2DES: A meet-in-the-middle attack has 2^60 bits of effective security. It is possible to break the encryption using approximately 2^60 operations.

For modern computers, this is not a reliable form of encryption anymore so the solution to fixing 2DES was 3DES. 3DES uses 3 steps of encryption instead of 2 steps of encryption. For 3DES, you need to pick two keys k_1 and k_2. Then the plaintext P is encrypted using the following function from DES:

$$E_{k_i}(P) = C$$

And the ciphertext is decrypted using the following function:

$$D_{k_i}(C) = P$$

To encrypt P:

$$C = E_{k_1}(D_{k_2}(E_{k_1}(P)))$$

To decrypt C:

$$P = D_{k_1}(E_{k_2}(D_{k_1}(P)))$$

For 3DES, a man-in-the-middle attack does not work because there are

$$2^56 * 2^56 = 2^112$$

effective bits of security. So, this attack is not plausible for modern attack.

3DES is still considered secure and gets used today, but is not recommended for new applications.

How do we encrypt things that are larger than 64 bits? 1 Method would be to break the plaintext into 64 bit chunks and encrypt each chunk separately.
Modes of Operation:
The method of breaking the plaintext into 64 bit chunks and encrypting each block separately is called Electronic Code Blocking (ECB).
Another mode is Cipher Block Chaining (CBC): This is when the plaintext is broken up into chunks

$$p_1, p_2, ..., p_n$$

. Then, to encrypt: Fix c_0 = all zeros or some random string of bits, then

$$c_j = E_k(P_j \oplus C_{j-1})$$

To decrypt: Send c_0 in plaintext and compute

$$P_j = D_k(C_j) \oplus C_{j-1}$$

Cipher Feedback (CFB): This mode of operations introduces 2 functions: head(X) which return the first n bits of a string X (n depends of the encryption algorithm that you are using; so for DES n=8 or for SDES n=4) and tail(X) which returns the string of bits after the first n bits.

Ex (using DES):

$$head(101100001011) = 1011$$

$$tail(101100001011) = 00001011$$

For CFB, we will break our plaintext into chunks of length X = length(head(X)). Fix an initial X_0 of the same length as the block. Then compute:

$$O_i = head(E_k(X_i))$$

$$C_i = P_i \oplus O_i$$

$$X_{i+1} = tail(X_i)||C_i$$

where || is the append operation. And to decrypt, we will compute

$$P_i = C_i \oplus O_i$$

Output Feedback (OFB): Pretty much the same setup as CFB. But,

$$O_i = head(E_k(X_i))$$

$$X_{i+1} = tail(X_i)||O_i$$

$$C_i = P_i \oplus O_i$$

Some benefits of OFB is that there is no error propogation. Since O_i doesn't depend on the plaintext or the ciphertext it can be precomputed. Then the only step for encryption is to compute

$$P_i \oplus O_i$$

This is also referred to as a stream cipher.
Counter (CTR): This is another stream cipher. First you initialize X_0 then compute

$$C_i = P_i \oplus E(X_i)$$

$$X_i = X_i + 1(mod 2^{blocksize})$$

Advanced Encryption Standard (AES): The NIST put a call out for proposals for a new encryption standard. The system that the NIST chose was Rijndael. AES is not a feistal system (to make encryption faster). This was designed to be completely open so there are no backdoors hidden anywhere. Also designed to be faster and more secure that the current encryption standard (DES).

An example of a simplified version of AES called SAES. This simplified version of AES follows this pattern: Plaintext => Add round key => Round 1 (Substitute => Shift Rows => Mix Column => Add round key) => Round 2 (which consists of Round 1 not including the "mix columns step"). Taking a look at what the "substitute" step does: SAES has 1-box which takes in 4-bits and

returns 4-bits. The four bits that it returns $b_0b_1b_2b_3$ are the coefficients of a degree 3 polynomial in F_{16}. The irreducible prime polynomial is: x^4+x+1. So the steps for the sbox are: compute $(b_0x^3+b_1x^2+b_2x+b_0)^{-1} = c_0x^3+c_1x^2+c_1x+c_0$. Then,

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

And the output of the sbox is: $d_0d_1d_2d_3$.

Notes for 4/6/17

Kevin Shhultis

April 13, 2017