DES Block Cipher

DES: A remarkable well-engineered algorithm, that's old but had a powerful influence on cryptography (still used in ATM machines)

In 1972, NBS (now NIST) solicited for an encryption algorithm -- IBM responded with their *Lucifer* algorithm

- The key-length is $k = 56$ bits, and a block-length is $n = 64$ bits
- It consists of 16 rounds of what is called a "Feistel network"

Algorithm overview:

function $\mathrm{DES}_K(M)$ $\quad$ // $|K| = 56$ and $|M| = 64$
$\quad(K_1, \ldots, K_{16}) \leftarrow \mathit{KeySchedule}(K)$ $\quad$ // $|K_i| = 48$ for $1 \le i \le 16$
$\quad M \leftarrow \mathit{IP}(M)$
$\quad$Parse M as $L_0 \,\|\, R_0$ $\quad$ // $|L_0| = |R_0| = 32$
$\quad$**for** $r = 1$ to 16 **do**
$\quad\quad L_r \leftarrow R_{r-1}$; $R_r \leftarrow f(K_r, R_{r-1}) \oplus L_{r-1}$
$\quad C \leftarrow \mathit{IP}^{-1}(L_{16} \,\|\, R_{16})$
$\quad$**return** C

Figure 2.1: The DES blockcipher. The text and other figures describe the subroutines $\mathit{KeySchedule}, f, \mathit{IP}, \mathit{IP}^{-1}$.

DES Block Cipher

The *KeySchedule* produces from the 56-bit key *K* (as input), a sequence of 16 subkeys (each 48-bits long), one for each of the rounds that follow

The initial permutation *IP* simply **permutes** the bits of *M* as given in the following table.

Here, the table indicates that bit *1* of the output is bit *58* of the input, bit *2* is bit *50*, ..., bit *64* is bit *7* of the input

The key is NOT involved in this permutation and therefore, this permutation does not appear to affect the cryptographic strength of the algorithm

IP

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

IP^{-1}

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

Figure 2.2: Tables describing the DES initial permutation IP and its inverse IP^{-1}.

The permuted plaintext now enters a loop, which iterates for 16 *Feistel* rounds

Key Recovery Attacks on Blockciphers

S-box functions are applied in these rounds, and are the **heart** of the algorithm

S-boxes are functions taking 6 bits and returning 4 bits, and are basically a lookup-table

One of the design goals of DES is speed, so all functions are easily mapped into hardware

DES is impressively strong -- to this day, the best known attack is still exhaustive key search

NO blockcipher is perfectly secure

Best you can do is make exhaustive search computationally *prohibitive*

But how long does the exhaustive search take?

On average about 2^{k-1} calculations of the blockcipher (worst case is of course 2^k) (directly related to the key size)

Consider DES: with 1.6 Gbit/sec and a plaintext length of *64*-bit, we can perform 2.5 * 10^7 DES computations/sec

Key Recovery Attacks on Blockciphers

To carry out 2^{55} computations (with $k = 56$), we need $2^{55}/(2.5 * 10^7) = 1.44 * 10^9$ seconds or about 45.7 years!

However, recently Electronic Frontier Foundation built a parallel machine for $250,000 that finds the key in 56 hours

The main short-coming of DES was it *key-length* -- this prevented it from resisting exhaustive key searches. Proposed solutions Triple-DES + others