JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 18, 349-365 (2002)

An Extended DES

YI-SHIUNG YEH AND CHING-HUNG HSU*
Institute of Computer Science and Information Engineering
**Institute of Computer and Information Science*
National Chiao Tung University
Hsinchu, 300 Taiwan
E-mail: ysyeh@csie.nctu.edu.tw

The original S-boxes and algorithms of DES are designed to resist differential attack[11]. We propose eight more new S-boxes with the same cryptographic properties as S-boxes in DES. These 16 S-boxes are used to construct the extended DES, which is double the size of the original DES. The differential and linear cryptanalyses of the extended DES are given. The complexities of the two attacks are found to be 2^{112} and 2^{142} respectively.

Keywords: cryptography, DES, extended DES, differential attack, linear attack, S-box

1. INTRODUCTION

DES is one of the most popular block ciphers. It is a block cipher encrypting 64 bits of data block with a 56-bit key size. The small key size and increased computing power of modern computers make DES unsafe even under exhaustive search attack. Therefore, a cipher based on DES with a larger key size is necessary.

Each S-box consists of four boolean functions mapping from 6-bit input data to 4-bit output data. It is the major part of DES which defends against cryptanalysis. S-boxes are designed to defend against differential attack [11]. In addition to the design criteria that have been made public, many cryptographic properties of S-box have been studied [1, 4, 5, 10].

Differential attack [7] makes use of the exclusive-or difference of plaintext and cipher pairs. It estimates the probability that certain a plaintext difference will result in a certain ciphertext difference, by estimating the probability of intermediate difference patterns in the DES algorithm. This exclusive-or difference sequence of plaintext, intermediate state, and ciphertext is the characteristic. A plaintext pair is the right pair for a characteristic if their xor sequence in encryption is the same as the characteristic. Right pairs could be used to analyze the correct key value but the analysis of wrong pairs suggests random values. To reduce the complexity of differential attack, one has to find a high probability characteristic.

Biham and Shamir showed that DES can be broken by a differential attack involving 2^{47} chosen plaintexts or 2^{55} known plaintexts. Many modified variants of DES result in a weaker DES-like cipher [7]. DES with independent sub keys can be broken by a differential attack involving 2^{60} chosen plaintexts or 2^{61} known plaintexts.

Received March 13, 2001; accepted August 22, 2001.
Communicated by Ja-Ling Wu.

To increase the key size without lossing strength against differential attack, eight more S-boxes are proposed. They have the same cryptographic properties and design criteria of the original S-boxes. These 16 S-boxes are used to construct the extended DES, which has a 112-bit key size.

2. PROPERTIES OF S-BOXES

An approach to generating new S-boxes is to construct S-boxes using 4 small boolean functions, which are called as row function defined in section 2.1. Each row function is a boolean function mapping 4-bit data to 4-bit output data. A new S-box similar to the original S-box, for example S-box i + 8, can be constructed by 4 row functions that have the same cryptographic properties as the row functions of S-box i, where i = 1, …, 8.

An exhaustive search for row functions can be used to construct 8 more new S-boxes the same as the original S-boxes in terms of their cryptographic properties.

2.1 S-Boxes

Let x be a bit string, and let |x| denote the number of bits in x. Three operations, $\vee$, $\wedge$, and $\oplus$, are defined as the bit-wise "OR," "AND" and "Exclusive-OR," respectively. Throughout this paper we will assume that each value is a bit string of length n unless otherwise indicated. Let h be a hamming function; i.e., h(x) is the number of 1's in x. A value v is called a unit if h(v) = 1. The i$\underline{\text{th}}$ unit is denoted as U_i if U_i is a unit and the i$\underline{\text{th}}$ bit of U_i is 1. Let N = {0, 1, …, 15}.

An S-box in DES is a function f from X to Y, where X and Y are the sets of bit strings of lengths 6 and 4, respectively, satisfying the following criteria [11]:

a. Each row is a permutation on N and is called row function. There are 4 rows in an S-box.
b. It is a nonlinear function and not an affine function.
c. $h(f(x)\oplus f(x\oplus U_i)) \geq 2$, where i = 1, …, 6.
d. $\forall x \in X$, $h(f(x)\oplus f(x\oplus 001100)) \geq 2$.
e. $\forall x \in X$, $f(x) \neq f(x\oplus 11ef00)$ for e, f $\in$ {1, 0}.
f. If the value of any single input bit is fixed, then the number of inputs for which any fixed output bits has the value 0 (or 1) is "close" to $2^5/2 = 16$. In other words, an S-box is a 1-order 0-1 balance in tolerance 3 function as will be discussed latter.

These criteria are needed to defend against differential cryptanalysis in DES. Any violation will weaken DES.

2.2 Notations and Definitions

Let f be a boolean function from X to Y, where X and Y are the sets of bit strings of lengths n and m, respectively. Let f = (f_1, …, f_m), where f_i is a mapping from X to {0, 1}, for *i* = 1, …, m. We shall define the following terms.

1) Output sequential function(Seq)[1]: $P_i(\mathrm{Seq}(f_j)) = f_j(i)$ for some j, i, $x \in X$, $P_i(x)$ is the ith bit in x. That is, $\mathrm{Seq}(f_j)$ represents the output sequence bit string of f_j.
2) An affine function (af) from X to {0, 1}: $af(x) = h(a \wedge x) + c \bmod 2$, where c is a constant in {1, 0}, $x \in X$, and a is a constant bit string. There are a total of 2^{n+1} affine functions.[1]
3) Nonlinearity (N) of f_i [4]: $N(f_i) = \min_{af} h(\mathrm{Seq}(f_i)\oplus \mathrm{Seq}(af))$.
4) Global nonlinearity (GN) of f[5]: $GN(f) = \sum_i N(f_i)$.
5) Completeness[2]: The boolean function f is complete if SAC-map[i, j]$\neq$0 for every i, j where SAC shall be defined later.
6) 0-1 Balance[1]: If $\sum_x f_i(x) = 2^{n-1}$ for $i = 1, \ldots, m$, then f is called 0-1 balance.
7) Linear structure[3]: If $f_i(x)\oplus f_i(x\oplus c)$ is a constant for all x and for any i, i.e., $\sum_x f_i(x)\oplus f_i(x\oplus c) = \begin{cases} 0 \\ 2^n \end{cases} \forall i,$ then c is called a linear structure of f. If c is a linear structure, the value $f(x)\oplus f(x\oplus c)$ is the same for all x. A linear structure in S-boxes can be used to attack DES. If the value $f(x)\oplus f(x\oplus c)$ is the same for most x, then c is a linear-like structure. Evertse found that a linear-like structure in S-boxes can be used to break 6 round DES.
8) Strict Avalanche Criterion[2]: If $\sum_x f_j(x)\oplus f_j(x\oplus U_i) = 2^{n-1}$ for $i = 1, \ldots, n$, $j = 1, \ldots, m$, then f is an SAC function.
9) An SAC-map is an n X m matrix, and each entry SAC-map[i, j] $= \sum_x f_j(x)\oplus f_j(x\oplus U_i)$ for $i = 1, \ldots, n$, $j = 1, \ldots, m$.
10) A perfect SAC-map is an SAC-map, where SAC-map[i, j] = 2^{n-1} for all i, j. If an SAC-map of f is a perfect SAC-map, then f is an SAC function [2]. From the design criteria of S-boxes, it is impossible to find a perfect SAC-map.

