notions of the completeness and the so-called *avalanche effect* [Fe73]. Forre [Fo88] and Lloyd [Ll89] discussed this strict avalanche criterion for some cryptographic functions. Definitions of these criteria are summarized as follows.

Definition 6 (Avalanche Effect)
A function exhibits the *avalanche effect* if and only if an average of half of output bits change whenever a single input bit is complemented. □

Definition 7 (Strict Avalanche Criterion)
A function satisfies the *strict avalanche criterion* if and only if each output bit changes with probability 1/2 whenever a single input bit is complemented. □

In Sections 4.3.1 and 4.3.2., we showed that *avalance effect* is satisfied for function F based on $C(n,\ d)$ circuit for large n and d. Furthermore, the relationship between completeness and the strict avalanche effect is obtained for $C(n,\ d)$ circuit as follows.

Lemma 10 *If function F based on $C(n,\ d)$ circuit is complete, then it satisfies the strict avalanche criterion.*

Proof. It is clear from the definitions and Lemma 8. □

4.4 Equivalent Keys

Let π_n be the group of all permutations on $\{0,\ 1\}^n$, the set of n-bit messages. For a given key $\mathbf{K}$, function $F(\mathbf{K},\ .)$ defines an element of π_n. Note that the group π_n has cardinality $(2^n)!$. The key space generates a subset of π_n. Since a $\mathbf{K}$ key defining $F(\mathbf{K},\ .)$ is a sequence of nd symbols from $\{\ 0,\ 1,\ -,\ +\}$, the number of distinct $\mathbf{K}$ keys is 4nd. That is, the cardinality of the key space is 4nd. If $d > n/2$, then $4^{nd} > 2^n!$, so equivalent keys must exist.

4.4.1 Equivalent Keys for One Input-Output Pair

When an input-output pair $(I,\ O)$ is given, we need to know how many distinct equivalent keys exist. Distinct equivalent keys for pair $(I_i,\ O_j)$ are defined as $\mathbf{K_1}$ and $\mathbf{K_2}$ such that:

$$O_j = F(\mathbf{K_1},\ I_i) = F(\mathbf{K_2},\ I_i), \quad \mathbf{K_1} \neq \mathbf{K_2}.$$

Table 5 shows all the distinct equivalent keys when $n = 2,\ d = 1$.

Table 5. Distinct equivalent keys when $n = 2,\ d = 1$

$O \backslash I$	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	(0,0), (−,−) (0,+), (+,0)	(0,1), (−,+) (0,−), (+,1)	(1,0), (+,−) (−,0), (1,+)	(1,1), (+,+) (−,1), (1,−)
(0,1)	(0,1), (−,+) (−,1), (0,−)	(0,0), (−,−) (−,0), (0,+)	(1,1), (+,+) (+,1), (1,−)	(1,0), (+,−) (+,0), (1,+)
(1,0)	(1,0), (+,−) (−,0), (1,−)	(1,1), (+,+) (−,1), (1,+)	(0,0), (−,−) (0,−), (+,0)	(0,1), (−,+) (0,+), (+,1)
(1,1)	(1,1), (+,+) (+,1), (1,+)	(1,0), (+,−) (+,0), (1,−)	(0,1), (−,+) (−,1), (0,+)	(0,0), (−,−) (−,0), (0,−)

Let $N_{ij}(n,\ d)$ be the number of distinct $\mathbf{K}$ keys such that $O_j = F(\mathbf{K},\ I_i)$ for fixed input I_i $(1 \leq i \leq 2^n)$ and fixed output O_j $(1 \leq j \leq 2^n)$. From Table 5, we can observe that $N_{ij}(2,\ 1) = 4$ for all i and j. As for $N_{ij}(n,\ d)$, we get the following theorem.

