Introduction to Cryptography
Solution to Maman 12

Yehuda Lindell

November 11, 2007

Solution 1: The "complement property" of DES can be used to carry out a brute-force attack in time 2^{55} (instead of the naive 2^{56}). In order to do this, first obtain DES encryptions of m and $\overline{m}$; let c_1 be the encryption of m and let c_2 be the encryption of $\overline{m}$. Next, notice that it is possible to compute a single DES encryption and rule out two keys k and $\overline{k}$. Namely, for any key k, compute $DES_k(m)$. If $DES_k(m) \neq c_1$ then k is clearly not the key (because c_1 is the encryption of m by the oracle). Furthermore, if $DES_k(m) \neq \overline{c}_2$, then $\overline{k}$ is not the key. This latter fact is because if $\overline{k}$ were the key, then it must hold that $c_2 = DES_{\overline{k}}(\overline{m})$ (recall that c_2 is an encryption of $\overline{m}$ with the oracle). Since $DES_k(m) = \overline{DES_{\overline{k}}(\overline{m})}$, we have that $\overline{c}_2 = \overline{DES_{\overline{k}}(\overline{m})} = DES_k(m)$. Given the above, it suffices to traverse *half* of the keys, ruling out two for each encryption. The complexity of the attack is therefore 2^{55} DES encryptions.

Solution 2: 1 gigahertz means that the computer can carry out 10^9 cycles per second. Given the assumption that a single cycle suffices for an encryption, we have that an exhaustive search takes this amount of time:

1. **DES:** for 2^{56} keys we have $2^{56}/10^9 = 72,057,594$ cycles, or equivalently, seconds. This comes out to be 834 days, or about two and a third years.
2. **AES:** for 2^{128} keys we have $2^{128}/10^9$ which comes out to about 10^{22} *years*. Now, for a brute force search there is no reason whatsoever to use AES with keys longer than 128 bits (even though it's true that machines or distributed nets can be used that can compute quicker than this, nothing can get close to this computing power). The only reason that you may want longer keys is to preempt any possible algorithmic/cryptanalytic shortcut that may be discovered in the future.

Solution 3: The first method is exactly the same as basic 2DES. Therefore it is still vulnerable to a meet-in-the-middle attack. The second method contains inside it the computation $DES_{k_1}(DES_{k_2}(DES_{k_1}(x)))$ and is therefore no worse than 3DES with two keys (a method that is believed to be very strong). Note that 3DES uses DES-inverse for the middle computation but this makes no difference to the security.

Solution 4: If all of the round keys are identical and the permutation is the identity permutation, then each of the m blocks of the plaintext are affected by only ℓ bits of the key (and one S-box). Therefore, it is possible to separately guess the ℓ bits of the key for each block, and verify if the guess is correct. Since it takes 2^ℓ guesses at the most, and this is carried out separately for each of the m blocks, we have that the entire key may be recovered in time $m \cdot 2^\ell$. From here it is clear that the role of the permutation π_P in an SPN is to ensure that all parts of the ciphertext are affected by all the bits of the key.

Solution 5:

1. The permutation IP has no effect on the security of DES. In order to see this, note that it is known and unkeyed. Thus, a plaintext/ciphertext pair for the modified DES can be transformed into a plaintext/ciphertext pair for the original DES by just computing IP (specifically, for a pair (m, c) from the modified DES, compute $m' = IP(m)$ and $c' = IP^{-1}(c)$ to get a pair (m', c') for the original DES).

2. First, if you look at two plaintexts that differ only in the first bit, we obtain that the outputs will differ only in the first bit. Thus, the cipher is clearly not a random-looking function. To make things worse, it is possible to isolate bits of the key as in solution 4 to learn the key (I leave the details to you).