The Avalanche Effect

- A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext. In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext. This is referred to as the avalanche effect
- 1 bit change in the plain text ⇔ changes around 34 bits of cipher text
- Similarly 1bit change in Key ⇔ changes around 35 bits of cipher text

Avalanche Effect in DES: Change in Plaintext

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

Avalanche Effect in DES: Change in Key

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	27
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30

THE STRENGTH OF DES: Use of 56-Bit Keys

- With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.2×10^{16} keys.
- A brute-force attack appears impractical.
- single machine performing one DES encryption per microsecond would take more than a thousand years to break the cipher.
- 1977, Diffie and Hellman postulated that the technology existed to build a parallel machine with 1 million encryption devices, each of which could perform one encryption per microsecond [DIFF77]. This would bring the average search time down to about 10 hours. The authors estimated that the cost would be about $20 million in 1977 dollars.

THE STRENGTH OF DES: Use of 56-Bit Keys

- With current technology, it is not even necessary to use special, purpose-built hardware. Rather, the speed of commercial, off-the-shelf processors threaten the security of DES.
- A single PC can break DES in about a year; if multiple PCs work in parallel, the time is drastically shortened.
- Today's supercomputers should be able to find a key in about an hour.
- Key sizes of 128 bits or greater are effectively unbreakable using simply a brute-force approach.
- Even if we managed to speed up the attacking system by a factor of 1 trillion (10^{12}), it would still take over 100,000 years to break a code using a 128-bit key.

Average Time Required for Exhaustive Key Search

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^{9} years	6.3×10^{6} years

THE STRENGTH OF DES: Nature of the DES Algorithm

- The possibility of exploiting the characteristics of the DES algorithm.
- The focus of concern has been on the eight substitution tables, or S-boxes, that are used in each iteration.
- The design criteria for these boxes, and indeed for the entire algorithm, were not made public
- Speculation is that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes.
- Unexpected behaviors of the S-boxes have been discovered.
- No known cases of exploiting this case has been recorded till date.

THE STRENGTH OF DES: Timing Attacks

- A timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts.
- A timing attack exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs.
- DES appears to be fairly resistant to a successful timing attack

Differential Cryptanalysis

- Differential cryptanalysis is a method for breaking certain classes of cryptosystems
- It was invented in 1990 by Israeli researchers Eli Biham and Adi Shamir
- However, apparently the IBM researchers who designed DES knew about differential cryptanalysis, as was indicated by Don Coppersmith of TJ Watson Research Center

Differential Cryptanalysis

- Differential cryptanalysis is efficient when the cryptanalyst can choose plaintexts and obtain ciphertexts (chosen plaintext cryptanalysis)
- The known plaintext differential cryptanalysis is also possible, however, often the size of the known text pairs is very large
- The method searches for plaintext, ciphertext pairs whose difference is constant, and investigates the differential behavior of the cryptosystem
- The difference of two elements P1 and P2 is defined as P1 $\oplus$ P2 (bit-wise XOR operation) for DES
- The difference may be defined differently if the method is applied to some other cryptosystem

Differential Cryptanalysis

- Differential cryptanalysis is applicable to the iterated ciphers with a weak round function (so-called Feistel ciphers)
- The summary of the technique:
  - Observe the difference between the two ciphertexts as a function of the difference between the corresponding plaintexts
  - Find the highest probability differential input (called characteristic) which can be traced through several rounds
  - Assign probabilities to the keys and locate the most probable key

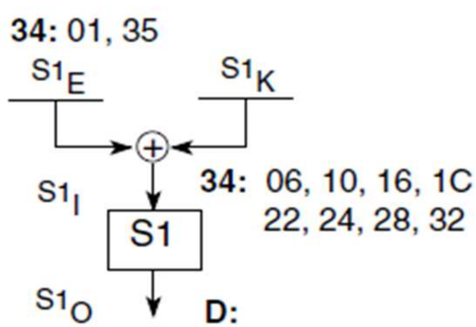
$S1$ Differential Distribution Table

Input x'	Output y'															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01	0	0	0	6	0	2	4	4	0	10	12	4	10	6	2	4
02	0	0	0	8	0	4	4	4	0	6	8	6	12	6	4	2
03	14	4	2	2	10	6	4	2	6	4	4	0	2	2	2	0
⋮																
0C	0	0	0	8	0	6	6	0	0	6	6	4	6	6	14	2
⋮																
34	0	8	16	6	2	0	0	12	6	0	0	0	0	8	0	6
⋮																
3E	4	8	2	2	2	4	4	14	4	2	0	2	0	8	4	4
3F	4	4	4	2	4	0	2	4	4	2	4	8	8	6	2	2

