Question 1:

1(a)

[10 points] A cryptosystem that offers perfect secrecy prevents an eavesdropper who observes an encrypted transmission from learning anything about the plaintext, other than its size. Show with a counterexample that the Substitution Cipher doesn't provide perfect secrecy.

A substitution cipher doesn't offer perfect secrecy as it is still vulnerable to frequency analysis attacks (including identifying patterns of characters in sequence).

For instance, a simple cipher text "SGGA Q VGGSSN DQDDGZI" shows a few patterns that can be used to narrow down the substitutions. Immediately, we see a single letter Q - this must be either I or A. Next there is a double G in the first word, double G and double S in the second word, and double D in the last word, which immediately limit what G, S and D most likely are.

In addition, with a longer cipher text, frequency analysis would suggest that the most frequent letter to occur is E in plain text, and so on. As more letter substitutions are guessed, the remaining substitutions become easier to decode based on the letters around them.

In this case, it would be feasible to decode this cipher text to "LOOK A WOOLLY MAMMOTH", simply with the occurrences of double letters and reuse of letters in these words, even without enough characters to do a full frequency analysis.
As even just the one letter word "Q" narrows down this letter to being A or I, substitution cipher does not provide perfect secrecy.

1(b)

[10 points] Consider the following modification to one-time pad (OTP) encryption. Rather than share a single one-time pad, Alice and Bob have shared knowledge of two pads, P_1 and P_2. Given a plaintext M, Alice creates the ciphertext $C = M \oplus P1 \oplus P2$, where $\oplus$ denotes xor and $|M| = |P1| = |P2|$ (i.e., the size of the message and the two pads are all equal). To decrypt, Bob takes the ciphertext and xors it with $P1$ and $P2$; i.e., $D(C) = C \oplus P1 \oplus P2$. Argue that if a one-time pad offers perfect secrecy, then the above scheme must also be perfectly secure.

As a one-time pad offers perfect secrecy, so does the above scheme.

This is because $P1 \oplus P2$ is the XOR of two binary objects of the same length, and therefore the result is also a binary object, $P3$ of the same length.

This means that $C = M \oplus P1 \oplus P2 = M \oplus P3$, i.e. this operation is the same as using a single one-time pad, and therefore offers the same level of secrecy.

1(c)

[5 points] Prof. Pedantic, the esteemed Ineptitude Professor of Computer Science and Quackery at Wikipedia University, is developing a new terminal program (and associated service) to log into the servers in his lab. Although he is aware of ssh, he refuses to use it because he doesn't like being hushed. Instead, he decides to construct his own novel protocol. Like telnet and ssh, his remote console/terminal program should allow a remote user to type commands and execute them on a remote machine. Since Prof. Pedantic doesn't trust anyone - particularly the students in his introduction to network security class—he decides that all communication should be encrypted.
Prof. Pedantic decides to use the AES encryption algorithm in ECB mode. Is this a good choice? Give **two** reasons why or why not.

The issue with AES in ECB mode is that there is no initialisation vector or counter to alter the plain text in a non-deterministic way before the encryption.

As the encryption acts on each block, it will therefore produce the same output for every block with identical plaintext.

While this won't allow the key to be found, it does reveal information about the structure, such as knowing whenever the same 16 byte chunks are transmitted.

1(d)

[15 points] Prof. Pedantic designed a “secure” communication protocol for two parties (Alice and Bob) that have preshared secrets k_1 (the confidentiality key) and k_2 (the authenticity key).

Prof. Pedantic doesn’t believe in traditional MACs, so he constructs his protocol as follows: to send a message m, Alice (A) sends to Bob (B) the following:

$$\begin{aligned} A \to B : \langle \quad & r, \\ & iv_1, \\ & iv_2, \\ & S(k_1, iv_1) \oplus (m||r) \\ & S(k_2, iv_2) \oplus (m||r) \quad \rangle \end{aligned}$$

where || denotes concatenation, r is a nonce (to prevent replay attacks), iv_1 and iv_2 are fresh initialization vectors (IVs), and $S(k, iv)$ denotes a cryptographically secure pseudorandom sequence based on key k and IV iv (i.e., a stream cipher).

The professor claims that the protocol achieves confidentiality and authenticity, as defined as follows:

- *confidentiality*: an eavesdropper that observes a run of the protocol cannot learn the message m unless it knows the confidentiality key k_1; and
- *authenticity*: if Bob receives $\langle r, iv_1, iv_2, S(k_1, iv_1) \oplus (m||r), S(k_2, iv_2) \oplus (m||r)\rangle$ and r is a fresh nonce and the decryption of $S(k_1, iv_1) \oplus (m||r)$ equals the decryption of $S(k_2, iv_2) \oplus (m||r)$ (using the corresponding IVs and keys), then message m must have been transmitted by a party that knows both the confidentiality and authenticity keys (i.e., k_1 and k_2).

The professor’s intention is that Bob obtains m by decrypting $S(k_1, iv_1) \oplus (m||r)$ using key k_1 and iv_1. Further, Bob performs an authenticity check by ensuring that the decrypted message matches the decryption of $S(k_2, iv_2) \oplus (m||r)$ (via key k_2 and IV iv_2). He reasons that only a sender that knows both k_1 and k_2 can cause the decryptions to match.

Does Prof. Pedantic’s scheme achieve confidentiality and/or authenticity, as defined above? Briefly argue why or why not, for both confidentiality and authenticity. Assume that k_1 and k_2 are random 128-bit keys that have been securely shared apriori between Alice and Bob, that $k_1 \neq k_2$, and that the two IVs are also fresh.

Confidentiality - Yes - due to the different keys and changing ivs, nonce on message to prevent replay attacks and known plaintext attacks, it would not be possible to determine what the plaintext message was without one of the keys.

Authenticity - No - As the message is simply XORed with the stream cipher, it is possible for an attacker to copy an old message, and simply alter some of the bytes that correspond to m and optionally r (ensuring to alter all instances of each in the same way).

In this case, the IVs would need to stay unchanged so that after the recipent decryptes (XORing with their $S(k_x, IV_x)$ calculations) that the resulting communication would be in the form of $m_{altered}||r$, since the recipent will expect the nonce sent at the start of the communication to be present at the end of the message.
The decryption would still work as there is no verification of the message content (such as by using an HMAC), however the resulting plaintext would be changed to gibberish (or at least the attacker would not be able to transmit a particular message).

Despite the gibberish, as the nonce transmitted matches the nonce after decryption, the recipent can't determine if the gibberish was the sent message or if it had been altered.

Question 2: Eavesdropping on unencryptedim

allthepoints