Modes of Operation

Camryn Truban

March 29, 2020

1 Modes of Operation

Problem: We have more plaintext thenfits into one block.

Break ciphertext int o multiple blocks

$$P_1, P_2, P_3$$

How do we encrypt all of these blcoks of plaintext?

2 Electronic Code Book(ECB)

$$C_i = E_i(P_i)$$

Each block of plaintext is encrypted seperately.
Benefit: Super easy
Downside: The same block will always encrypt to the same block of ciphertext.

3 Cipherblock Chaining(CBC)

Start with an initial
C_0 - Random block
Sent in clear text(Unencrypted)
Method of encryption

$$C_i = E_k P_i \oplus C_(i-1)$$

Because each plaintext gets *XOR'd* with the previous sciphertxt before encypting even if we sort the same plaintext over and over each time it would get *XOR'd* with a different ciphertext and so the results would always be different.

Note that bob knows the values of all the C_i
To decrypt Bob computes

$$D_x(C_i) = P_i \oplus C_(i-1)$$
$$P_i = D(C_i \oplus C_(i-1)$$

4 Cipher Feedback(CFB)

C_0 - Random sent in clear text
Instead of encrypting the plaintext we use our encryption algorithm to generate a random stream which will encrypt the plaintext like a one-time-pad
To encrypt

$$C_i = E_k(C_(i-1) \oplus P_i$$

Note the plaintext is outside the encryption!
"Encryption is by *XOR'd* with the "random" string generated Ek(Ci)

5 Output Feedback(OFB)

O_0 = random block sent in clear text To encrypt

$$O_i = E(O_(i-1))$$

$$C_i = P_i \oplus O_i$$

To Decrypt:

$$O_i = E_k(O_(i-1))$$
$$P_i = C_i \oplus O_i$$

Benefit: All of the output blocks Oi can be pre-computed
Good for streaming or other mediums which require lots of blocks to be encrypted quickly.

6 Counter(CTR)

X_0 = All zero or random number
To encrypt

$$X_i = X_(i - 1) + 1$$
$$C_i = E(X_i) \oplus C_i$$

Benefit of CTR is that any block can be encrypted or decrypted without computing all imtermediate blocks
Also it doesn't have the problem that one mistake along the way messes up all future blocks(Problem especially for CBC)
Most websites use GCM Which is basically same as CTR using a finite field

---

Recalll SDES
12 bit block size
9 bit master key
3 rounds
Actual DES
64 bit blocks
56 bit master key(64 bits 8 check bits)
Steps of DES are bascially the same as SDES but there was one extra step before the first round the bits we permuted using an initial permutation
No cryptographic purpose just faster to load MTO memory on 1970's era hardware.
Since master keys are 56 bits there are 2^{56} possible master keys.
Brute force attacks require 2^{56} different necryption.
By the 1990's it started to become feasible to perform such an attack
The electronic frontier built a super computer specifically to attack DES in the late 90's, could brute force 1 DES key in about 24 hours.
At this point DES was no longer considered secure.