Encryption Schemes and Definitions of Security

1. Suppose the plaintext and the key length is n. Hence, for any given plaintext m and ciphertext c of length n there is exactly one key k such that $\mathcal{E}_k(m) = c$. The key is given by $k = m + c \bmod 26$. Hence $Pr[\mathcal{E}_k(m) = c] = \frac{1}{26^n}$ for all m and c, if the key is chosen uniformly at random. The cipher is therefore perfectly secure.

2. Arguments: a) It is difficult to keep an algorithm secret.
   b) The algorithm should be available for investigation by a large community.
   c) The designers may not be aware of weaknesses.
   Counter-arguments: a) The algorithm is not easily available to attackers.
   b) Vulnerabilities are more likely to remain unknown.

3. Suppose a key k of length n is used twice to encrypt plaintexts of length $2n$, i.e., $k\|k$ is added to the plaintext. Let m_0 and m_1 be two different strings and c any string of length n. Then

$$Pr[\mathcal{E}_k(m_0\|m_0) = (c\|c)] = \frac{1}{2^n}, \text{ but } Pr[\mathcal{E}_k(m_0\|m_1) = (c\|c)] = 0.$$

   If the ciphertext is $c\|c$, then a plaintext $m_0\|m_1$ with $m_0 \neq m_1$ is impossible. This implies that the cipher does not have perfect secrecy.

4. Let c be a given ciphertext. Perfect secrecy implies that any plaintext is possible (i.e., with probability > 0) for the given ciphertext c. Hence for every $m_0 \in \mathcal{M}$, there must exist a key $k \in \mathcal{K}$ such that $\mathcal{D}_k(c) = m_0$. If the resulting ciphertext is fixed, then different plaintexts are encrypted using different keys. So there are at least as many keys as plaintexts.

5. A bit permutation is not perfectly secure. The number of zeros and ones is preserved by a bit permutation. Hence there exist many plaintext-ciphertext combinations (m, c) that never occur so $Pr[m|c] = 0$.

6. One has $Pr[b' \neq b] = 1 - Pr[b' = b]$. Then

$$\mathrm{Adv}(A) = \mid Pr[b' = b] - Pr[b' \neq b] \mid = \mid 2 \cdot Pr[b' = b] - 1 \mid .$$

   Since b is chosen uniformly at random, we have $Pr[b' = b] = Pr[b' = b \mid b = 1]$ and $Pr[b' \neq b] = Pr[b' \neq b \mid b = 0]$. This implies

$$\mathrm{Adv}(A) = \mid Pr[b' = 1|b = 1] - Pr[b' = 1|b = 0] \mid .$$

7. In the EAV experiment, the adversary chooses two plaintexts of the same length and obtains one ciphertext from the oracle. The adversary has to find out which plaintext was encrypted. The CPA experiment is similar, but gives the adversary more power: the adversary can freely choose any plaintext (even the chosen two plaintexts) and gets the ciphertext from the oracle. Therefore, a deterministic scheme can be EAV-secure but not CPA-secure.

8. Suppose the adversary A chooses m_0 and m_1 in the EAV experiment. If the scheme is perfectly secure, then the ciphertext $c = \mathcal{E}_k(m_b)$ does not provide any information about the plaintext. We have $Pr[m_0|c] = Pr[m_1|c]$ and hence $\mathrm{Adv}^{\mathrm{eav}}(A) = 0$ for any adversary. Perfect security is much stronger than EAV security: all plaintexts have the same probability if a ciphertext is given, which is not required for EAV security. A perfectly secure scheme cannot be broken

even with unlimited resources. EAV security 'only' requires that an adversary cannot successfully find the correct plaintext (from the two given candidates) in polynomial time.

9. Consider a Vigenère cipher has length n. The plaintext of a Vigenère cipher can be longer than the key. An adversary A chooses two plaintexts m_0, m_1 of length $2n$ such that $m_0 = x\|y$ and $m_1 = x\|y'$ where $y \neq y'$, and obtains the ciphertext $c = \mathcal{E}_k(m_b) = m_b \oplus (k\|k)$. The adversary only has to check whether $c + m_0$ or $c + m_1$ (modulo 26) is a string of type $k \parallel k$. This reveals the correct plaintext, so that the scheme is not EAV-secure.

10. The generator a) is not pseudorandom, since the output can be easily distinguished from a random sequence: binary addition of the $n + 2$ output bits gives zero. The generator G^+ in part b) is in fact pseudorandom if the original generator G is pseudorandom. Only the last bit is output of the new generator and the first n bits are used as seed for the next iteration. Each output bit and the generated seed bits are pseudorandom.