Theorem 4 *For all $(n,\ d)$ pairs, number $N_{ij}(n,\ d)$ is the same for all i and j. Number $N_{ij}(n,\ d)$ can be abbreviated as $N(n,\ d)$, and expressed by*

$$N(n,\ d) = 2^{(2d-1)n}.$$

Proof. Note that the cardinality of the key space is 4nd. Assume that an input is fixed as I_i. When $F(\mathbf{K},\ I_i)$ is computed with 4nd distinct keys, output values $\{0,\ 1\}^n$ are classified into 2^n distinct classes. Since the $\mathbf{K}$ keys are uniformly distributed, the output values are also uniformly distributed by the complementation property shown in Theorem 3. Thus, $N_{ij}(n,\ d)$ is the same for a fixed i and for all j $(1 \leq j \leq n)$. Furthermore, it is expressed by

$$N_{ij}(n,\ d) = \frac{4^{nd}}{2^n} = 2^{(2d-1)n} \quad \text{for all } j.$$

It is clear that this holds for all i. Thus, the theorem has been proved. □

Through computer simulation, we have confirmed that Theorem 4 holds. Table 6 shows $N(n,\ d)$ values such that $1 \leq n \leq 4,\ 1 \leq d \leq 3$.

Table 6. Number of distinct equivalent keys for one input-output pair

	n=1	n=2	n=3	n=4
$d=1$	2	4	8	16
$d=2$	8	64	512	4096
$d=3$	32	1024	32768	1048576

Furthemore, the following can be observed:

(1) Ratio R of number $N(n,\ d)$ to the cardinality of the key space is:

$$R = N(n,\ d)/4^{nd} = 2^{(2d-1)n}/4^{nd} = 1/2^n.$$

Ratio R converges to 0 as $n \to \infty$.

(2) If a key size is fixed as $c = nd$, then $N(n,\ d)$ value is maxmized as $2^{2c-1} = 2^{2nd-1}$ when $n = 1$ and $d = c$, and minmized as $2^c = 2^{nd}$ when $n = c$ and $d = 1$.

4.4.2 Completely Equivalent Keys

Key pair $(\mathbf{K_1},\ \mathbf{K_2})$ is called *completely equivalent* if

$$F(\mathbf{K_1},\ I) = F(\mathbf{K_2},\ I), \quad \mathbf{K_1} \neq \mathbf{K_2} \quad \text{for all } I.$$

Lemma 11 *Let* $\mathbf{K}^{\pm} = k_1^{\pm} k_2^{\pm} \ldots k_{nd}^{\pm}$ *be a sequence of symbols from* $\{-,\ +\}$, *and* $\mathbf{K}^{\oslash} = k_1^{\oslash} k_2^{\oslash} \ldots k_{nd}^{\oslash}$ *be a sequence of symbols from* $\{0,\ 1\}$. *If*

$$\text{either } (k_i^{\pm} = -,\ \ k_i^{\oslash} = 0) \text{ or } (k_i^{\pm} = +,\ \ k_i^{\oslash} = 1) \quad \text{for all } i \text{ such that } 1 \leq i \leq nd,$$

then, pair $(\mathbf{K}^{\pm},\ \mathbf{K}^{\oslash})$ *is completely equivalent.*

Proof. If $\mathbf{K}^{\pm}$ is used, then only even events occur for all I, which implies that the input entry corresponding to $+$ is complemented and the one corresponding to $-$ is not. If $\mathbf{K}^{\oslash}$ is used, then the input entry corresponding to 1 is complemented and the one corresponding to 0 is not. Thus we have

$$F(\mathbf{K}^{\pm},\ I) = F(\mathbf{K}^{\oslash},\ I), \quad \mathbf{K}^{\pm} \neq \mathbf{K}^{\oslash} \quad \text{for all } I. \quad \square$$

The number of keys for both $\mathbf{K}^{\pm}$ and $\mathbf{K}^{\oslash}$ is 2nd. The number of pairs of completely equivalent keys is also 2nd. Thus, the number of completely equivalent keys is $2 \cdot 2^{nd}$ $(= 2^{nd+1})$. Note that the ratio of the number of completely equivalent keys to the cardinality of the key space is $2^{nd+1}/2^{2nd} = 2^{1-nd}$.

4.5 Complementation Property

Let $\overline{X}$ denote the complement of X. When X is a sequence of bits, $\overline{X}$ represents the bitwise inverse of X. In Particular, the complement of symbols $-$ and $+$ are defined as $+$ and $-$. Our proposed function f has the following complementation property:

Property 1: $f(K, I) = f(\overline{K}, \overline{I})$ for all K keys and I inputs.