However, an S-box close to a perfect SAC-map is still novel. We define three SAC-map estimate distances to indicate the SAC degree of a function:

11) Global SAC-map distance(GD): $GD(f) = ((\sum_{i,j}(\mathrm{SAC\text{-}map}[i, j] - 2^{n-1})^2)/(m*n))^{1/2}$.
12) Input SAC-map distance(ID): $ID(f) = (\sum_i ((\sum_j \mathrm{SAC\text{-}map}[i, j]) - m*2^{n-1})^2)/n)^{1/2}$.
13) Output SAC-map distance(OD): $OD(f) = (\sum_j (\sum_i (\mathrm{SAC\text{-}map}[i, j]) - n*2^{n-1})^2)/m)^{1/2}$.
14) 0-1 Balance in the tolerance T: The function f satisfying $2^{n-1} - T \leqq \sum_x f_i(x) \leqq 2^{n-1} + T$, for $i = 1, \ldots, m$, is called 0-1 balance in the tolerance T if T is the minimal value needed to satisfy the above inequality equation. Obviously, f is 0-1 balance if T = 0. The tolerance T is a value used to show the 0-1 balance degree of a function.
15) Linear structure in the tolerance T_c: A constant c is called a linear structure in the tolerance T_c if $T_c = \max_i\{|\sum_x f_i(x)\oplus f_i(x\oplus c) - 2^{n-1}|\}$. T_c is the linear structure in the tolerance for c. The linear structure in the tolerance of f is T if $T = \max_c\{T_c\}$. This linear structure tolerance value of f is indicates how close a linear-like structure is that can be found.

16) Mask(x, M) returns a string which is the substring obtained by discarding some bits of x. For any bit whose value is 1 in M, the corresponding bits of x are discarded.

Take(x, M) returns a string which is the substring obtained by discarding some bits of x. For any bit whose value is 0 in M, the corresponding bits of x are discarded.

For example, the underlining bits in first argument are discarded:

Mask(101$\underline{001}$, 000$\underline{111}$) = 101 and Take($\underline{101}$001, $\underline{000}$111) = 001.

t-order sub function of f:

Given two values M, M’, where h(M) = t |M| = n and |M’| = t, we can define a function $f^t_{M,M'}$: $2^{n-t} \rightarrow 2^m$;

$f^t_{M,M'}(x') = f(x)$, where mask(x, M) = x’, take(x, M) = M’.

t-order decomposition of f:

Given a value M, where h(M) = t, the t-order decomposition of f is a set of functions $\{f^t_{M,M'} \,|M` \in 2^t\}$. Each element in this set is a t-order sub function of f with mask M.

17) A function f with the t-order property B: If all t-order sub functions of f are functions satisfying the property B, then f is a function with the t-order property B. For example, a t-order complete boolean function is a boolean function all of whose t-order sub functions are complete. Other t-order properties are also defined in the same way. (Note that B in “property B” is the name of the property.)

18) A DDT map is the difference distribution Table [7] of a function. It is an output xor distribution map. For any a ∈ X, b ∈ Y, DDT[a, b] is the number of elements in the set, $\{z|\ z \in X, \ni x \in X$ such that $x \oplus z = a$ and $f(x) \oplus f(z) = b\}$. Moreover, two functions *f1*, *f2* are semi-similar if the zero columns of their DDT maps are the same, i.e., $DDT_{f1}[a, 0] = DDT_{f2}[a, 0]$ for any a.

2.3 Coupling Rows and Construction of S-boxes

An S-box is composed of 4 rows and a special 2-order sub function of the S-box. The four rows of an S-box is a 2 order decomposition of the S-box where m = 100001. The row(s, i) represents the *i*th row of S-box #s, for *i* = 1, ..., 4.

Four special 1-order functions can be determined by ignoring the leftmost or the rightmost bit where m = 100000 or 000001 in the mask function. They are called the coupling-rows composed by two rows. The number of input bits in an S-box is 6 while the number of input bits in the corresponding coupling rows is 5. The couple-row(s, *i*, *j*) represents the coupling-rows composed by *i*th and *j*th row functions in the S-box #s. The possible values for (*i*, *j*) in the couple-row(s, *i*, *j*) are (1, 3), (2, 4), (1, 2), and (3, 4) which are also called couple-rows1, couple-rows2, couple-rows3, couple-rows4, respectively. Two types of the coupling rows can be defined. The Type-I of the coupling-rows which are the 1-order decomposition of S-box #s where m = 100000 are the couple-row(s, 1, 3) and the couple-row(s, 2, 4). The Type-II of the coupling-rows which are the 1-order decomposition of S-box #s where m = 000001 are the couple-row(s, 1, 2) and the couple-row(s, 3, 4). The 5th input bit and 1st input bit are used to indicate the selected row in Type-I and Type-II coupling rows respectively. Output is decided by other input bits.

