Calculation of DES Round Keys (Key Schedule)

The total key including parity bits is 64 bits. The parity bits are bits 8, 16, 24, ... 64. The other 56 bits are the key K. Here is how you compute each K_i :

1) Compute PC-1(K) = C_0D_0, where C_0 is the leftmost 28 bits of PC-1(K), and D_0 is the rightmost 28 bits of PC-1(K). PC-1 is a fixed permutation, provided here:

http://orion.towson.edu/~mzimand/cryptostuff/DES-tables.pdf

2) Here is the computation of the key schedule:

```
for i=1 to 16 {
    $C_i = LS_i(C_{i-1})$
    $D_i = LS_i(D_{i-1})$
    $K_i = PC\text{-}2(C_iD_i)$
}
```

PC-2 is another fixed permutation. LS_i is a left-shift of either 1 bit or 2 bits. If i=1,2,9, or 16, then LS_i is a left-shift of 1 bit. Otherwise it is a two bit left shift.

You can use these directions to completely calculate which bits from the key will appear where in each round key.

When we apply PC-1, our two buffers, C_0 and D_0 look like this:

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36

63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

This represents which bit keys (locations) are now in which locations in C_0 and D_0. For example the bit from location 57 of the key is the first bit in C_0.

Now, when we do a left cyclic shift of one bit (since LS_1 is 1 bit) on both C_0 and D_0, we get the following for these buffers:

49	41	33	25	17	9	1
58	50	42	34	26	18	10
2	59	51	43	35	27	19
11	3	60	52	44	36	57

55	47	39	31	23	15	7
62	54	46	38	30	22	14
6	61	53	45	37	29	21
13	5	28	20	12	4	63

Finally, consider applying PC-2 the current buffer:

PC-2 starts out:

14	17	11	24	1	5	3	28

This means that the first 8 bits from the original key that we grab to start making k_1, the round key for the first round are:

10	51	34	60	49	17	33	57

Thus we would grab the 10th bit of the original key, followed by the 51st bit of the original key, and so on. This means that we can pre-calculate each round key, because we know which positions from the original key to grab each desired bit.

<u>Characteristics of the S-boxes, as pointed out by the NSA</u>

1) Each row is a permutation of the values 0, 1, ..., 15.

2) No S-box is a linear or affine function of its inputs.

3) Changing one input bit to an S-box causes at least 2 output bit changes.

4) For all x, $S(x)$ and $S(x \oplus 001100)$ differ in at least 2 digits.

5) $S(x) \neq S(x \oplus 11ef00)$, for all binary bits e and f.

6) If you fix a single input bit and observe a particular output bit, that output bit is relatively random. (The 32 possible inputs (when fixing a bit) lead to at worst a 13-19 split of 0s and 1s in any particular output bit.)