First Scientific Conference of the State Company of Internet Services, Baghdad, Iraq. October 22nd - 23rd , 2002.

DES80 – A DES Variant Cryptographic System

Dr. Abdul-Karim A-R. Kadhim
Computer Engineering Dep.
Al-Rasheed Engineering College
Baghdad, IRAQ

Mohammed M. Alani
Computer Engineering Dep.
College of Engineering, Nahrain University
Baghdad, IRAQ

Abstract

The Data Encryption Standard (DES) has shown noticeable weaknesses during the last decade. In this paper we develop a system that is a DES-variant but has more resistance towards the latest attacks against DES. The developed system has a sub-key generation algorithm that is totally different from the original DES one.

The developed system uses a 70-bit initial key instead of the 56-bit key originally used. It has substitution boxes inside the key generation algorithm and mod2 addition. The choice of arrangement of substitution boxes in the main algorithm for each round is sub-key dependent.

The result of our design is a DES-variant cryptographic system that has higher resistance towards brute-force attack, differential cryptanalysis, and linear cryptanalysis. Our design also cancelled the weak-keys and complement-keys properties of the DES.

Design Concepts

Our design criterion was concentrated to overcome the DES weaknesses with the least possible losses in time and memory.

The weak-keys, semi-weak keys, and possibly-weak keys were one thing to cancel because of their effect on generating the sub-keys. Also the complement-keys property was to be cancelled by our design.

The key length problem was one of the major ones. With the increasing computing abilities now days, the brute-force attack to a 56-bit seems feasible.

The most important attacks to fight were the differential and linear cryptanalysis. These two attacks depend heavily on the design of the S-Boxes of the DES.

The Algorithm

The proposed key generation algorithm has an 80-bit key length from which only 70 bits are used after removing the parity bits, 7-bit left shift each round, a part to indicate the arrangement of the S-Boxes of each round, a stage of S-Boxes inside the key algorithm itself, and more linear permutations to provide more diffusion.

The proposed key generation algorithm is described in the following steps:

Step 1: The 80-bit key enters an initial permutation that discards the 10 parity bits to give out a 70-bit key. These permutations are shown in Figure 1.

Step 2: The 70 bits are now divided into three parts:

A. 48 bits: Enter the S-Boxes to produce a 32-bit output.

B. 16 bits: Enter a secondary permutation box to produce a permuted 16-bit output. These secondary permutations are shown in Figure 2.

C. 6 bits: The first two are XORed and the last two are XORed also, and the other two are left with no operation to produce 4 bits.

Step 3: The leftmost 16 bits of the 32-bit output of (Step 2,A) are swapped with the 16-bit output of (Step 2,B) and all outputs of (Step 2) are combined to produce a 48-bit block to be sent to the main algorithm as K1.

Step 4: The 4-bit output from (Step 2,C) is used twice after adding 1 to the two least significant bits and discarding the carry. First, the 4 bits are sent to the main algorithm to control the arrangement of the S-Boxes. The first bit determines whether to swap the boxes 2 and 3, the second bit is used to control the swapping of boxes 1 and 7, the third controls boxes 4 and 6, and the fourth controls the boxes 5 and 8.

Then, the 4 bits are recombined with the 48 bits to prepare the sub-key of the next round.

Step 5: For the next round, a shift of 7 bits to the left takes place and the next 48 bits are algorithm and the left four bits are dealt with as the output of (Step 2,C), and so on for 16 rounds.

Figure 3 shows the complete sub-key generation algorithm.

The only change to the main algorithm was the 4 bits sent with each sub-key to determine the arrangement of the S-Boxes for each round.

57	43	47	25	62	2	61	66	45	4
34	70	30	77	22	5	76	20	42	55
75	38	41	50	52	79	23	67	51	21
73	19	53	17	9	44	13	74	10	27
29	1	31	35	14	65	33	11	28	6
3	58	12	78	7	37	18	15	63	26
54	39	71	36	59	60	68	49	69	46

