Class Notes 4/4/17

Ross Jones

April 20, 2017

Summary: On this day we discussed the differences between DES and SDES encryption. We were also taught about the Differential Cryptanalysis of DES and the strengths of Double DES.

Class Notes:

Main differences in DES:

- DES has 64 bit blocks (12 in SDES) with 56 bit keys
- There is an initial permutaion of the plaintext before the first step (put the bits in a fixed order other than how they started out)
- L_o and R_o have 32 bits each
- 16 rounds of encrpytion

The DES Function Steps:

1. Expander function takes 32 bits $\rightarrow$ 48 bits
2. Use the 56 bit master key to make 48 bit round keys
3. Then XOR E(R_o)$\oplus$(k_i)
4. Break into 8 pieces (6 bits each) and feed pieces into 8 different S-boxes
5. Each S-box returns 4 bits then put pieces back together
6. XOR this with L_1 and this becomes the new R_{i+1}

The DES Differential Cryptanalysis:

-The Differential Cryptanalysis is slower than brute force against DES with 16 rounds, but would have been faster had DES used only 15 rounds.

-To attack DES, brute force and we try all 2^{56} possible keys.

-In the early 90's, Electronic Frontier Foundation built a supercomputer that could do one DES brute force attack in 24 hours, today it is possible in 2-3 hours, making it insecure.

-To make DES more secure from today's computers without changing the widely used algorithm, a double encryption is used that requires much more computing power.

Double DES:

1. Pick 2 different 56 bit keys, k_1, k_2
2. DES uses the expander function E_k(P)
3. To perform 2DES, we encrypt plaintext P using C=E_{k2}((E_{k1}(P))

A few question that we ask ourselves about 2DES are

-Is this any different from doing DES with some other key?

-If we pick k_1, k_2, does there exist a key k_3 such that

E_{k2}((E_{k1}(P))=E_{k3}(P) * ?

Note that * is true for Caesar, Affine, Vigenere, and Hill Ciphers but not for DES.

How to Attack 2DES:

-Brute force will not work because 2DES requires trying every possible pair of keys k_1, k_2.

-This means there are $2^{56} * 2^{56} = 2^{112}$ possible pairs, which is too big for any supercomputer today

-2DES can be attacked using a meet-in-the-middle attack which work against <u>any</u> double encryption

Meet-in-the-middle Attack:

-Meet-in-the-middle is a known plaintext attack and must know two plaintexts P_1, P_2 and their corresponding ciphertexts C_1, C_2

$C_1 = E_{k2}(E_{k1}(P_1))$

$C_2 = E_{k2}(E_{k1}(P_1))$

Goal: Find k_1 and k_2

-Now take the decryption function of both equations using k_2 (D_{k2})

$D_{k2}(C) = D_{k2}(E_{k2}(E_{k1}P)))$

$D_{k2}(C) = E_{k1}(P)$

-Then create 2 tables with 2^{56} rows containing every possible key

-In the first table, solve for every possible value of $D_{k2}(C)$

-In the second, solve for every possible value of $E_{k1(P)}$

-Look for rows that appear in both tables

What is the probability that 2 rows agree? $(\frac{1}{2})^{64}$ or $\frac{1}{2^{64}}$

-Since there are 2^{112} pairs of rows, we expect $2^{112} * \frac{1}{2^{64}} = 2^{48}$ different matching rows.

-Repeat this again using P_2 and C_2

-The probability that k_1 and k_2 match in both sets of tables is $\frac{1}{2^{16}}$ and should only happen once corresponding to the actual values of k_1 and k_2

How many steps is this?

- 2^{56} rows in each table
- 2^2 tables
- Search through these tables for matches are 4 (2^2) times additional work
- This adds to 2^{60} operations which is still reasonable for a supercomputer today