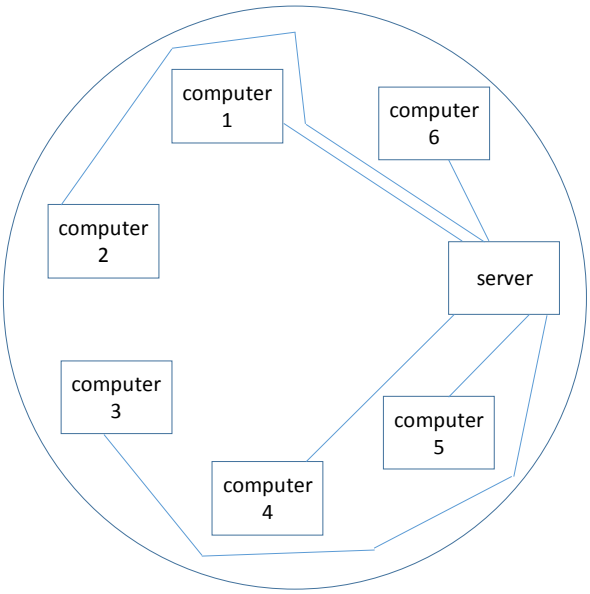
MODD / Mr. Hansen
12/17/2015

Name: _____KEY_____

Test #2 (100 points)
Calculator Allowed

1.(a) What term do we use for the logical configuration of a network, which may or may not resemble the actual physical layout of the computers and servers that make up that network? *Hint:* It is a branch of mathematics that starts with the letter T. _____topology_____

(b) A *ring* network is one in which the computers are connected in a circular chain, each computer linked to the next. The network below certainly looks like a ring, but is it a ring? ___no___ If not, what is it? ___a star___



2. Use $X(0) = 17$ as the starting point for a pseudorandom number generator (PNG). What do we call this starting value? Answer: the _seed_ . Then, use values of $A = 179$, $B = 773$, and $N = 256$ to execute 5 iterations of the PNG to generate 5 outputs. Convert each output to hex and then to an 8-bit value. Chain all 5 of your 8-bit values together to make a 40-bit bitstream. Show all work below. Remember, the next X is always found by the formula $(AX + B) \bmod N$, utilizing the previous X. When counting to 5, do not count the $X(0)$ value of 17. In other words, generate 5 new values of X in order to form your 40-bit bitstream.

			output =	output	output						
i	**X(i)**	**179X(i) + 773**	**result mod 256**	**(Hex)**	**(Binary)**		40-bit bitstream generated from the 5 outputs:				
0	17	3816	232	0xE8	11101000		1110100000111101101011000100100100010000				
1	232	42301	61	0x3D	00111101						
2	61	11692	172	0xAC	10101100						
3	172	31561	73	0x49	01001001						
4	73	13840	16	0x10	00010000						
5	16	3637	53								

3. Use your ASCII table to encode the message

STA#1

as hex. Then XOR that message with the 40-bit bitstream you created in question 2 (or just 40 made-up bits if you couldn't answer question 2) to create an encrypted bitstream. **Give your encrypted bitstream in both binary and hex.**

STA#1 = 0x5354412331 = 0101 0011 0101 0100 0100 0001 0010 0011 0011 0001

40-bit bitstream (earlier) = 1110 1000 0011 1101 1010 1100 0100 1001 0001 0000

Result of XOR operation =1011 1011 0110 1001 1110 1101 0110 1010 0010 0001

Result of XOR operation (in hex) = 0xBB69ED6A21

4. Explain briefly how the recipient should proceed in order to decode whatever you produced in question 3. Be sure to explain how the recipient can somehow manage to have a 40-bit bitstream that would match the one you produced in question 2. If you couldn't answer #2 or #3, simply explain what *would happen* if everything were working correctly.

The sender and recipient need to have a shared secret key. This can be accomplished either by direct exchange (old school) or by a PKI (much better, since a session key can be generated even if the parties have never previously met, and the session key never has to be communicated between them). The shared secret key (17 in our example) can then be used as the seed for a PNG. When the recipient generates the 40-bit bitstream, the result should be the same as what we found at the end of question 3. The recipient then performs an XOR operation of that 40-bit bitstream with the garbled message (0xBB69ED6A21), and voila! The output should be 0x5354412331, which represents the ASCII message STA#1.

5. What is even parity used for (2 words)? __error___ _detection_ Explain briefly how even parity works.
The parity bit is added, usually at the beginning or end of a byte, so that the total number of 1's, including the parity bit, is even. For example, 0xAE is binary 10101110, which has an odd number of 1's. Therefore, the byte 0xAE, if communicated using an even-parity protocol, would be considered an error. However, 0xA9 is binary 10101001, which has an even number of 1's, as desired, and would therefore pass the parity check.

6. State 2 examples of lossy compression (___MP3_ and __JPEG____) and one example of lossless compression (__RLE, ZIP, etc.__).

7. What name do we give to a form of "data hiding" in which the secret message is hidden or interspersed within a seemingly innocent-looking file? _____steganography__ *Hint:* The word starts with the letter S.

8. Convert each of the following power ratios to dB: (Write +__dB or –___dB in each case.)

   350 W increased to 700 W = ___+3 dB___

   200 mW decreased to 1 mW = __–23 dB_____

   5 mW increased to 1 GW = __+113 dB____

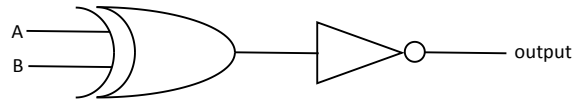
9. Convert each of the following decibel changes to a power ratio:

   +6 dB = ___4__ : ___1___

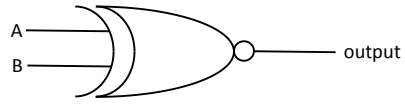
   –45 dB = __1____ : _30,000_

   +10 dB = __10__ : __1____

10. Sketch a circuit diagram for ~(A xor B) below. Then, use a truth table (either below or on the reverse side of this sheet) to prove that ~(A xor B) is always equivalent to saying that A and B have the same truth value.



Also OK:



A	B	A xor B	~(A xor B)	A equ B
1	1	0	1	1
1	0	1	0	0
0	1	1	0	0
0	0	0	1	1

Since the last 2 columns are always the same, for any values of A and B, the ~(A xor B) circuit is the same as testing for whether A and B are equivalent.