An S-box can be decomposed into the two coupling-rows. If we want to construct a S-box similar as DES S-box 1,for example, the basic approach to construct the S-box is to select rows with similar properties as S-box 1 from the 16!, the number of permutations on N, search spaces. These rows are used to construct the Type-I of the coupling rows which still have similar properties as the corresponding Type-I coupling-rows of S-box 1. The well-selected rows to construct the Type-I of the coupling-rows are very important because the construction of an S-box is based on the coupling-rows. After an S-box is formed, two corresponding Type-II coupling rows are occurred. Next we will test the two Type-II coupling rows to see if they have similar properties as Type-II coupling rows of S-box 1. The two types of the coupling-rows must all have similar properties as in S-box 1, otherwise the S-box taken should not be considered and we have to find another one by repeating the previous processes.

In the section 2.2, we define many mathematical properties that can be applied to a non-specific boolean function. The definitions 1, …, 15 can be used to test whether two row functions or coupling-rows functions have similar properties.

2.4 New S-box Properties

8 new S-boxes are proposed in and listed in the following tables. Table 1 shows the similarity between the new S-boxes and original S-boxes. S-box$_{i+8}$ is cryptographically similar to S-box$_i$, i = 1, …, 8, and they are also semi-similar. The new S-boxes are listed in Table 2 which is also a comparison between the original and new S-boxes.

Table 1. The similarity of new and original S-boxes.

Our design	Original	LST	B1	B2	C order	GD	ID	OD	L_1	L_2	L_3	L_4	GL	None-zero rate
S-box #9	S-box #1	20	3	3	1	9.31	32.25	46.56	18	20	22	18	78	79.4%
S-box #10	S-box #2	28	3	3	1	11.22	35.81	56.32	22	20	18	18	78	78.6%
S-box #11	S-box #3	24	3	4	1	12.65	41.70	63.62	18	22	20	18	78	79.6%
S-box #12	S-box #4	12*	3	2*	2*	8.16*	32.66	44.00	22	22	22	22	88	68.5%
S-box #13	S-box #5	20	3	2*	1	9.90	35.81	55.32	22	20	18	20	80	76.5%
S-box #14	S-box #6	24	3	3	1	11.31	38.85	59.53	20	20	20	20	80	80.4%
S-box #15	S-box #7	24	3	3	1	12.17	43.45	65.18	18	22	14	20	74	77.2%
S-box #16	S-box #8	20	3	2*	1	10.95	38.71	56.21	22	20	20	22	84	77.1%

LST: Linear structure tolerance.
B1: First order 0-1 balance tolerance.
B2: Second order 0-1 balance tolerance.
C order: Maximum order of completeness.
GD: Global SAC-map distance.
ID: Input SAC-map distance.
OD: Output SAC-map distance.
L*i*: Nonlinearity of output bit *i*.
GL: Global nonlinearity.
None-zero rate: Percentage of none zero entry in the DDT map.

Table 2. Extended S-boxes.

3 0 9 7 15 12 6 11 14 13 2 1 5 10 8 4 0 3 5 8 9 15 12 6 13 10 11 7 14 4 2 1 15 5 12 2 0 11 9 14 4 3 1 8 10 6 7 13 9 15 0 5 10 6 3 8 2 12 13 11 4 1 14 7 **S-box #9**	1 10 15 12 8 3 6 5 13 4 0 7 14 9 11 2 4 7 10 0 15 9 1 12 8 14 3 13 5 2 6 11 2 5 4 10 7 12 9 3 11 8 14 1 13 6 0 15 7 0 9 3 4 15 10 6 2 13 5 14 11 8 12 1 **S-box #10**
15 4 12 1 5 10 2 13 3 8 6 11 0 7 9 14 6 13 15 2 8 4 5 11 0 7 9 12 3 10 14 1 4 13 15 10 2 1 8 6 14 3 0 5 11 12 7 9 13 3 1 4 11 14 2 8 7 10 12 15 0 5 9 6 **S-box #11**	10 7 15 12 4 2 1 11 0 13 5 3 9 14 6 8 6 13 12 0 1 7 11 14 3 8 9 15 10 4 5 2 4 1 2 11 15 12 8 6 7 10 14 5 0 9 13 3 1 11 7 14 12 0 2 5 13 6 4 9 3 10 8 15 **S-box #12**
4 7 1 12 14 11 8 2 13 10 6 9 0 5 3 15 13 0 2 7 4 14 1 11 3 12 5 10 15 9 8 6 10 1 12 11 9 2 7 14 6 13 15 4 5 8 0 3 7 11 9 4 2 1 14 13 0 6 10 3 12 15 5 8 **S-box #13**	2 14 15 0 12 11 9 5 4 13 8 3 1 6 7 10 12 5 9 10 7 0 2 15 3 6 14 13 8 11 4 1 12 2 3 14 15 4 10 9 11 1 5 8 6 13 0 7 1 15 12 5 10 9 7 2 6 8 0 14 3 4 13 11 **S-box #14**
13 2 4 7 3 12 8 1 0 15 14 9 5 10 11 6 3 8 14 13 9 2 5 11 15 4 0 10 12 7 6 1 2 11 8 13 15 0 4 14 12 5 1 6 10 3 7 9 13 6 1 8 2 11 14 5 10 9 12 3 7 4 0 15 **S-box #14**	12 2 10 7 1 4 15 8 11 14 0 9 13 3 6 5 2 1 9 4 7 14 12 11 13 8 3 15 10 5 0 6 1 11 15 8 4 13 2 7 14 0 5 6 3 10 9 12 11 13 6 1 8 2 5 14 4 7 10 12 15 9 3 0 **S-box #16**

3. THE EXTENDED DES

The extended DES is very similar to DES. It has exactly the same data flow and is based in the concept of DES. Eight more S-boxes are used in the extended DES to double the size. Some modifications must be made to the P-box and key scheduling algorithms.

3.1 Algorithm of the Extended DES

The extended DES encrypts a 128-bit data block using 112 key bits. All data bits go through an initial permutation (IP block in Fig. 1). The data bits are then split into two 64-bit data blocks that are right and left data blocks. These two data blocks then go through 32 identical rounds. As shown in Fig. 1, there is no swap of the two data blocks in the last round. After the last round, the two data blocks are combined to form a 128-bit block. The result will be through the inverse initial permutation.

In each round, the right data block and 96-bit sub-key (R_{i-1} and K_i in Fig. 2) are combined by a round function called F. The output of F is then combined with the left part data block by an xor operation. The two data blocks swap in the next round.