Figure 1 -- The Proposed System Main Permutations

12	9	10	5	1	13	16	11
14	4	6	3	8	2	7	15

Figure 2 -- The Proposed System Secondary Permutations

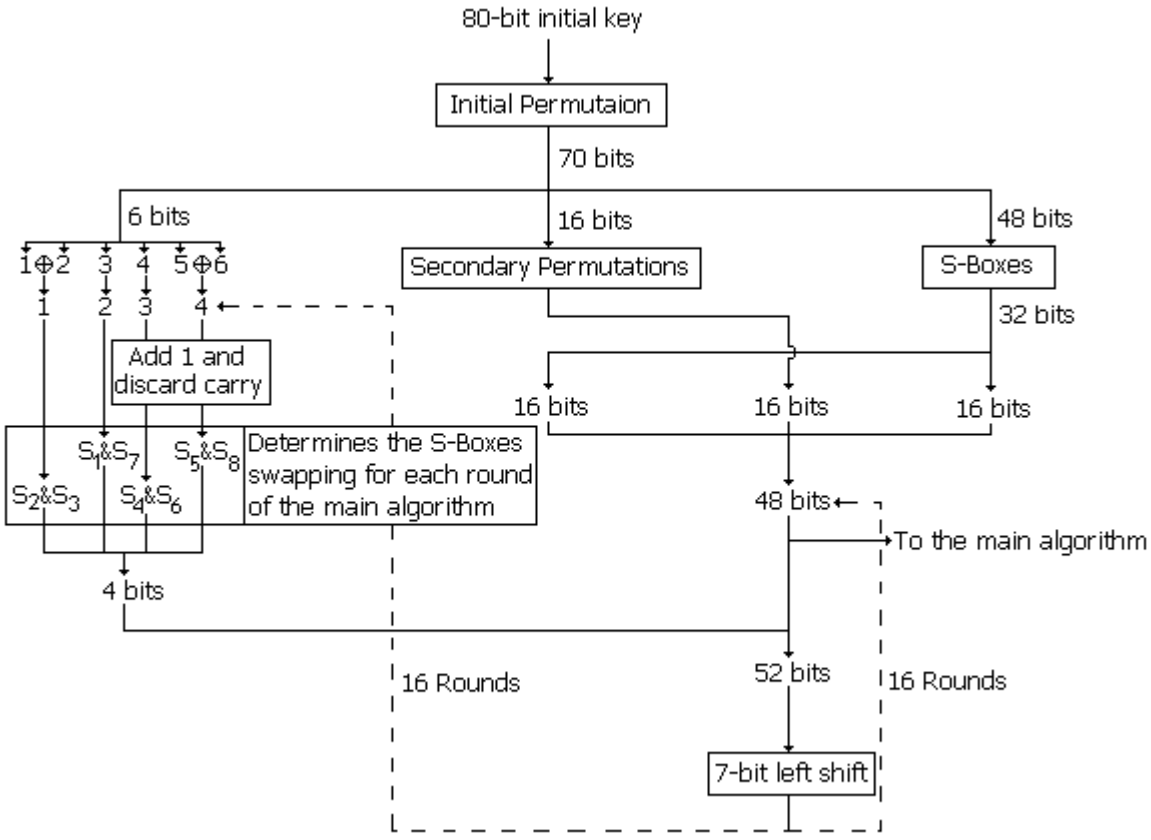

Figure 3 – The DES80 Sub-Key Generation Algorithm

System Description

The DES80 has the following advantages over the DES:

1. The 70-bit key instead of the original 56-bit key is aimed to resist brute-force attack. This gives $2^{70}=1.18*10^{21}$ instead of $2^{56}=7.2*10^{16}$. The key was not increased more than that to follow the criterion that a good system must have minimum possible key.
2. The S-Boxes inside the key generation algorithm are aimed to reduce linearity. This is to resist linear cryptanalysis providing a non-linear operation. The non-linear operation was chosen to be the same S-Boxes of the main algorithm in order to reduce the memory requirements (for software implementation) and the components needed (for hardware implementation). And it is done only once to reduce the time required for sub-key generation, and it is convenient for key generation [3].
3. The permutations inside the sub-key generation algorithm are to provide more diffusion [5][2].
4. Controlling the arrangement of the S-Boxes of the main algorithm of each round by the sub-keys is aimed to resist both linear and differential cryptanalysis. Because these two attacks depend heavily on the structure of the S-Boxes [2] [3] [4].

5. The addition of 1 and neglecting the carry provides two benefits; first, the cancellation of weak, semi-weak, possibly-weak keys. Second is the cancellation of the complement-keys property.
6. The 7-bit left shift will provide, at least, 15 different sub-keys for each key. Any number less than 7 will provide more similar sub-keys. It is clear that the number of shifts should not be a factor of 52, because it will cause alike patterns.

System Testing

Three tests were done on the system; a test for randomness, and two tests for avalanche effect [1].

1. Randomness test:

This test was to calculate the number of zeros and ones in the ciphertext resulting from the DES80 encryption and the result were compared with the plaintext. The test was implemented on text files, audio files, video files document files, and program files as these are the most common files to be encrypted. The results were as shown in table 1.