- The 6-bit differential input x’ takes 64 values: 00 (hex) to 3F (hex)
- The 4-bit differential output y’ takes 16 values: 0 (hex) to F (hex)
- The first row has zeros in all but the first column, because when $x' = x \oplus x* = 0$, the same input occurs twice.
- Therefore, the same output must also occur both times and $y' = y \oplus y* = 0$
- The later rows are more interesting:
- For example, when x’ = 01, five of the sixteen possible y’ values 0, 1, 2, 4, 8 occur with zero probability (i.e., never occurs)
- A occurs with probability 16/64. 9 and C occur with probability 10/64
- This is a highly non-uniform distribution
- This differential non-uniformity is observed in all of the S-boxes S1, S2, . . . , S8

Determination of the key:

Suppose we know two inputs to $S1$ as 01 and 35 which XORs to 34, and the output XOR as D



XOR is 34

6	32
10	24
16	22
1C	28

The input XOR is 34, regardless of the value of the key because

$$\begin{aligned} S1'_I &= S1_I \oplus S1^*_I \\ &= (S1_E \oplus S1_K) \oplus (S1^*_E \oplus S1_K) \\ &= S1_E \oplus S1^*_E \\ &= S1'_E \end{aligned}$$

Also since

$$S1_I = S1_E \oplus S1_K$$

we have

$$S1_K = S1_I \oplus S1_E$$

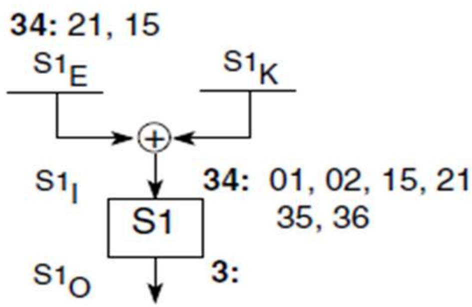
which gives

$$\begin{array}{rcl@{\qquad}rcl} 06 \oplus 01 &=& 07 & 06 \oplus 35 &=& 33 \\ 10 \oplus 01 &=& 11 & 10 \oplus 35 &=& 25 \\ 16 \oplus 01 &=& 17 & 16 \oplus 35 &=& 23 \\ 1C \oplus 01 &=& 1D & 1C \oplus 35 &=& 29 \\ 22 \oplus 01 &=& 23 & 22 \oplus 35 &=& 17 \\ 24 \oplus 01 &=& 25 & 24 \oplus 35 &=& 11 \\ 28 \oplus 01 &=& 29 & 28 \oplus 35 &=& 1D \\ 32 \oplus 01 &=& 33 & 32 \oplus 35 &=& 07 \end{array}$$

Thus, possible keys are:

$$\{07, 11, 17, 1D, 23, 25, 29, 33\}$$

Furthermore, suppose we know two inputs to $S1$ as 21 and 15 which XORs to 34, and the output XOR as 3



This gives the key values:

$$
\begin{array}{rclcrcl}
01 \oplus 21 & = & 20 & \quad & 01 \oplus 15 & = & 14 \\
02 \oplus 21 & = & 23 & & 02 \oplus 15 & = & 17 \\
15 \oplus 21 & = & 34 & & 15 \oplus 15 & = & 00 \\
21 \oplus 21 & = & 00 & & 21 \oplus 15 & = & 34 \\
35 \oplus 21 & = & 14 & & 35 \oplus 15 & = & 29 \\
36 \oplus 21 & = & 17 & & 36 \oplus 15 & = & 23
\end{array}
$$

as

$$\{00, 14, 17, 20, 23, 34\}$$

The correct key value must appear in both of these sets:

