J. Korean Math. Soc. **33** (1996), No. 2, pp. 259-268

EVALUATE THE DISTANCE TO THE AFFINE FUNCTIONS OF BOOLEAN VECTOR FUNCTIONS

Y. O. Sung, J. H. Jeong and C. H. Seo

1. Introduction

The Data Encryption Standard(DES)[3,8,9] was developed by an IBM team around 1974 and adopted as a national standard in 1977. Since that time, many cryptanalysts have attempted to find shortcuts for breaking the system.The entire algorithm was published in the Federal Register[1], Boolean functions from $GF(2^n)$ into $GF(2)$ are commonly found in cryptographic applications. Usually they are designed to be nonlinear and to produce a balanced output, and often one finds the additional requirement that from knowledge of the output bit it should not be possible to reliably guess one or more input bits. Consider for example DES, where the S-box[2,7] in the encryption system is originally defined as a device with n-bit input and n-bit output so that 2^n output vectors are some permutation of 2^n input vectors. Thus the Boolean vector function of an n-bit input/n-bit output S-box can be considered as an injective function,whose domain and range are both $GF(2^n)$. We can extend the definition of S-box for the case when the number n of input bits is greater than the number m of output bits. Then the Boolean vector function is from $GF(2^n)$ to $GF(2^m)$ with the additional requirement that the output should be balanced. In fact, in DES, the S-boxes with 6-input/4-output are employed. We can even further extend the definition of S-box by removing the condition that the output should be balanced. The balanced Boolean vector function is generally nonlinear. The linear function is easy to attack via differential cryptanalysis. It is generally believed that the further away from linear functions a balanced Boolean vector function is the more difficult it is to attack the S-box. In the sense, we are interested in the

Received November 26, 1994.
1991 AMS Subject Classification: 46S20, 94A60.
Key words: Hamming distance, Boolean vector function.

Boolean vector function which is furthest away from linear Boolean functions[4] in view of the Hamming distance[5].

We have outlined the criteria[6,13] that IBM used to design the S-boxes and permutation.These criteria were developed specifically to thwart attacks based on differential cryptanalysis. A measure of the success of IBM's approach to the design of the S-boxes and permuation is the enormous amount of chosen plaintext (in excess of 10^{15} bytes) required by Biham and Shamir's attack[6].

In this paper, the lower and the upper bounds on the maximum distance to the Affine functions of Boolean vector functions and balanced Boolean vector functions from $GF(2^n)$ to $GF(2^m)$ are derived. In section 2, we state some necessary notation and definition,extend distance concept to Boolean vector functions, in section 3,the Boolean vector functions of S-boxes in DES are investigated and the distances of the Boolean vector functions of the 32 4-bit input/4-bit output S-boxes are evaluated. In the appendix, the functions are tabulated and computed the distance to Affine functions of each of the 32 Boolean vector functions.The implementation has been done on a SUN SPARC-2 station using C-language.

2. Preliminaries

We state here some necessary notation and definition in order to compute the distance to Affine functions and construct DES like S-boxes. Let $\oplus$ denote the addition over $GF(2^n)$ or the bit-wise exclusive-or, $|\cdot|$ denotes the cardinality of a set. $wt()$ denotes the Hamming weight function,$C_i^{(n)}$ denotes an n dimensional vector with Hamming weight 1 at the i-th position.

DEFINITION 2.1. The Hamming weight of a word $C_0, C_1, \cdots, C_{n-1}$, $C_i \in GF(2)$ is defined as the sum of the Hamming weights of its digits, where

$$wt(C_i) = \begin{cases} 0 & \text{if} \quad C_i = 0 \\ 1 & \text{if} \quad C_i \neq 0. \end{cases}$$

Let f and g be Boolean functions from $GF(2^n)$ which is n-dimensional vector space over $GF(2)$ to $GF(2)$: $GF(2^n) \rightarrow GF(2)$. The Hamming distance $d(f,g)$ between two Boolean functions f and g is defined as

$$(1) \qquad d(f,g) = wt[f+g].$$

Now, we can extend this distance concept to Boolean vector functions, namely F and G. Let F and G be m-dimensional vectors such that each component of F and G is same Boolean function in $GF(2^n)$. Then the distance between F and G can be defined as

$$(2) \qquad d(F,G) = |\{x \in GF(2^n) | F(x) + G(x) \neq 0\}|$$

To find the distance from a Boolean function f to Affine functions is an interesting subject in the study of Boolean functions. Here the distance D_f from a function f to Affine functions is defined as

