Class Notes, 10-20-2016

Dalton Watts

November 22, 2016

DES uses a 56 bit key. 2^56 possible keys $7.2 * 10^{16}$ Brute force attack against DES try all possible keys 1990s first specialized computers could brute force DES in a few days. Now a few hours. Double encryption with two different keys k, k‘ (he’s writing k1, k2, but that conflicts with the SDES notes) Encrypt plaintext using $E_{k'}(E_k(P))$ Decrypt ciphertext using $D_k(D_{k'}(C))$ Naively it seems like this is much more secure. Brute Force attack: 2^{56} possibilities for k and 2^{56} possiblities for k‘. Combined, it’s 2^{112} possibilities. $5.2 * 10^{33}$

Brute force doesn’t work well against Double DES. It’s still ridiculously long to solve, even today. Sadly, there’s a problem with a ”Meet in the Middle” attack. It’s a Known Plaintext attack Know that the plaintext P encrypts to C and P‘ encrypts to C‘

take P and compute $E_k(P)$ for all 2^{56} possible keys. Put into table 1 take C and compute $D_{k'}(C)$ for all 2^{56} possible key‘s. Put into table 2

C = E‘(E(P)) D‘(C) = E(P)

So, link the two tables and find where they’re equal! Any pair of entries gives possible values for k and k‘

Suppose Encryption function produces an essentially random string of bits If we fix k and k‘, What is the probability that D‘(C) = E(P)? Works out to be $\frac{1}{2}^{64}$ probability that these are the same.

2^{112} possible pairs of k and k‘. So, we expect to narrow it down to 2^{48} pairs

Repeat for (P‘, C‘) D‘(C‘) = E(P‘)

Then the probability that it’ll match up in all four tables is $(\frac{1}{2}^{64})^2$

$\frac{1}{2}^{128} * 2^{112} = 2^{-16}$

It’s likely that you will have just one pair of k and k‘ remaining, which should be the set of keys used in the encryption. Using this strategy on DDES is equivalent to doing 4 brute force attacks on regular DES encryptions (1 for each table) Effective security is equivalent to $4 * 2^{56} = 2^{58}$ bit key.

What about Triple DES? Pick 3 keys, k k‘ k“ (compared to the board, he’s still using numbers, so k = k1, k‘ = k2, k“= k3) Encryption is E“(E‘(E(P))) Decryption is D(D‘(D“(C)))

If we try to do a meet in the middle attack against TDES, we’d have to create tables for: E‘(E(P)) however, this is $2^112 entries, which is outside the capabilities of modern computers. D“(C) 2^56$

So, TDES is secure for now. (I would personally expect that QuadDES or QuintDES would be even safer for little extra cost.) Effective Key Length of

TDES is like a 2^{112}-bit key. In practice, they use a trick so that only 2 keys are stored in TDES today, k and k‘. The encryption function is E(D‘(E(P)) Decryption function is D(E‘(D(C)))

Another method used today is DES-X Choose 2-64 bit keys k and k‘ and 1-56 bit key k“

Encryption function is k $\oplus$ E“(P XOR k‘)

Down side to DES is that messages must be 64-bits long. How do we send longer messages? Split the bits into different parts, and if one segment is not a full 64-bits, pad it with a bunch of 0’s

Modes of Operation: Electronic Codebook (ECB) Break our message into blocks of length 64. Encrypt each block separately using the same keys.