Question 1: A definition for a secure cryptographic scheme: It's the year 1940 and as a budding cryptographer you've heard that the Enigma cipher has been broken. You're unsure if this is even possible, but realize that without a proper definition of security there's no way to mathematically prove whether a particular crypto scheme is secure.

Let's come up with a definition of security and see if it holds up to the properties we want a secure system to have: At a high level, we want our scheme to ensure secrecy of communication against an eavesdropper who can observe everything being sent across the channel between Alice and Bob.

Definition 1: It should be impossible for the attacker to determine the key shared by the parties.

Describe potential flaws with this definition

- A. Too broad
- B. Too narrow
- C. Other issues

Definition 2: An encryption scheme is secure if and only if it is impossible for the attacker to learn the plain text.

Describe potential flaws with this definition

- D. Too broad
- E. Too narrow
- F. Other issues

Definition 3: An encryption scheme scheme is secure if it is impossible for the attacker to learn any character of the plain text.

Describe potential flaws with this definition

- G. Too broad
- H. Too narrow
- I. Other issues

Question 2: One Time Pads: In modern cryptography, we transform our "plaintext" to "ciphertext" using bitwise operations. The most commonly used bitwise operation in cryptography is the exclusive OR operator or XOR. Here's a review of XOR.

The XOR operator takes two bits and outputs one bit:

0 🕀 0 = 0
0 🕀 1 = 1
1 🕀 0 = 1
1 🕀 1 = 0

Useful properties of XOR:

x ⊕ 0 = x
x ⊕ x = 0
$x \oplus y = y \oplus x$
$(x \oplus y) \oplus z = x \oplus (y \oplus z)$
$(x \oplus y) \oplus x = y$

Question 2 Part A :Let's try out an example:

0100 1000 ^ 1010 0001 = ?

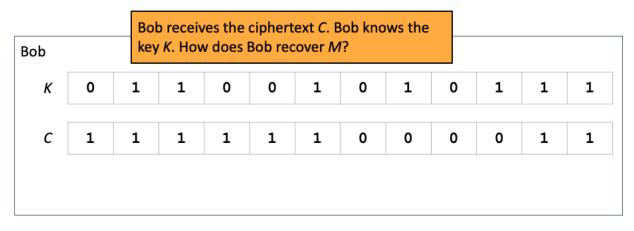
Question 2 Part B :The one time pad is a simple and idealized encryption scheme that helps illustrate some important concepts in security. What is the XOR of Alice's message with the key?

One-Time Pads: Encryption

Alice	The plaintext <i>M</i> is the bitstring that Alice wants to encrypt.						Idea: Use XOR to scramble up <i>M</i> with the bits of <i>K</i> .					
к	0	1	1	0	0	1	0	1	0	1	1	1
м	1	0	0	1	1	0	0	1	0	1	0	0

Question 2 Part C : Can Bob recover Alice's plaintext? Try using XOR of the Key and Ciphertext and see if you can retrieve the plaintext.

One-Time Pads: Decryption



Question 2 Part D : Is a one-time pad secure?

- What if the OTP Key is less than the length of the message? What if we reuse the OTP key?
- What are some issues that you can identify with OTPs?

Question 3: Random Number Generators

Part A: What are good examples of random number generators? Where does randomness come from?

- A. Radioactive Decay
- B. RF emissions from the big bang
- C. Thermal Emissions
- D. Mouse and Keyboard movements

Part B:Below is the source code for C's random number generator. Under what circumstances is this random number generator "truly random"?

```
unsigned long int next = 1;
/* rand: return pseudo-random integer on 0...32767 */
int rand(void){
    next = next * 11-3515245 + 12345;
    return (unsigned int) (next/65536) % 32768;
}
/* srand: set seed for rand() */
void srand(unsigned int seed){
    next = seed;
}
```

Part C: We've seen that random number generators are generated on most machines from physical inputs such as mouse clicks etc .How are random numbers generated for Virtual Machines and Servers?