Fig. 2 shows the function F in detail. The 64-bit right data block is expanded to 96 bits by means of expansion permutation (the expansion block shown in Fig. 2). After combining with the 96-bit sub-key, the 96-bit data is distributed to all 16 S-boxes as input. Each S-box has 4 output bits. Therefore, 64-bit data is used in the next step. This is P-box permutation. Note that the elliptic shadowed area shown in Fig. 2 is a round of function F shown in Fig. 1.

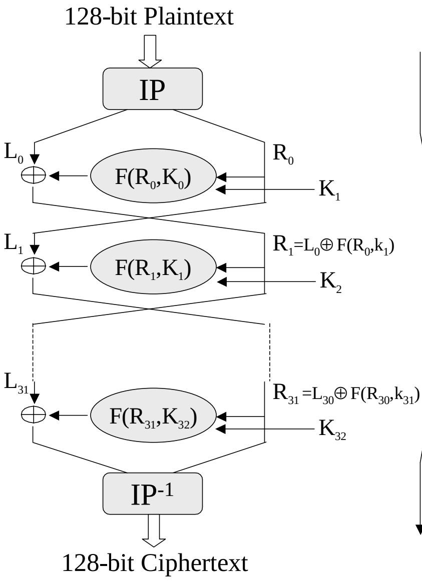

Fig. 1. 128-bit extended DES.

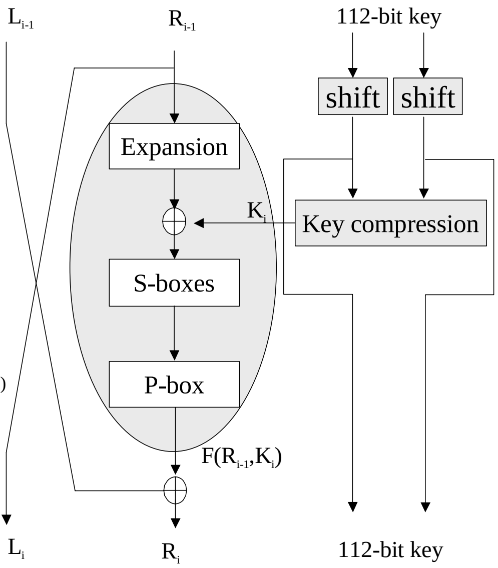

Fig. 2. One round of 128-extended DES.

The real key size of the extended DES is 112 bits. The key-scheduling algorithm generates a different 96-bit sub-key in each round. The key-scheduling algorithm of the extended DES is the same as the one of DES except that the key initial permutation, key compression permutation and shift permutation are replaced by those shown in Table 3, Table 4 and Table 5 respectively.

Table 3. Key initial permutation.

121	113	105	97	25	17	9
57	49	41	33	89	81	73
1	122	114	106	98	26	18
65	58	50	42	34	90	82
10	2	123	115	107	99	27
74	66	59	51	43	35	91
19	11	3	124	116	108	100
83	75	67	60	52	44	36
127	119	111	103	31	23	15
63	55	47	39	95	87	79
7	62	118	110	102	30	22
71	126	54	46	38	94	86
14	6	61	117	109	101	29
78	70	125	53	45	37	93
21	13	5	28	20	12	4
85	77	69	92	84	76	68

Table 4. Key compression permutation.

31	17	2	24	42	10
43	50	34	14	1	26
44	28	13	53	20	6
45	12	37	5	52	23
36	19	27	11	3	48
29	38	47	15	7	54
22	8	33	56	49	40
51	25	18	39	32	9
76	91	62	107	83	69
82	89	68	100	60	75
80	57	73	65	94	105
90	108	101	59	74	66
92	77	85	99	106	61
104	81	112	70	88	97
63	96	103	110	71	87
111	86	102	95	78	64

Table 5. Shift permutation.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Rotations	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

3.2 Permutations in the Extended DES

The following tables list the permutations used in the extended DES. These tables should be read from left to right and from top to bottom. For example, Table 6 shows the initial permutation (the IP block shown in Fig. 1). After the initial permutation, the 114th data bit is permuted to the first bit, and the 82th data bit is permuted to the third bit.

Table 6. Initial permutation(IP).

114	98	82	66	50	34	18	2
116	100	84	68	52	36	20	4
118	102	86	70	54	38	22	6
120	104	88	72	56	40	24	8
122	106	90	74	58	42	26	10
124	108	92	76	60	44	28	12
126	110	94	78	62	46	30	14
128	112	96	80	64	48	32	16
113	97	81	65	49	33	17	1
115	99	83	67	51	35	19	3
117	101	85	69	53	37	21	5
119	103	87	71	55	39	23	7
121	105	89	73	57	41	25	9
123	107	91	75	59	43	27	11
125	109	93	77	61	45	29	13
127	111	95	79	63	47	31	15

Table 7. Expansion permutation.

64	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	33
32	33	34	35	36	37
36	37	38	39	40	41
40	41	42	43	44	45
44	45	46	47	48	49
48	49	50	51	52	53
52	53	54	55	56	57
56	57	58	59	60	61
60	61	62	63	64	1

4. DIFFERENTIAL CRYPTANALYSIS OF THE EXTENDED DES

The reader is expected to be familiar with differential cryptanalysis and the corresponding full 16-round attack on DES [7]. Any pair of encryptions forms a characteristic. A characteristic is the xor value of the two plaintexts, the xor value of the two ciphertexts, and the xor of the two input and output values of F-function in every round.

The 2-round iterative characteristic interchanges the left and right parts after 2 rounds of DES-like cryptosystems. The characteristic works as shown in Fig. 3.

For a pair of plaintexts to be a 2-round characteristic, two outputs of the F-function must be the same. In addition, the two inputs of the F-function must differ in at least 3 neighboring S-boxes [8] which are called active S-boxes. Moreover, a smaller number of active S-boxes results a higher probability that a 2-round characteristic will exist. The probability of a 2-round characteristic depends only on the zero column of the S-boxes.

Table 8. Reverse initial permutation(IP^{-1}).