$$\{07, 11, 17, 1D, 23, 25, 29, 33\}$$

$$\{00, 14, 17, 20, 23, 34\}$$

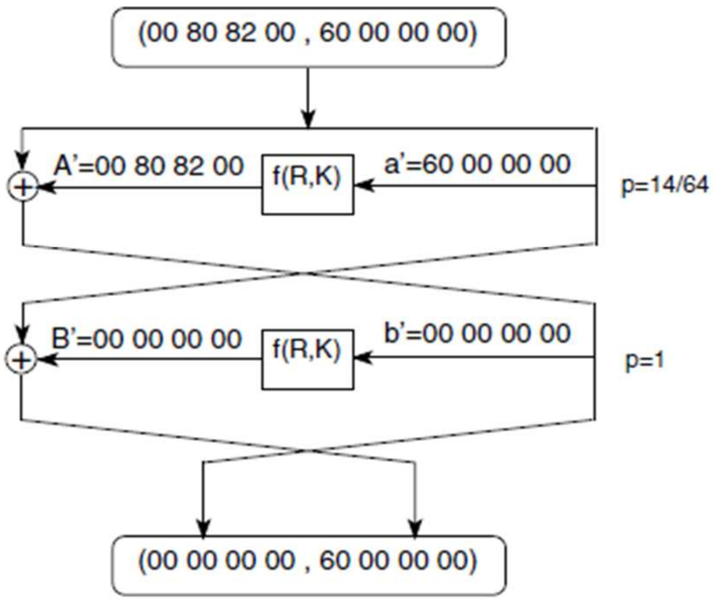
Intersecting these two sets, we obtain

$$\{17, 23\}$$

Thus, the key value is either 17 or 23

In order to determine which one of these is the correct value, we need more input/output XORs

A 2-Round Characteristic



The differential input to F in the first round is $a' = 60\ 00\ 00\ 00$

The expansion operation puts these half bytes into the middle four bits of each S-box in order $6 = 0110$ goes to $S1$ and $0 = 0000$ goes to $S2, \ldots, S8$

Since all the edge bits are zero, $S1$ is the only S-box receiving non-zero differential input

$S1$'s differential input is $0\ 0110\ 0 = 0C$ while the differential inputs of $S2, \ldots, S8$ are all zero

Looking in $S1$'s differential distribution table, we find that when $x' = 0C$, the highest probability differential output y' is $E = 1110$, which occurs with probability $14/64$

All the other S-boxes have $x' = 0$ and $y' = 0$ with probability 1

The S-box outputs go through the permutation P before becoming the output $f(R,K)$

As shown, the differential output of $f(R,K)$ is

$$A' = P(E0\ 00\ 00\ 00) = 00\ 80\ 82\ 00$$

$A' = 00\ 80\ 82\ 00$ is then XORed with $L' = 00\ 80\ 82\ 00$ to give 00 00 00 00

Thus, in the second round all S-boxes receive their differential inputs as zero, producing the differential outputs as zero

The ouput of $f(R,K)$ in the second round is zero, giving the differential output as depicted: (00 00 00 00 , 60 00 00 00)

Differential Cryptanalysis of 2-Round DES

This analysis assumes the initial (IP) and final (FP) permutations are removed from the DES algorithm

Step 1: Generate a plaintext pair (P, P^*) such that

$$P' = P \oplus P^* = 00\ 80\ 82\ 00\ 60\ 00\ 00\ 00$$

This is done by generating a random P and XORing it with

00 80 82 00 60 00 00 00

to generate P^*

Step 2: Give the plaintext pair (P, P^*) to your opponent who enciphers it and gives you the ciphertext pair (T, T^*)
(chosen plaintext cryptanalysis)

Step 3: Compute $T' = T \oplus T^*$ and see whether it is equal to

$$00\ 00\ 00\ 00\ 60\ 00\ 00\ 00$$

If it does not, the characteristic has not occurred and this pair is not used. Go to Step 1 and generate a new plaintext pair.

If T' is equal to

$$00\ 00\ 00\ 00\ 60\ 00\ 00\ 00$$

then the characteristic has occurred, and we know the values of A' and B'. Go to Step 4.

Step 4: Since $S2, \ldots, S8$ have their differential inputs equal to zero, no information can be gained about $S2_K, \ldots, S8_K$

Because, in the differential distribution table of $S1$, we have $0C \rightarrow E$ with probability 14/64, only 14 of 64 possible $S1_K$ values allow

$$a' = 60\ 00\ 00\ 00$$

to produce

$$A' = 00\ 80\ 82\ 00$$

These 14 allowable values can be determined by XORing each possible $S1_K$ with the corresponding six bits of $S1_E$ and $S1^*_E$, computing $S1$'s differential output $S1'_O$ and checking if it is equal to E

Put these 14 values of $S1_K$ in a table

Step 5: Compute the intersection of these tables

Since the correct key value must occur in each table, it will be in the intersection

If more than one $S1_K$ value results, we do not have enough plaintext, ciphertext differential pairs to uniquely determine $S1_K$. Go to Step 1 and generate additional data