It can be proved that Property 1 holds: For the tester positions of the keys, it is clear that $(K) \oplus (I) = (\overline{K}) \oplus (\overline{I})$. Since the complementation of both key K and input I does not change the parity event, the output bits at the inverter keys do not change. Notice that the above complementation property does not always hold for function F when $d \geq 2$.

We can now make the following remarks. DES cipher algorithm E has the following complementation property.

Property 2: If $O = E(K, I)$, then $\overline{O} = E(\overline{K}, \overline{I})$ for all K keys and I inputs,

Using this property and the weak keys, it is easy to find collisions for hash functions based on DES and the Meyer-Matyas chaining scheme [MOI90]. If a hash function is designed based on $C(n, d)$ circuit and a certain chaining scheme, it is desirable that $d \geq 2$.

5. Applications

5.1 Key Generation

Each circuit layer key $\in \{0, 1, -, +\}$ is coded as 2-bit information so that "00" $\leftrightarrow$ "0", "01" $\leftrightarrow$ "1", "10" $\leftrightarrow$ "$-$", "11" $\leftrightarrow$ "$+$". Since each j iteration $(1 \leq j \leq d)$ of the $L(n)$ parity layer uses a different $2n$-bit key, the $C(n, d)$ parity circuits use $2nd$-bit *internal keys*. The internal key size is determined to be as large as needed to secure the cryptosystem.

Internal keys, $\mathbf{K} = (K_1, K_2, \cdots K_d)$, are generated from *external key* K_E (supplied by the user). Thus, a key generation scheme (or, so-called key schedule calculation scheme) is needed to map from the external key to the internal one. In DES, a 768 $(= 48 \times 16)$-bit internal key is generated from a 56-bit external key by using the algorithm described in [NBS77]. However, it is said that the 56-bit key length is *not* sufficiently secure against exhaustive search attacks [Ma88] or chosen plaintext attacks [Si90]. External key size, of say 64-bits or 128-bits, is determined from the viewpoints of security, compatibility and standardization. Our circuit easily generates internal key $\mathbf{K}$ from external key K_E by applying the $C(n, d)$ circuit itself as in [Me89]. Thus, the $L(n)$ circuit layer is recursively used for both data randomization and key generation. There are a lot of variants for key generation schemes. One possibility uses the CBC mode with external key K_E and initial fixed key K_I, which is randomly chosen and shared between the sender and the receiver.

5.2 Design Principles and Criteria

The $C(n, d)$ parity circuit can be used in any *mode of operation* currently defined by ISO.

Thus, the circuit of n-bit width is directly applicable to block ciphers whose block size is n bits. Furthermore, $C(n, d)$ parity circuit can be also applied to the *F-function* of Feistel type ciphers whose block size is $2n$ bits. In any application schemes of $C(n, d)$, the block size can be flexible, say 64-bits, which is compatible with DES or FEAL.

If a block cipher is used as the hashing function in a certain chaining mode, the size of the hashed value (or *digest*) must be chosen securely. It is recommended [MOI90] that the size of the hashed value must be 128-bits to counter "meet-in-the-middle attacks". Thus, block size must be 128-bits. The $C(n, d)$ parity circuit, when $n = 128$, can achieve much faster speed while preserving security.

Recently, Quisquarter and Delescaille have found 21 equivalence key pairs for DES (64-bit input and 56-bit key),for fixed input-output pairs [QD89]. These key pairs are also called keys with collision. Their collision search algorithm is based on Pollard's ρ method. Even if we apply their collision search algorithm, it appears difficult to search for collisions in our $F(., .)$, with, say, 128-bit inputs and 128-bit external keys that generate 4nd symbols of internal keys.

5.3 Hardware Implementation

The $C(n, d)$ circuits can be implemented with high performance in hardware as well as in software.