72	8	80	16	88	24	96	32
104	40	112	48	120	56	128	64
71	7	79	15	87	23	95	31
103	39	111	47	119	55	127	63
70	6	78	14	86	22	94	30
102	38	110	46	118	54	126	62
69	5	77	13	85	21	93	29
101	37	109	45	117	53	125	61
68	4	76	12	84	20	92	28
100	36	108	44	116	52	124	60
67	3	75	11	83	19	91	27
99	35	107	43	115	51	123	59
66	2	74	10	82	18	90	26
98	34	106	42	114	50	122	58
65	1	73	9	81	17	89	25
97	33	105	41	113	49	121	57

Table 9. P-box permutation.

16	39	52	21
61	44	60	49
33	15	55	58
5	18	31	42
2	8	24	14
32	27	35	9
19	45	62	38
54	11	4	25
48	7	20	53
29	12	28	17
1	47	23	26
37	50	63	10
34	40	56	46
64	59	3	41
51	13	30	6
22	43	36	57

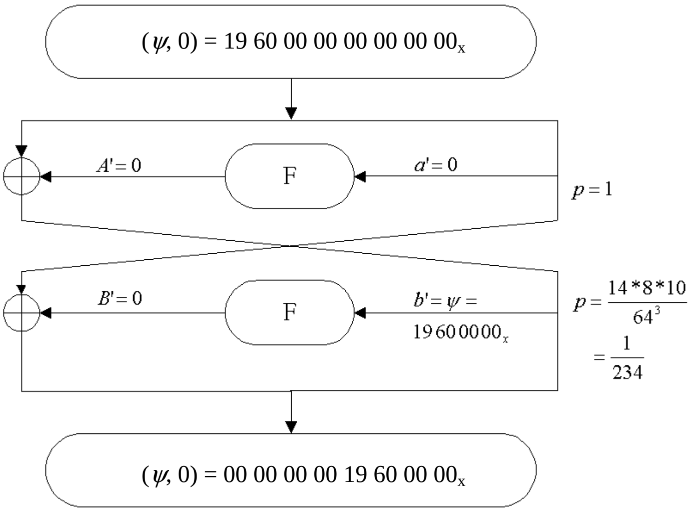
(ψ, 0) = 19 60 00 00 00 00 00 00x
A' = 0
F
a' = 0
p = 1
B' = 0
F
b' = ψ = 19600000x
p = 14*8*10 / 64^3 = 1/234
(ψ, 0) = 00 00 00 00 19 60 00 00x

Fig. 3. 2-round characteristic of DES.

In this work, every new S-box is a semi-similar type of original S-boxes. That is, $\mathrm{DDT}_{\text{S-boxi+8}}$[x, 0] = $\mathrm{DDT}_{\text{S-boxi}}$[x, 0] for i = 1, …, 8 x = 0, …, 64. Therefore, the best probability that a 2-round iterative characteristic will exist in the extended DES is exactly the same as in DES.

The 2-round iterative characteristics were used by Biham and Shamir [7] as the best characteristics in a differential attack designed to break DES. Some researchers have concluded that this is, in fact, the case [8]. The characteristic (ψ, 0) = 19 60 00 00 00 00 00 00_x used in [7] can be used in differential cryptanalysis of the extended DES with some modification.

The differential crytanalysis of the full 32-round extended DES is base on Biham and Shamir's work [7]. Fig. 4 shows how the extended DES could be analyzed. The 2-round iterative characteristic "19 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00_x" is used in rounds 2 to 30, and the overall probability is about $\left(\frac{1}{234}\right)^{14} = 2^{-110.2}$. Note that the probability of the 2-round characteristic is the same as the probability of the characteristic used in [7]. We will show that the extended DES using 16 S-boxes can also be broken by Biham's method of differential attack toward DES using 8 S-box. That is, the 32 rounds of the extended DES give more security then 16 rounds of DES. The additional round will not reduce the probability of the characteristic by the following method.

The output xor υ of the first round shown in Fig. 4 has only 12 non-zero bits, and there are in total 2^{12} such values. Biham constructed 2^{13} plaintexts by choosing an arbitrary plaintext P and defined

$$P_i = P \oplus (\upsilon_i, 0), \overline{P}_i = (P \oplus (\upsilon_i, 0)) \oplus (0, \psi) \quad \mathit{for}\ 0 \le i < 2^{12}.$$

For any i, there must be a j that makes the first round output xor of the plaintext pair $(P_i, \overline{P}_j)$ equal to $(\upsilon_i \oplus \upsilon_j)$, which is a possible value of υ. Therefore, there are 2^{12} such plaintexts out of the 2^{24} pairs. If we can examine these 2^{24} pairs in the order of 2^{12}, then the probability of the first round characteristic can be ignored. If the pair $(P_i, \overline{P}_j)$ is the desired pair, then there are 4*13 zero-bits in the round 32 input xor shown in Fig. 4. All 2^{12} plaintexts $\overline{P}_j$ are hashed by these zero-bits. For a plaintext P_i, a constant time hash table lookup is used to examine if any $\overline{P}_j$ forms the desired zero xor value in these bits.

The counting algorithm used by Biham and Shamir does not employ a counter. In this work, the counting algorithm is similar to Biham's work, but more then one key value will be analyzed for each pair. If the output xor and the non-zero input xor of an S-box are known and if the input value is also known, the number of possible key used by the S-box can be reduced from 64 to 4 [7]. In Fig. 4, the input and output xor pairs of S-box1, 2 and 3 in round 1 are known, which give 4 possible keys of each active S-box [7]. In addition, the input and output xor pairs of the last round are also known, which also give 4 possible keys of each S-box in the last round except S-box 1 and S-box 2 which have zero input xor values. Some key bits of S-box1, 2 and 3 in round 1 are used by S-box1, S-box2, …, S-box8 in round 32. The counter algorithm makes use of the common key bits in the first and last round to further reduce the number of possible keys.

For two S-boxes, S1 and S2, the numbers of possible keys are m and n, respectively, and K_{1i}, K_{2j} are the possible key values of S1, S2, respectively, where i = 1, …, m and j = 1, …, n. Suppose they share k common key bits. The numbers of possible keys for the S-box can be reduced if the possible keys have different common key bit values in S1 and S2. The probability that K_{1i} and K_{2j} have different common key bit values is $1 - {}^{1}\!/_{2^k}$. For any i, j. The expected number of possible keys for S1 that has the same common key bit value as S2 is:

$$m \bullet (1 - (1 - {}^{1}\!/_{2^k})^n). \tag{4.1}$$

Table 10 lists the common key bits of S-box1, S-box 2 and S-box 3 in round 1 and S-box1, S-box2, …, S-box8 in round 32. Using Eq. 4.1, the number of possible key values of S-box 1, …, S-box 8 in round 32 can be calculated, and the results are shown in Table 11.