$$(3) \qquad D_f = \min_{\ell \in L_n} \{d(f,\ell)\}.$$

where L_n is the set of all Affine Boolean functions in $GF(2^n)$. Similarly we can define the Hamming distance to Affine functions of a Boolean vector function. Let $F : GF(2^n) \to GF(2^m)$ be a Boolean vector function. The Hamming distance D_F of a Boolean vector function F to Affine functions is defined as

$$(4) \qquad D_F = \min_{L} \{d(F,L)\},$$

where $L = (\ell_1, \ell_2, \cdots, \ell_m)$ is an Affine Boolean vector function such that $\ell_i \in L_n$, for all $i = 1, 2, \cdots, m$.

For a Boolean vector function $F = \{f_1, f_2, \cdots, f_m\}$, the definition in(4) can be equivalently expressed as

$$(5) \qquad D_F = 2^n - \max_{L} \{wt[\prod_{i=1}^{m} (f_i + \ell_i)]\}.$$

Now, let us introduce the parameter $d(n,m)$. Let $F_{n,m}$ be the set of all Boolean vector functions from $GF(2^n)$ to $GF(2^m)$. Then $d(n,m)$ is defined as

$$(6) \qquad d(n,m) = \max_{F \in F_{n,m}} \{D_F\}.$$

Especially when $m = 1$, to find out $d(n,1)$ is equivalently to find out the maximum of the weights of coset leaders in a first-order Reed-Muller code[5]$R(1,n)$ of length 2^n.

For even values of n, the Boolean functions which achieves $d(n,1)$ are called bent functions[4] and it is known that

$$d(n,1) = 2^{n-1} - 2^{\frac{n-2}{2}}, \quad n \ is \ even. \tag{7}$$

For odd values of n, we have derived the following lower bound.

THEOREM 2.1. $d(2m+1,1) \geq 2d(2m,1)$.

Proof. Let $\overline{x} = (x_1, x_2, \cdots, x_{2m+1})$ be in $GF(2^{2m+1})$. Set $\overline{y} = (x_2, \cdots, x_{2m+1})$. Then any Boolean function $f(\overline{x})$ in $GF(2^{2m+1})$ can be expressed as

$$f(\overline{x}) = x_1 g(\overline{y}) + h(\overline{y}),$$

and any Affine function $\ell(\overline{x})$ in $GF(2^{2m+1})$ can be expressed as

$$\ell(\overline{x}) = c_1 x_1 + \ell_1(\overline{y}),$$

where g and h are Boolean functions in $GF(2^{2m})$.c_1 is in $GF(2)$ and ℓ_1 is an Affine Boolean function in $GF(2^{2m})$.

THEOREM 2.2. $d(2m+1,1) \leq 2^{2m} - 2^{m-1} - 2$.

Proof. If $d(2m+1,1) > 2(2^{2m-1} - 2^{m-1}) = 2d(2m,1)$ then it means that for any Affine Boolean function ℓ_1 in $GF(2^{2m})$ and any number c_1, there exists some functions g and h such that

$$wt[h + \ell_1] + wt[g + h + \ell_1 + c_1] > 2(2^{2m-1} - 2^{m-1}). \tag{8}$$

Since inequality(8) should be satisfied for any c_1, we may say that for any ℓ_1

$$wt[h + \ell_1] \leq 2^{2m-1} \tag{9}$$

$$wt[g + h + \ell_1 + c_1] \leq 2^{2m-1}. \tag{10}$$

From(8)-(10), it can be concluded that $D_h > 2^{2m-1} - 2^m$ and $D_{h+g} > 2^{2m-1} - 2^m$. And in this case, the maximum of the LHS of (8) is no greater than $(2^{2m-1} - 2^{m-1} - 1) + (2^{2m-1} - 1)$. Therefore

$$d(2m+1,1) \leq 2^{2m} - 2^{m-1} - 2. \qquad Q.E.D.$$

3. Evaluate Distance to the Affine Function of Boolean Vector Functions with Balance Property

As mentioned in the introduction, the Boolean vector functions of a S-box must satisfy the balance property. Let $F : GF(2^n) \to GF(2^m)$ be the Boolean vector function of an S-box with n-bit input and m-bit output. The balance property implies that the number of $\overline{x} \in GF(2^n)$ such that $F(\overline{x}) = \overline{c}$ is exactly 2^{n-m} for every $\overline{c} \in GF(2^m)$.

