Section 1: One-Time Pad

1. Who is your partner?

Josh Rymkiewicz

2. If we use XOR as our **encryption** operation, how will **decryption** be performed? (hint: work together to figure out what logical operator will take c [YOUR OPERATOR] k and produce m, the original message. Use a "mini message" and key like "1100" and "1010" with your partner to figure this out.)

oldString 1100
Key 1001
XOR 0101 (newString)

newString 0101
Key 1001
XOR 1100 (oldString)

You would run the key on the encrypted message for the XOR operation.

3. Agree on a secret key that is made of 3 * 8 bits (you'll be encoding & decoding a 3 letter long message). What is your key?

1110 1101 1000 1100 0101 1111

4. Choose a secret message made of three uppercase letters A-Z. Do not tell your partner what it is! Write down your message & convert it to binary here.

0101 0011 0101 0101 0101 0000

5. Encode your message using your answers from 3 & 4 and the XOR operator. Write your encoded message (your cipher text) here.

0101 0011 0101 0101 0101 0000

1110 1101 1000 1100 0101 1111

1011 1110 1101 1001 0000 0000

6. Tell your partner your encoded message. Write down the cipher text that your partner gave you.

1010 1111 1100 0101 0001 1000

1110 1101 1000 1100 0101 1111

0100 0010 0100 1001 0100 0111

7. Decode the cipher text that you received from your partner using the operation identified in question #2. Write that down here and translate the decoded message back to ASCII.

BIG

Section 2: One-Time Pad—security & usage

1. What happens if we use the same One Time Pad twice? What information would an eavesdropper have access to?

There could be a security threat, an eavesdropper would be able to get the encrypted messages and discover the key.

2. What information about the messages would the eavesdropper be able to recover? (What is *c1* XOR *c2* equivalent to? Where *c1* is the first cipher text and *c2* is the second cipher text)

The eavesdropper would be able to find out the key and decode any interaction between parties.

3. If I have a message of n bits, how many bits must my One Time Pad be?

n length

Section 3: Diffie-Hellman key exchange

---

1. (part 1 of shared common knowledge) With your partner, pick a prime number, *p.*Write it down. We recommend a number larger than 2 but smaller than 20, but feel free to use wolfram alpha or your favorite calculator later if needed!

   p=7

2. (part 2 of shared common knowledge) Pick a primitive root modulo *p*. This is an integer *r* between [1, *p* - 1] such that the values of (*r* ^ *x*) % *p* for all *x* in range [0, *p* - 2] are different. Write this down. You may need to test out different numbers!
   example: if p is 2, and you want to know if 7 is a primitive root modulo 2, calculate (7 ^ 0) % 2, (7 ^ 1) % 2, (7 ^ 2) % 2, (7 ^ 3) % 2, (7 ^ 4) % 2, (7 ^ 5) % 2. If the answers are **all different**, 7 is a primitive root modulo 2. (this is an example of a number that is not).

   r = 3

3. Choose a secret number. Do not share it with your partner! Write it down. (this is your private key)

   19

4. Compute your public key. Do this by computing (*r* ^ (*private key*)) % *p*. Tell your partner your public key. Write down the public keys here.

   Public key is 3

5. Compute your shared secret. Do this by computing ((*your partner's public key*) ^ (*your private key*)) % *p*.

   4^19 % 7

6. Compare your answer from #5 with your partner. Were they the same?

   Yes they were the same, we got both got 4.