The number of plaintext, ciphertext differential pairs needed is approximately equal to the inverse of the probability of the characteristic used; in this case $64/14 \approx 5$ pairs are needed

If a single $S1_K$ value results, it is correct. Go to Step 6

Step 6: At this point we have recovered the 6 bits of the key comprising $S1_K$

Use similar characteristics to recover the 6 bits of key which are XORed with $S2$ through $S8$'s inputs in the first round

Step 7: At this point we have 48 bits of the key which comprise S_K, or equivalently $S1_K$ through $S8_K$

Find the remaining 8 bits of K by exhaustive search over the 64 possible values

Differential Cryptanalysis Compares Pairs of Encryptions

- Differential cryptanalysis compares two related pairs of encryptions
- with known difference in the input $m_0||m_1$
- searching for a known difference in output
- when same subkeys are used

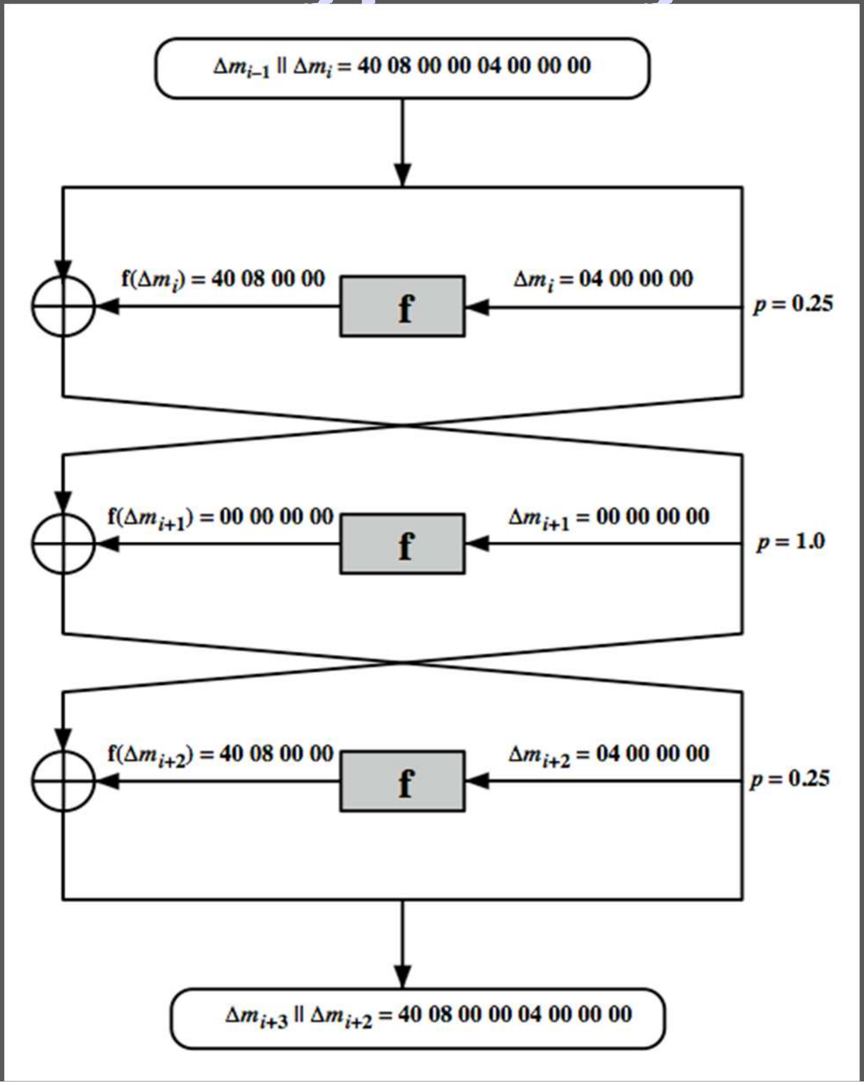
$$\begin{aligned}
\Delta m_{i+1} &= m_{i+1} \oplus m'_{i+1} \\
&= [m_{i-1} \oplus \mathrm{f}(m_i, K_i)] \oplus [m'_{i-1} \oplus \mathrm{f}(m'_i, K_i)] \\
&= \Delta m_{i-1} \oplus [\mathrm{f}(m_i, K_i) \oplus \mathrm{f}(m'_i, K_i)]
\end{aligned}$$