In this section, we are going to investigate into the distance from the Boolean vector functions to Affine functions. Let's define the maximum distance parameter $D(n,m)$.

DEFINITION 3.1.

$$D(n,m) = max_F\{min_L\{d(f,\ell)\}\},$$

where F is a balanced Boolean vector function from $GF(2^n)$ to $GF(2^m)$.

As in section 2, let us first consider the case when $m = 1$. For the even values of n, it is obvious that there exists some balanced Boolean function which is $2^{\frac{n-2}{2}}$ apart from some bent functions.

Thus we have

$$D(2m,1) \geq 2^{2m-1} - 2^m. \tag{11}$$

Note that when $m = 2$, the equality in (11) is achieved, i.e. $D(4,1) = 4$.

Therefore we can have the following lower bound.

THEOREM 3.1. *For even n and $m \leq \frac{n}{2}$,*

$$D(n,m) \geq (2^m - 1)(2^{n-m} - 2^{\frac{n}{2}-m+1}).$$

The proof of Theorem 3.1 is given by showing the existence of a specific Boolean vector function of S-box which yields $(2^m - 1)(2^{n-m} - 2^{\frac{n}{2}-m+1})$ as the distance to Affine function. Particularly (for $m = \frac{n}{2}$), here is the construction.

First, construct m bent functions using Nyberg's Construction method[7]. i.e., let $\overline{x_1} = (x_1, x_2, \cdots, x_{2m-1})$ and $\overline{x_2} = (x_2, x_4, \cdots, x_{2m})$, then the Nyberg's m bent functions $f_i(\overline{x})$, $i = 1, 2, \cdots, m$ is given by

$$f_i(\overline{x}) = A^{i-1}(\overline{x_1}) \circ \overline{x_2}.$$

where A is the state transition function of a linear feedback shift register of length m with a linear feedback polynomial.

Next, modify the each f_i as $g_i = f_i + h_i$ where

$$h_i(\overline{x}) = x_{2i} \prod_{j=0}^{m-1} (x_{2j+1} + 1).$$

Then $wt[h_i(\overline{x})] = 2^{m-1}$ for all i,and in the position where $h_i(\overline{x}) = 1$, $\overline{x_1}$ is automatically $\overline{0}$, so that $f_i(\overline{x}) = 0$. Thus the modified function g_i is balanced and in fact $G = (g_1 \cdots, g_m)$ is also balanced,and finally it is easy to see that

$$d(\sum_{i=1}^{m} c_i(g_i + x_{2i})) = 2^{2m-1} - 2^m$$

for any combination $(c_1, c_2, \cdots, c_m)$. Thus $D_G = (2^m - 1)(2^m - 2)$.

EXAMPLE 3.1. When n is equal to 4, m is equal to 2, and

$$f_1 = x_1x_2 + x_3x_4, \quad f_2 = x_1x_2 + x_2x_3 + x_1x_4.$$

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
f_1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0
f_2	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	1
g_1	0	0	0	1	1	1	0	1	0	0	0	1	1	1	1	0
g_2	0	1	0	0	0	1	1	1	0	1	0	1	1	0	0	1
$g_1 + x_2$	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1
$g_2 + x_4$	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0
$g_1 + g_2 + x_2 + x_4$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1

For odd n and $m < \frac{m}{2}$, we can have a similar lower bound as follows.

THEOREM 3.2. *For $n = 2k + 1$ and $m \leq k$,*

$$D(n, m) \geq (2^m - 1)(2^{n-m} - 2^{\frac{n+3}{2} - m}).$$

In DES, there are 8 different S-boxes employed and they are all with 6-bit input and 4-bit output. But in fact, each Boolean vectorfunction f_i of an S-box S_i can be decomposed of $F_i(\overline{x}) = (x_0 + 1)(x_5 + 1)G_{i0}(\overline{y}) + (x_0 + 1)x_5 G_{i1}(\overline{y}) + x_0(x_5 + 1)G_{i2}(\overline{y}) + x_0 x_5 G_{i3}(\overline{y})$ where $\overline{y} = (x_1, x_2, x_3, x_4)$ and each of the four functions $G_{ij}(\overline{y})$, $j = 0, 1, 2, 3$, can also serve as a balanced Boolean vector function of the 4-bit input/4-bit output S-box. In other words, the 6-bit input/4-bit output S-boxes in DES are the combination of four 4-bit input/4-bit output S-boxes. In the appendix, the functions are G_{ij}, $i = 5, 6$, $j = 0, 1, 2, 3$ tabulated. Also we computed the distance D_{ij} to Affine functions of each of the 32 Boolean vector functions G_{ij}.

