Double DES

$$p \rightarrow E(k_1, \quad p) \rightarrow E(k_2, \quad E(k_1, \quad p)) = C$$

Double DES has a 112-bit key and enciphers blocks of 64 bits.

DES is not a group; i.e., $E(k_2, \quad E(k_1, \quad p))$ is not equivalent to DES encryption using a single key. Recall that, for example, the Caesar cipher is a group. If a message were encrypted with the Caesar cipher with a key of 3 and then re-encrypted with the Caesar cipher with a key of 5, the result is equivalent to encrypting the message with the Caesar cipher with a key of 8. For the Caesar cipher, double encryption does not increase security. DES is not a group; double encryption is not equivalent to single encryption. Security does increase by double encryption, but it does not increase much.

The security of DES depends on its having a large key space; so large that (at least when it first began being used in the 1970's a brute force attack was not practical [that has now changed]). Recall that DES has a 56-bit key (the key is actually 64 bits, but every 8^{th} bit is a parity check; so, only 56 or the 64 bits are meaningful); therefore, the size of the key space is $2^{56} = 72,057,594,037,927,936$. Recall that the algorithm that was originally proposed had a 128-bit key, but the size of the key space was reduced by the NSA (for some reason).

Intuitively, double encryption should double the size of the key space. But, that is not the case with DES.

Meet-in-the–middle attack on double encryption

This attack requires knowing some plaintext/ciphertext pairs. Let's assume that we have a plaintext/ciphertext pair; i.e., we know the plaintext p and the corresponding (double DES enciphered) ciphertext C. Attacks on DES have typically been brute force attacks (see "Breaking DES"); so, we will use brute force here.

Here is the double encryption:

$$p \rightarrow E(k_1, \quad p) \rightarrow E(k_2, \quad E(k_1, \quad p)) = C$$

Encrypt p using all 2^{56} possible keys, and store the results. (Storage could be a problem.) The stored results will include all possible encryptions $p \rightarrow E(k_1, \quad p)$.

Then decrypt C using all 2^{56} possible keys.

$$D(k_2, \quad C) = D(k_2, \quad E(k_2, \quad E(k_1, \quad p))) \rightarrow E(k_1, \quad p)$$

After decrypting with each key, check for a match with the stored outputs of the 2^{56} possible encryptions. When we have a match, we have located a possibly correct pair of keys. Now, perhaps more than one pair of keys will result in a match, but the number of pairs of keys that return matches should be small. We could try each possible pair of keys. If more than one plaintext/ciphertext correspondence is known (for the key pair), then other correspondences could be used to check which of the keys is correct.

So, it only takes twice as long to break double DES using brute force. Because DES has 56-bit security, double DES has $2 \times 2^{56} = 2^{57}$ security.

Triple DES – 3DES

3DES was developed in 1999 by IBM – by a team led by Walter Tuchman. 3DES prevents a meet-in-the-middle attack. 3DES has a 168-bit key and enciphers blocks of 64 bits. 3DES effectively has 112-bit security.

3DES can be done with 2 or 3 keys.

3-key encryption

$$E\left(k_3, \quad D\left(k_2, \quad E\left(k_1, \quad p\right)\right)\right)$$

2-key encryption

$$E\left(k_1, \quad D\left(k_2, \quad E\left(k_1, \quad p\right)\right)\right)$$

Why would we want to do decryption as the second step? One reason might be that by taking $k_2 = k_1$; 2-key, 3DES becomes single DES with key k_1. 3DES can communicate with single DES.