Differential Cryptanalysis

Input round i

Input round i+1

Overall probabilty of given output difference is (0.25)(1.0)(0.25) = 0.0625



Linear Cryptanalysis

- another fairly recent development
- also a statistical method
- must be iterated over rounds, with decreasing probabilities
- developed by Matsui et al in early 90's
- based on finding linear approximations
- can attack DES with 2^{43} known plaintexts, easier but still in practice infeasible

Linear Cryptanalysis

- find linear approximations with prob p != ½

$$P[i_1,i_2,...,i_a] \oplus C[j_1,j_2,...,j_b] = K[k_1,k_2,...,k_c]$$

where i_a,j_b,k_c are bit locations in P,C,K

- gives linear equation for key bits
- get one key bit using max likelihood alg
- using a large number of trial encryptions
- effectiveness given by: $|p-{}^1/_2|$

Block Cipher Design Principles

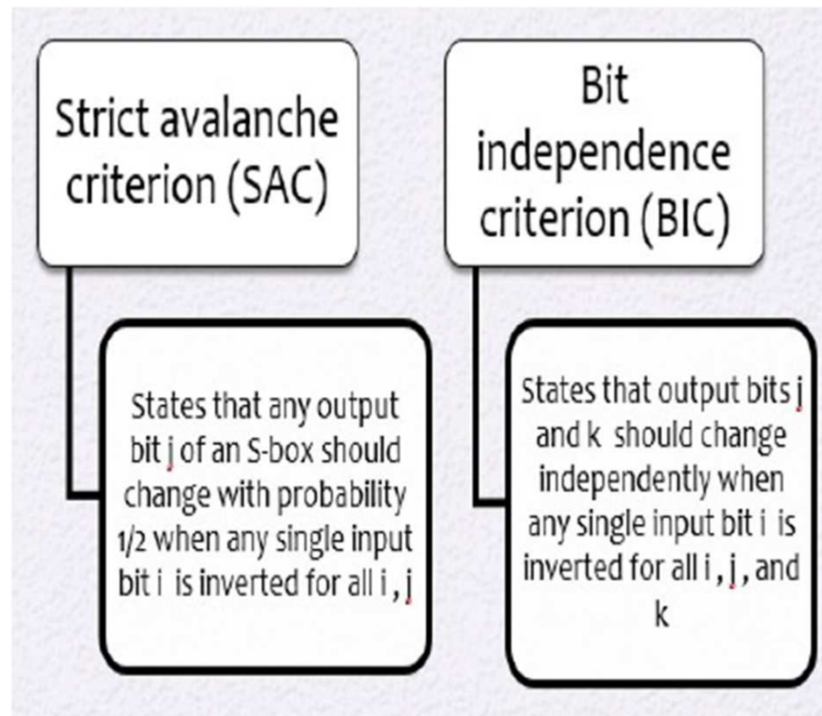
- The cryptographic strength of a Feistel cipher derives from three aspects of the design:
  - the number of rounds,
  - the function F,
  - and the key schedule algorithm.

Block Cipher Design Principles: number of rounds

- The greater the number of rounds, the more difficult it is to perform cryptanalysis, even for a relatively weak F
- In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack
- If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search
- Schneier [SCHN96] observes that for 16 round DES a Differential cryptanalysis attack is slightly less efficient than brute force attack.

Block Cipher Design Principles: Design of Function F

- The heart of a Feistel block cipher is the function F
- The more nonlinear F, the more difficult any type of cryptanalysis will be
- The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function
- The algorithm should have good avalanche properties



Block Cipher Design Principles: key schedule algorithm

- With any Feistel block cipher, the key is used to generate one subkey for each round
- In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key
- It is suggested that, at a minimum, the key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion

Problem Solving-3

(1) This problem provides a numerical example of encryption using a one-round version of DES. We start with the bit pattern for the plaintext, as:

in hexadecimal notation: 0 1 2 3 4 5 6 7 8 9 A B C D E F

in binary notation: 0000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1011 1100 1101 1110 1111

And the 64 bits Key as:

in hexadecimal notation:1 3 3 4 5 7 7 9 9 B B C D F F 1

a. Derive $K1$, the first-round subkey.
b. Derive $L0$, $R0$.
c. Expand $R0$ to get EXP($R0$).
d. Calculate A = EXP($R0$) XOR $K1$.
e. Group the 48-bit result of (d) into sets of 6 bits and evaluate the corresponding S-box substitutions.
f. Concatenate the results of (e) to get a 32-bit result, B.
g. Apply the permutation to get P(B).