4. Appendix

The Boolean vector function $F_i(\overline{x})$ of the i-th S-box S_i is given by

$$F_i(\overline{x}) = \begin{cases} G_{i0}(x_1, \cdots, x_4) & x_0 = 0 \quad x_5 = 0 \\ G_{i1}(x_1, \cdots, x_4) & x_0 = 0 \quad x_5 = 1 \\ G_{i2}(x_1, \cdots, x_4) & x_0 = 1 \quad x_5 = 0 \\ G_{i3}(x_1, \cdots, x_4) & x_0 = 1 \quad x_5 = 1 \end{cases}$$

and $G_{ij} = \{g_{ij0}, g_{ij1}, g_{ij2}, g_{ij3}\}$. The following is the table of g_{ijk} and $D_{ij} = D_{G_{ij}}$.

$$\begin{aligned} g_{500} =& x_1 + x_4 + x_1x_3 + x_2x_3 + x_3x_4 + x_1x_3x_4 + x_2x_3x_4 \\ g_{501} =& x_2 + x_3 + x_4 + x_1x_3 \\ g_{502} =& 1 + x_1 + x_3 + x_4 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4 + x_1x_2x_3 \\ & + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 \\ g_{503} =& x_2 + x_1x_3 + x_1x_4 + x_2x_4 + x_3x_4 + x_1x_2x_4 + x_2x_3x_4 \\ D_{50} =& 7 \end{aligned}$$

$$\begin{aligned} g_{510} =& 1 + x_1 + x_2 + x_3 + x_1x_2 + x_3x_4 + x_1x_2x_4 + x_1x_3x_4 \\ g_{511} =& 1 + x_3 + x_4 + x_1x_2 + x_1x_3 + x_2x_4 + x_3x_4 + x_1x_2x_3 \\ & + x_2x_3x_4 \\ g_{512} =& 1 + x_1 + x_2 + x_1x_3 + x_2x_4 + x_3x_4 + x_1x_3x_4 \\ g_{513} =& x_1 + x_4 + x_2x_3 + x_3x_4 + x_1x_2x_4 + x_1x_3x_4 \\ D_{51} =& 8 \end{aligned}$$

$$\begin{aligned}
g_{520} &= x_1 + x_2 + x_2x_3 + x_3x_4 + x_1x_2x_3 \\
g_{521} &= 1 + x_2 + x_3 + x_4 + x_1x_2 + x_1x_3 + x_3x_4 \\
&\quad + x_1x_2x_3 + x_2x_3x_4 \\
g_{522} &= x_1 + x_2 + x_4 + x_1x_2 + x_1x_3 + x_1x_2x_4 + x_1x_3x_4 \\
g_{523} &= x_1 + x_3 + x_1x_2 + x_2x_4 + x_1x_2x_3 + x_1x_3x_4 \\
D_{52} &= 8
\end{aligned}$$

$$\begin{aligned}
g_{530} &= 1 + x_1 + x_2 + x_1x_4 + x_2x_4 + x_3x_4 + x_1x_2x_3 \\
&\quad + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 \\
g_{531} &= x_1 + x_3 + x_1x_2 + x_2x_3 + x_2x_4 + x_1x_2x_3 \\
g_{532} &= 1 + x_2 + x_3 + x_4 + x_1x_2 + x_1x_4 + x_1x_2x_4 \\
g_{533} &= 1 + x_1 + x_3 + x_4 + x_1x_3 + x_1x_2x_3 + x_1x_2x_4 \\
D_{53} &= 8
\end{aligned}$$

$$\begin{aligned}
g_{600} &= 1 + x_1 + x_4 + x_1x_2 + x_2x_3 + x_3x_4 + x_2x_3x_4 \\
g_{601} &= 1 + x_1 + x_2 + x_3 + x_4 + x_1x_3 + x_2x_4 \\
g_{602} &= x_3 + x_1x_2 + x_2x_4 + x_1x_2x_4 + x_1x_3x_4 \\
g_{603} &= x_2 + x_4 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_2x_3x_4 \\
D_{60} &= 9
\end{aligned}$$