File type	File size (bits)	No. of Zeros in plaintext	%	No. of Ones in plaintext	%	No. of Zeros in ciphertext	%	No. of Ones in ciphertext	%
Text	108288	59599	55.03	48689	44.97	53967	*49.83*	54321	*50.17*
Audio	99904	58891	58.94	41013	41.06	49857	*49.9*	50047	*50.1*
Video	128256	70163	54.7	58093	45.3	64186	*50.04*	64070	*49.96*
Document	139968	78149	55.83	61819	44.17	70054	*50.05*	69914	*49.95*
Program	1191936	644666	54.08	547270	45.92	578592	*48.54*	613344	*51.46*
Text	1332136	748934	56.22	583202	43.78	666037	*49.99*	666099	*50.01*
Audio	1000224	521409	52.12	478815	47.88	500189	*50*	500035	*50*
Video	988920	531032	53.69	457888	46.31	494396	*49.99*	494524	*50.01*
Document	1101880	480259	43.58	621621	56.42	550950	*50*	550930	*50*
Program	1090328	499503	45.81	590825	54.19	545110	*49.99*	545218	*50.01*

Table 1 -- The results of randomness test

2. Changing plaintext bits avalanche test:

One, two and three bits were changed in the plaintext and the change in the output ciphertext was measured by hamming distance measurement. Table 2 shows the results of this test.

3. Changing key bits avalanche test:

Also, one, two, and three bits were changed in the initial key bits and the change in the output ciphertext was measured by hamming distance measurement. Table 3 shows the results of this test.

Input No.	Hamming distance for changing 1 bits	Hamming distance for changing 2 bits	Hamming distance for changing 3 bits
1	*31*	*42*	*52*
2	*35*	*40*	*43*
3	*32*	*38*	*40*
4	*36*	*39*	*42*
5	*34*	*31*	*41*
6	*37*	*34*	*39*
7	*41*	*34*	*38*
8	*40*	*32*	*50*
9	*37*	*38*	*55*
10	*32*	*37*	*46*
11	*29*	*53*	*42*
12	*33*	*50*	*37*
13	*35*	*41*	*40*
14	*37*	*43*	*43*
15	*33*	*44*	*44*
16	*34*	*37*	*44*
17	*30*	*32*	*37*
18	*29*	*33*	*46*
19	*28*	*41*	*41*
20	*29*	*47*	*39*
21	*34*	*40*	*39*
22	*31*	*33*	*61*
23	*36*	*37*	*42*
24	*41*	*43*	*39*
25	*36*	*41*	*37*
Average	**34**	**39.2**	**43.08**

Table 2 – The results of changing plaintext bits

Input No.	Hamming distance for changing 1 bits	Hamming distance for changing 2 bits	Hamming distance for changing 3 bits
1	*30*	*40*	*43*
2	*34*	*41*	*44*
3	*35*	*37*	*38*
4	*38*	*35*	*44*
5	*31*	*33*	*43*
6	*36*	*35*	*37*
7	*39*	*37*	*38*
8	*43*	*31*	*48*
9	*37*	*39*	*53*
10	*30*	*36*	*41*
11	*29*	*48*	*40*
12	*31*	*31*	*39*
13	*34*	*43*	*43*
14	*38*	*41*	*41*
15	*35*	*40*	*39*
16	*33*	*40*	*43*
17	*33*	*33*	*39*
18	*28*	*31*	*37*
19	*29*	*39*	*46*
20	*30*	*45*	*37*
21	*33*	*41*	*42*
22	*30*	*35*	*51*
23	*33*	*33*	*52*
24	*40*	*41*	*39*
25	*38*	*43*	*37*
Average	**33.88**	**37.92**	**42.16**

Table 3 – The results of changing key bits.

Conclusion

As the DES shows many weaknesses, a DES-variant cryptographic system was designed to overcome most of these weaknesses.

The designed system, named DES80, is designed to resist brute-force attack, differential cryptanalysis [4], and linear cryptanalysis [3]. These attacks were in our design criterion because they are the most effective attacks against the DES yet.

DES80 has an 80-bit initial key, sub-key dependent S-Boxes for each round, S-Boxes inside the key generation algorithm, and other features that strengthen it against the known attacks.

The system was tested for randomness and avalanche effects. The results of these test has proven that DES80 is a reliable system. The system as compared to DES has better avalanche characteristics [1].

References:

[1] Lamia A. Muhammed, *Designing and Testing Cryptosystem*, Thesis, 2000.
[2] B. Schneier, *Applied Cryptography*, John Wiley & Sons, New York, 1996.
[3] Eli Biham, *On Matsui's Linear Cryptanalysis*, Technical Report CS0813, 1994.
[4] E. Biham and A. Shamir, *Differential Cryptanalysis of the Full 16-Round DES*, Technical Report CS0708, 1991.
[5] D.E. Denning, *Cryptography and PC Security*, McGraw-Hill Companies, London, 1997.