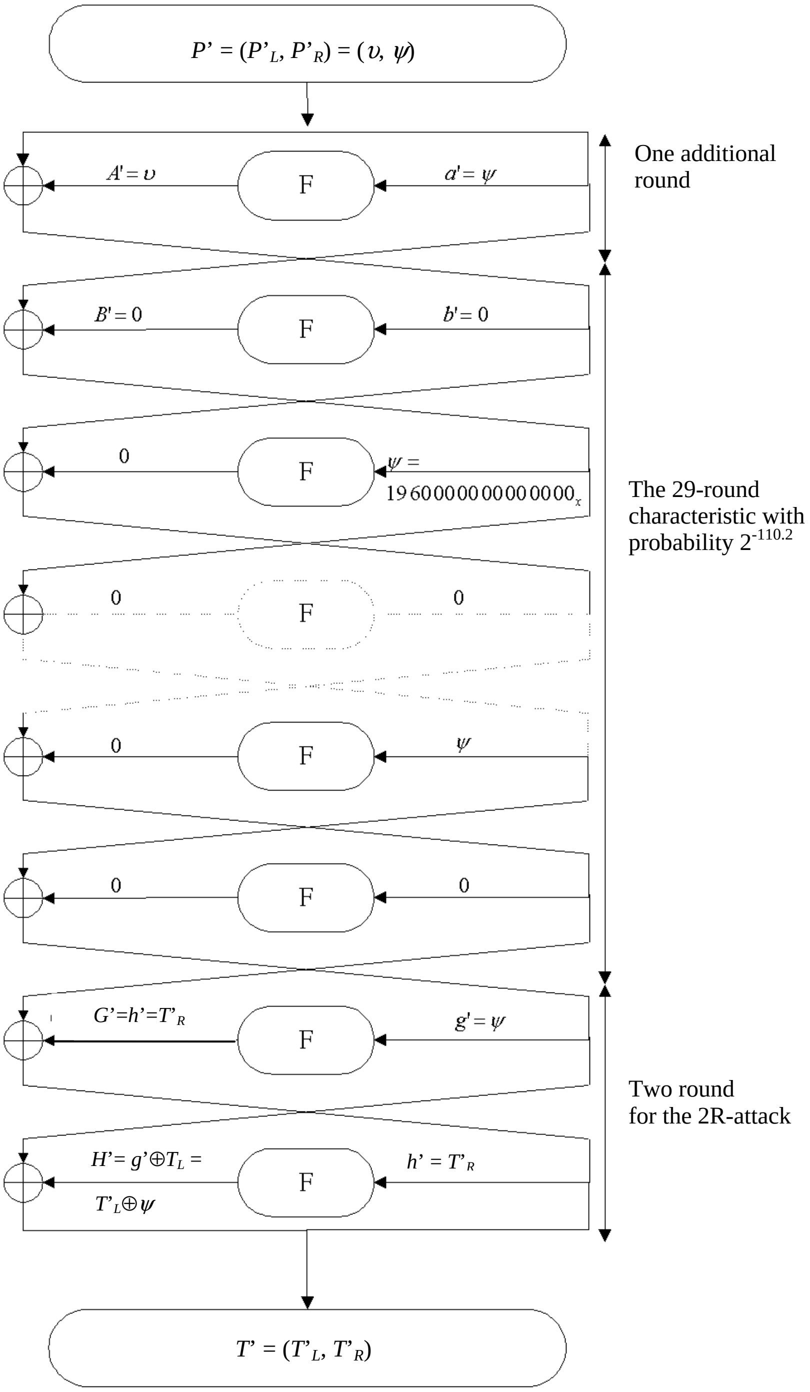

Fig. 4. Differential attack on the full 32 round extended DES.

Table 10. Number of common key bits.

	S-box 1	S-box 2	S-box 3
S-box 1	0	1	0
S-box 2	1	0	1
S-box 3	0	1	0
S-box 4	0	0	1
S-box 5	1	1	0
S-box 6	0	1	3
S-box 7	0	0	0
S-box 8	3	1	0

Table 11. Number of possible keys after reduction.

	Possible key #	Reduced key #
S-box 1	64	49.7
S-box 2	64	26.7
S-box 3	4	3.7
S-box 4	4	3.7
S-box 5	4	3.5
S-box 6	4	1.4
S-box 7	4	4
S-box 8	4	1.4

By analyzing the common key bits in round 1 and 32, the number of possible keys for S-box 1, …, S-box 8 in round 32 can be reduced to about 500000. The number of possible keys for S-box9, …,16 in round 32 is 4^8.

The signal to noise ratio is the ratio of the number of right pairs to the average count of incorrect subkeys. When the ratio is high enough, only a few right pairs are needed to identify the subkey. Biham observed that 4 right pairs are needed for a higher ratio and about 40 right pairs are needed for a ratio value of 1-2 to break DES. In Fig. 4, only the three active S-boxes in round 31 have non-zero xor value of output bits. There are (16 – 3) * 4 = 52 output bits which have zero xor values. Thus, only $m*2^{-52}$ pairs are analyzed among m chosen plaintexts. The counting algorithm analyzes 96 bits of key (the sub-key in the last round) and gives about 2^{35} possible keys for each analyzed plaintext pair. The S/N ratio is

$$S/N = \frac{m \bullet 2^{-110}}{m \bullet 2^{-52} \bullet 2^{35} / 2^{96}} = 2^3.$$

On average, 4 right pairs out of 2^{112} chosen plaintext pairs are needed. 2^{96} counters are used in the counting algorithm. Using the quartet structure [7] and another characteristic “1B 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00_x,” the number of plaintexts needed can be reduced by half. Therefore, about 2^{112} chosen plaintexts are needed to find enough right pairs for the 2-R attack in rounds 31 and 32. As in Biham’s work [7], the initial permutation is skipped.