$$\begin{aligned}
g_{610} &= 1 + x_1 + x_2 + x_3 + x_1x_2 + x_2x_4 + x_1x_2x_3 \\
g_{611} &= x_1 + x_2 + x_3 + x_4 + x_1x_3 + x_2x_4 + x_1x_3x_4 + x_2x_3x_4 \\
g_{612} &= 1 + x_3 + x_1x_2 + x_1x_4 + x_2x_4 + x_3x_4 + x_1x_2x_4 + x_1x_3x_4 \\
g_{613} &= x_2 + x_4 + x_1x_2 + x_1x_3 + x_3x_4 + x_1x_3x_4 \\
D_{61} &= 8
\end{aligned}$$

$$\begin{aligned}
g_{620} &= 1 + x_1 + x_2 + x_1x_2 + x_2x_3 + x_2x_4 + x_3x_4 + x_2x_3x_4 \\
g_{621} &= x_1 + x_3 + x_4 + x_1x_2 + x_1x_3 + x_2x_4 + x_3x_4 \\
&\quad + x_1x_2x_4 + x_1x_3x_4 \\
g_{622} &= x_1 + x_2 + x_3 + x_4 + x_1x_2x_4 \\
g_{623} &= 1 + x_2 + x_4 + x_1x_2 + x_1x_3 + x_2x_4 + x_3x_4 + x_1x_2x_3 \\
D_{62} &= 8
\end{aligned}$$

$$
\begin{aligned}
g_{630} =& x_1 + x_2 + x_1x_3 + x_2x_4x_3x_4 + x_1x_2x_4 + x_1x_3x_4 \\
g_{631} =& 1 + x_1 + x_2 + x_3 + x_4 + x_1x_3 + x_1x_2x_3 \\
g_{632} =& x_1 + x_3 + x_4 + x_1x_4 + x_2x_4 + x_1x_3x_4 \\
g_{633} =& x_1 + x_2 + x_4 + x_2x_4 + x_3x_4 \\
D_{63} =& 8
\end{aligned}
$$

References

1. *Data Encryption Standard Federal Information Processing Standards Publication*, 46, National Bureau. of Standards, 1977.
2. A. F. Webster and S. E. Tavares, *On the design of S-boxes*, Proc. of Crypto. **'86** (1987), 523-534.
3. A. Shamir,, *On the security of DES*, Proc. of Crypto. **'85** (1986), 280-283.
4. O. S. Rothaus, *On bent functions*, J. of Combin. Theory, **20A** (1976), 300-305.
5. D. C. Gorenstein and N. Zierler, *A class of Error-Correcting Codes in $GF(p^m)$*, J. Soc. Indus. Appl. Math. **9** (1961), 207-214.
6. E. Biham and A. Shamir,, *Differential Cryptanalysis of DES-like Cryptosystems*, J. Cryptology **4** (1991), 3-72.
7. K. Nyberg, *Perpect Nonlinear S-boxes*, Proc. of Eurocrypto. **'86** (1986), 284-286.
8. C. H. Meyer, *Ciphertext/Plaintext and Ciphertext/Key Dependence vs Number of Rounds for the Data Encryption Standard*, National Computer Conference (1978), 11-19.
9. E. Biham and A. Shamir, *Differential Cryptanalysis of the Full 16-round DES*, Proc. of Crypto. **'92** (1992), 487-496.
10. E. Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Technical Report ♯753, Computer Science Department, Technion-Istrael Institute of Technology, 1992.
11. E. Biham,, *personal communication*, 1993.
12. K. Kim, S. Park, and S.Lee, *Reconstruction of S2DES S-boxes and their Immunity to Differential Cryptanalysis*, Proc. of the 1993 Korea-Japan workshop on Information Security and Cryptography, Seoul, Korea 24-26 (1993), 282-291.
13. W. Meier, O. Staffelbach, *Nonlinerarity Criteria for Cryptographic Functions*, Proc. of Eurocrypt **'89** (1989.), 549-562.

Y. O. Sung
Department of Mathematics Education
Kongju University
Chungnam 314-701, Korea

J. H. Jeong
Department of Mathematics
Seonam University
Namwon 590-170, Korea

C. H. Seo
Institute Defence of Information Systems
Seoul 130-650, Korea