We estimated the encryption/decryption speed of our proposed $C(n, d)$ parity circuits according to the current hardware technology. Assume that $C(n, d)$ circuits are implemented by 1.5 μm CMOS gate-arrays. The encryption speed will nearly equal the decryption speed. The amount of the encryption time is mainly taken up by comparisons and calculations for bits corresponding to the *tester cells*. These operations are carried out for each $L(n)$ layer by EXOR gates. The time for one EXOR calculation can be estimated as 4 nano seconds. Assume that data randomization and key generation are simultaneously carried out using an $L(n)$ module of $L(n)$ with feedback registers. Since one EXOR requires 3 gates and one register requires 5.5 gates, the total hardware amount can be estimated as $50n$ gates. In the $C(n, d)$ circuit, the encryption time is $d\ (4\lceil \log_2 n \rceil + 25)$ nano seconds. Thus, the encryption speed S is expressed by

$$S = \frac{n}{d\ (4\lceil \log_2 n \rceil + 25)} \times 10^9 \text{ bps.}$$

Some reasonable implementation examples are shown in Table 7. Note that FEAL-8 LSI achieves 96 Mega bps with 4 Kilo gates for the core parts.

Also note that the n and d parameters for the $C(n, d)$ parity circuit can be flexibly designed while still preserving "good" cryptographic properties. Furthermore, encryption speed S increases as width n increases while still ensuring randomness. Circuit modules with a fixed n and d have a regular structure so the structure can be expanded by multiple modular connections at a minimal cost. This advanced feature is not found in any other existing block ciphers.

Table 7. Hardware Performance

Width n (bits)	Depth d (rounds)	Speed S (M bps)	Size (K gates)
64	8	160	3.2
64	16	80	3.2
128	8	300	6.4
128	16	150	6.4

6. Conclusions

A new family of cryptographic functions called parity circuits has been presented. Furthermore, we have clarified cryptographic properties such as involution, nonlinearity, the probability of bit complementation, avalanche effect, equivalent keys. Some recommended parameter values to preserve security have been shown. In addition, we estimated the speed of the parity circuits when implemented using the current hardware technology.

Acknowledgement

The authors would like to thank Hikaru Morita at NTT for valuable discussions.

References

[BPS90] Brown, L., J. Pieprzyk and J. Seberry: "LOKI - A cryptographic primitive for authentication and secrecy applications", Abstract of Auscrypt 90, U. of New South Wales, Sydney, Australia, January (1990).

[Fe73] Feistel, H.: "Cryptography and computer privacy", Scientic American, Vol.228, No.5, pp.15-23, (1973).

[Fo88] Forre, R.: "The strict avalanche criterion: spectral properties of Boolean functions and an extended definition", Proc. of CRYPTO'88, (1988).

[KD79] Kam, J. B. and G. I. Davida: "Structured design of substitution-permutation encryption network", IEEE Trans. on Computers, Vol.28, No.10, pp.747-753, Oct., (1979).

[Ko81] Konheim, A. G.: "Cryptography: A Primer", John Wiley & Sons, New York, (1981).

[Ll89] Lloyd, S.: "Counting functions satisfying a higher order strict avalanche criterion", Proc. of Eurocrypt'89, (1989).

[Ma89] Massey J.: "An introduction to contemporary cryptology", Proc. IEEE, Vol.76, no.5, pp.533-549, May (1988).

[MS89] Meier, M. and O. Staffelbach: "Nonlinearity criteria for cryptographic functions", Proc. of Eurocrypt'89 (1989).

[MOI90] Miyaguchi, S. , K. Ohta and M. Iwata:"128-bit hash function (N-Hash)", Proc. of Securicom'90 (1990).

[NBS77] National Bureau of Standards: " Data Encryption Standard", FIPS Publication 46, U. S. Dept. of Commerce, January (1977).

[Me89] Merkle, R. C.: "A software encryption function", private communication, (1989).

[MSS88] Miyaguchi, S., A. Shiraishi, and A. Shimizu: "Fast data encipherment algorithm FEAL-8", Review of the Electrical Communication Laboratories, vol. 36-4, (1988).

[QD89] Quisquarter, J. J. and J. P. Delescaille: "How easy is collision search? New results and applications to DES", Proc. of CRYPTO'89 (1989).

[Ru86] Rueppel R. A.: "Analysis and design of stream ciphers", Springer-Verlag, Berlin, (1986).

[Si90] Simmons, G. J.:"Predictions for the 1990's ", IACR Newsletter, vol. 7, No.1, January (1990).

[WT86] Webster, A. F. and S. E. Tavares.:"On the design of S-box ", Proc. of CRYPTO'85, Springer, (1986).