5. LINEAR CRYPTANALYSIS OF THE EXTENDED DES

The reader is expected to be familiar with linear cryptanalysis toward DES [9]. Linear cryptanalysis [9] was first introduced by Matsui. This cryptanalysis makes use of the probabilistic xor parity relation between certain bits of the plaintext, ciphertext, and the key. The xor value of some carefully selected bits from plaintext and ciphertext and the xor value of some bits from key are the same with probability not equal to 0.5. That is, every plaintext and ciphertext pair carries some information about the key.

A LADT map is the linear approximation distribution table [9] of a function. It is a distribution of input bits xor and output bits xor. LADT[a, b] = $|\{x \in X | \oplus(a \wedge x) = \oplus (f(x) \wedge b)\}|$ where a, b ∈ X. Each entry of LADT is a linear approximation for f. For S-box to resist against any linear attack, its LADT entry should be balance that is close to 32. We define absolute LADT to be ALADT[a, b] = |LADT[a, b] – 32|. To break DES by linear cryptanalysis, Matusi found high probability linear approximation from the LADTs of S-boxes. Matusi's linear approximation uses a special type of entries LADT[a, b] where h(a) = 1.

Every entry in the LADTs of the S-boxes is a linear approximation for the S-box. If an entry of an S-box's LADT is used in the linear approximation, the S-box is said to be active. By limiting the max number of active S-boxes in every round to 1, Matsui found a high probability linear approximation for 16 round DES. Biham had shown Matsui's approximation to be the best [12].

By adding two additional rounds in the first and last round, Matusi attacked 16 round DES using a 14 round linear approximation in Eq. 5.1.

$$R_{1}[8, 14, 25] \oplus L_{15}[17] \oplus R_{15}[3, 8, 14, 25] = J_{3} \oplus J_{7} \oplus J_{10} \oplus K_{14}[26] \qquad (5.1)$$

Where $J_i = K_i[26] \oplus K_{i+1}[4] \oplus K_{i+2}[26]$, for any i, j $K_i[j]$ is the value of the j^{th} bit of the sub-key in round i. $R_1[8, 14, 25]$ is the xor value of the 8^{th},14^{th},25^{th} bit of right data part in round 1. And $R_{15}[3, 8, 14, 25]$ is the xor value of the 3^{th}, 8^{th}, 14^{th}, 25^{th} bit of left data part in round 15.

Expression 5.2 is the detail 14 round linear approximation. A, B, C are one round linear approximation with only 1 active S-box. A "–" means that no approximation is needed in that round.

$$– ABC – CBA – ABC – C \qquad (5.2)$$

Table 12. The notations used in expression 5.2.

	Linear approximation expression	LADT entry of active S-box	Probability
A	R[17] ⊕ F[8, 14, 25] = K[26]	$LADT_{S\text{-}box5}[16, 14] = 42$	42/64
B	R[3] ⊕ F[17] = K[4]	$LADT_{S\text{-}box1}[4, 4] = 30$	30/64
C	R[17] ⊕ F[3, 8, 14, 25] = K[26]	$LADT_{S\text{-}box5}[16, 15] = 12$	12/64

The 3-round linear approximation "ABC" is an iterative structure in expression 5.2. Fig. 5 shows how this type of structure is possible. The active S-box in linear approximation A and C is S-box5. The active S-box in linear approximation B is S-box1.

Fig. 6 shows how a 3-round linear approximation is possible. B1 and B2 are the common input and output bits of S1 and S2. O1 and O2 are set of output bits for S1 and S2 respectively where O1 = O2⊕B2. A linear approximation used in each round approximates the relation of one single input bit value and the xor value of certain output bits of an active S-box. The probability of this type of linear approximation is minimized in the extended DES.

The extended DES has 16 S-boxes in the F-function. More S-boxes make the design of p-box more flexible. The p-box used in the extended DES is carefully find-tuned to reduce the number of 3-round iterative linear approximations and to minimize the prob-

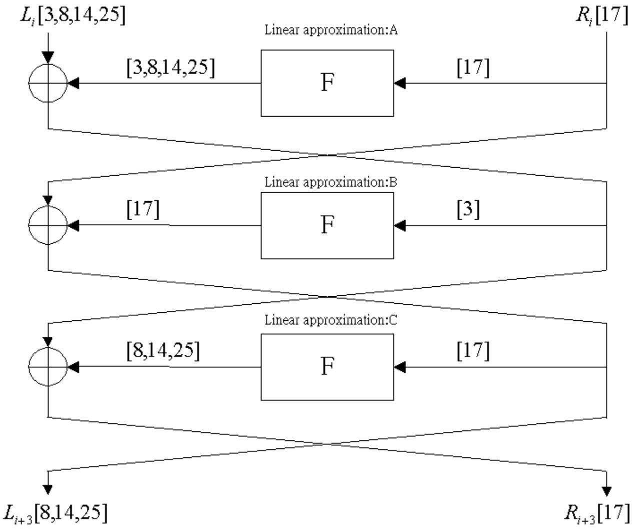

Fig. 5. 3-round iterative linear approximation.

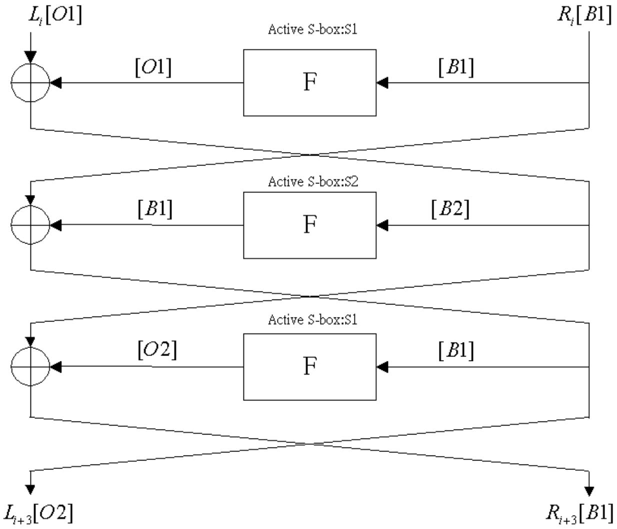

Fig. 6. A general 3-round iterative linear approximation.

ability of all existing 3-round linear approximations. The best 3-round linear approximation in the extended DES, "EFG", can be used to construct the following 30 round linear approximation.

$$- \text{EFG} - \text{GFE} - \text{EFG} - \text{GFE} - \text{EFG} - \text{GFE} - \text{EFG} - \text{G} \tag{5.3}$$

Table 13. The notations used in expression 5.3.

