ECE6280
Project 2 (Revised to add more detail to Section 2)
Due Monday October 6, 2019

Revised Oct 1, 2019

1 Problem 1 (15 points)

Do Stinson, 4th edition, 4.16

2 Problem 2 (30 points)

The design of substitution boxes is a crucial part of block cipher design. Please read the posted article Optimal-4bit-Sbox-2011-2018.pdf . Your task is to analyse a specific S-box, "S3" from Serpent. Serpent uses 8 substitution boxes, but only one is "golden" or optimal. You will use the criteria established in the article to show that Serpent is golden. You can refer to the Serpent website to get the description of S3.

These computations can be done on paper or by Matlab. More specifically,

- Show that Serpent S3 is optimal against differential attacks. You construct the input/output difference table in the same way as on Table 4.4 in the text (Page 101, 4th edition). The only difference is that you divide each tally by 16 to get a p value as in Definition 1 in the Saarinen article. Each element in the table should be less than or equal to 1/4.

- Show that Serpent S3 is optimal against linear attacks. This involves constructing the linear table as was done in class and that you can see in the slides. Then take these tallies, divide by 16 and subtract one half. The largest bias ϵ should be less than or equal to 1/4 for the S-Box to be optimal.

- Show that the branch number is 3. Branch number equation in the article is $BN = \min_{a,b,a\neq b}(wt(a \oplus b) + wt(S(a) \oplus S(b)))$ over all the inputs a and b are the possible different 4-bit inputs to the S-Box, and wt is the Hamming weight (number of non-zero bits). Branch number is about how often 1 input bit splits to 2 or more output bits. This makes it hard to construct a path through the S-Box that has only one output. If the branch number is high, then the number of S-Boxes in the final linear or differential attack will be high, which raises the number of input/output pairs needed to be able to detect whether a partially guessed subkey is a correct guess or not.

- Show that all output bits have algebraic degree 3 and are dependent on all input bits in a nonlinear fashion.

  Here is an example of a 3 bit S-Box where I calculate the algebraic degree of one of the output bits.

  Here, you can use the truth table to construct a binary algebraic representation of the truth table. The general algebraic form for the first output bit y_1 is $y_1 = c_{1,2,3}x_1x_2x_3 + c_{1,2}x_1x_2 + c_{1,3}x_1x_3 + c_{2,3}x_2x_3 + c_1x_1 + c_2x_2 + c_3x_3 + c_0$ If you do this for y_1, you end up with $y_1 = x_1x_2 + x_1x_3 + x_1 + x_2 + 1$. The method is to start with the constant term, i.e. set $x_1 = 0$, $x_2 = 0$, $x_3 = 0$, then the first row tells us that the constant is $c_0 = 1$. Then set $x_1 = 0$, $x_2 = 0$, $x_3 = 1$, to use the second row to give that $c_3 = 0$. The main point is that y_1 has monomials of second degree in it, x_1x_2 and x_1x_3. Because they are of second degree the S-box is as resistant to algebraic attacks as a 3-bit s-box can be. Furthermore, the nonlinear terms $x_1x_2 + x_1x_3$ contain all input bits. Your job is to make sure that for the Serpent S3 box, that the 4-bit S-Box output bits have algebraic degree 3, i.e. are maximally resistant to algebraic attacks and that the nonlinear terms contain all input bits.

Input	x_1	x_2	x_3	Output	y_1	y_2	y_3
0	0	0	0	5	1	0	1
1	0	0	1	6	1	1	0
2	0	1	0	3	0	1	1
3	0	1	1	1	0	0	1
4	1	0	0	2	0	1	0
5	1	0	1	7	1	1	1
6	1	1	0	0	0	0	0
7	1	1	1	4	1	0	0