	Linear approximation expression	LADT entry of active S-box	Probability
E	R[7] ⊕ F[34, 18] = K[10]	$\text{LADT}_{\text{S-box2}}[4, 3] = 36$	36/64
F	R[60] ⊕ F[7] = K[89]	$\text{LADT}_{\text{S-box15}}[2, 1] = 34$	34/64
G	R[7] ⊕ F[60, 34, 18] = K[10]	$\text{LADT}_{\text{S-box2}}[4, 7] = 38$	38/64

Applying Table 13 to expression 5.3 and Eq. 5.4 can be used in linear cryptanalysis for the full 32-round extended DES.

$$R_{1}[34, 18]\oplus L_{31}[7]\oplus R_{31}[60, 34, 18] = J_{3}\oplus J_{7}\oplus J_{10}\oplus J_{14}\oplus J_{18}\oplus J_{22}\oplus J_{26}\oplus K_{31}[10] \quad (5.4)$$

Where $J_i = K_i[10]\oplus K_{i+1}[89]\oplus K_{i+2}[10]$, for any i, j $K_i[j]$ is the value of the j^{th} bit of the sub-key in round i. $R_1[34, 18]$ is the xor value of the 34^{th}, 18^{th} bit of right data part in round 1. And $R_{31}[18, 34, 60]$ is the xor value of the 60^{th}, 34^{th}, 18^{th} bit of left data part in round 15.

By Matsui's piling-up Lemma, we can calculate the probability of Eq. 5.4.

$p = \frac{1}{2} + 2^{n-1}\prod_{i=1}^{n}(p_i - \frac{1}{2})$ where $\boldsymbol{P}$ is the probability of the Eq. 5.4 to hold. $\boldsymbol{P}_i$ is the probability of the i^{th} round linear approximation to hold. Using the probability value in Table 12 to evaluate Eq. 5.3, the probability $\boldsymbol{P}$ is $0.5 + 2^{69.3}$. By Matsui's rule of thumb, the linear cryptanalysis requires about $8/(\boldsymbol{P} - \frac{1}{2})^2$ known plaintexts. We have the complexity of linear cryptanalysis for the 32-round extended DES to be 2^{142} known plaintexts.

6. CONCLUSIONS

The original S-boxes and the algorithms of DES are designed to resist differential attack [11]. We have proposed new S-boxes having properties that are well-known or defined in this work. We have shown the similarity between new proposed S-boxes and original DES S-box and the approach to generate them. In addition, the algorithm of the extended DES is similar to that of DES. The extended DES encrypts a 128-bit data block using a key size of only 112 bits. The complexities of differential and linear attack are 2^{112} and 2^{142} respectively.

The original DES algorithm has some weakness in terms of memory requirement for storing S-boxes. Since our extension also uses S-boxes in our algorithm, it has the same weakness as DES from this aspect. Nevertheless, the similarity between our algorithm and DES makes it easy to provide a securer and larger block size cipher in the system which uses DES as the encryption solution.

Some researchers have proposed better S-boxes which they claim are better [6, 10]. These S-boxes can be used to further strengthen the extended DES. The number of S-boxes can be doubled again to obtain a larger key size and larger data size in further expansion.

REFERENCES

1. J. Seberry and X. M. Zhang, "Highly nonlinear 0-1 balanced Boolean functions satisfying strict avalanche criterion," *Advances in Cryptology-AUSCRYPT '92 Proceedings*, 1993, pp. 145-155.
2. A. F. Webster and S. E. Tavares, "On the design of S-boxes," *Advances in Cryptology-CRYPO'85 Proceedings*, 1986, pp. 523-534.
3. J.-H. Evertse, "Linear structure in block ciphers," in *Proceedings of Eurocrypt' 87*, pp. 249-266.
4. W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions," in *Proceedings of Eurocrypt' 89*, pp. 549-562.
5. X. M. Zhang, Y. Zheng, and H. Imai, "Relating differential distribution tables to other properties of substitution boxes," *Designs Codes and Cryptography*, Vol. 19, 2000, pp. 45-63
6. J. H. Cheon, S. Chee, and C. Park, "S-boxes with controllable nonlinearity," in *Proceedings of EUROCRYPT'99*, pp. 286-294.
7. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, New York:Springer-verlag, 1993.
8. L. R. Knudsen, "Iterative characteristics of DES and S^2-DES," *Advances in Cryptology-CRYPO'91, Proceedings*, Berlin: Springer-Verlag, pp. 497-511.
9. M. Matsui "Linear cryptoanalysis method for DES cipher," in *Proceedings of EUROCRYPT'93*, pp. 386-397.
10. W. Millan, L. Burnett, G. Carter, A. Clark, and E. Dawson, "Evolutionary heuristics for finding cryptographically strong S-boxes," *Information and Communication Security, Second International Conference*, 1999 pp. 263-274.
11. D. Coppersmith, "The data encryption standard (DES) and its strength against attacks," *IBM Journal of Research and Development*, Vol. 38, 1994, pp. 243-250.
12. E. Biham "On Matsui's linear cryptanalysis," *Advances in Cryptology—EUROCRYPT'94:/Workshop on the Theory and Application of Cryptographic Techniques*, pp. 341-355.

Yi-Shiung Yeh (葉義雄) received the MS and Ph.D. degrees in Computer Science, the Department of EE & CS from University of Wisconsin-Milwaukee in 1980 and 1985, respectively. Dr. Yeh is currently an associate professor of the Institute of Computer Science and Information Engineering at National Chiao Tung University. During July 1986 to August 1988, he was an associate professor in the Department of Computer and Information Science at Fordham University. During July 1984 to December 1984, he worked as a doctorate intern at Johnson Controls, Inc. During August 1980 to October 1981, he was a system programmer in the system support division of the Milwaukee County Government. His research interests include cryptography and information security, reliability and performance, DNA computation.

Ching-Hung Hsu (許景竑) is currently working towards the Ph.D. degree in Computer Science, Department of Computer & Information Science at National Chiao Tung University. He received the BS and MS degrees in Computer Science, Department of Computer & Information Science, National Chiao Tung University in 1994 and 1996, respectively. His research interests include data security and artificial intelligent.