11. Let l be the output length. Select two different plaintexts of length l, say m_0 and m_1. The adversary chooses (m_0, m_1) as the first and (m_1, m_1) as the second plaintext pair. If the challenge ciphertexts are different, then the first plaintext of each pair was encrypted. Otherwise, both ciphertexts are $m_1 \oplus G(k)$ and the second plaintext was encrypted. Thus, the adversary can easily win the game, and the scheme cannot be EAV-secure for multiple encryptions.

12. Suppose there is a polynomial-time predictor for the $(i+1)^{\text{st}}$ bit given the first i bits. An adversary A in the prg experiment would use the predictor as a subroutine. The first i bits of the challenge string c are taken as input. The output of the predictor is compared with the $(i+1)^{\text{st}}$ bit of c. A outputs 1 if the bits are equal, and 0 otherwise. If the predictor has a non-negligible advantage, i.e., does not pass the next-bit test, then $\text{Adv}^{\text{prg}}(A)$ is non-negligible, i.e., the generator is not pseudorandom.

13. In a CCA2 attack against a malleable scheme, an adversary can ask for the decryption of a ciphertext c' that is related to the challenge ciphertext c. The adversary gets the corresponding plaintext m' with which they can infer the correct plaintext m_b, at least if the chosen plaintexts m_0 and m_1 are not related.

14. A family of bit permutations is not pseudorandom. The number of zeros and ones is preserved by a bit permutation, whereas a random permutation does not have this property. Therefore, an adversary can distinguish the output of a bit permutation from a random permutation. An adversary can choose m to be an all-zero string; if the oracle outputs a zero string c then almost certainly a bit permutation was used. If c is nonzero, then the output was not computed by a bit permutation.

15. Suppose the block length is l. The adversary chooses two binary strings x, y of length l with $x \neq y$ and sets $m_0 = x\|x$, $m_1 = x\|y$. If the oracle outputs a ciphertext of type $c\|c$, then the plaintext m_0 was encrypted. Otherwise, the oracle encrypted m_1. An adversary can thus easily win the EAV experiment. A block cipher in ECB mode is not EAV-secure.

16. No, since an adversary knows the initial IV (and all following IVs) in advance and can therefore predict the first ciphertext block of the n-th encryption under a chosen plaintext attack. Although different encryptions of the same plaintext give different ciphertexts, the scheme is deterministic because the IV is not randomized.

17. Suppose the plaintexts m_0 and m_1 are encrypted in CTR mode using the same counter. If the ciphertexts are c_0 and c_1, then $c_0 \oplus c_1 = m_0 \oplus m_1$. An adversary who eavesdrops two ciphertexts learns the XOR combination of the corresponding plaintexts. The scheme is completely broken with a simple ciphertext-only attack. In particular, the scheme is not EAV-secure for multiple encryptions.

18. It cannot be perfectly secure, since the key space is smaller than the message space (strings of arbitrary length). There are many plaintext-ciphertext pairs that never occur if the key is shorter than the message.

19. a) $c = 0111\ 0010\ 1000$
    b) $c = 1010\ 0010\ 0110\ 0100$
    c) $c = 1010\ 1100\ 1000\ 1111$
    In part b) and c), the IV or the CTR value is the first ciphertext block c_0.

20. a) IV is missing: m_1 cannot be computed, but all following blocks can be decrypted correctly.
    b) Transmission error in block c_k: m_k and m_{k+1} are faulty, but the following blocks are correct.
    c) Bit error in block c_k: m_k is faulty and m_{k+1} has a bit error. The following blocks are correct.
    d) Ciphering error in block c_k: the following ciphertext blocks are also faulty, m_k and all following plaintext blocks are incorrect.
    Note the different consequences of errors during transmission and ciphering. In the latter case, the faulty ciphertext block c_k corrupts the computation of all subsequent ciphertext blocks.

21. The plaintext blocks are m_1, m_2 and m_3, where the last block contains only 44 bits. The ciphertext blocks are $c_0 = ctr$, c_1, c_2, c_3.
    a) First counter bit is flipped: all decrypted plaintext blocks are faulty since all counter values are incorrect.
    b) Bit error in the first bit of c_1: only the first plaintext bit is faulty and bits $2 - 300$ are correct.
    c) Error in block c_1: the block m_1 (bits $1 - 128$) is faulty and the plaintext bits $129 - 300$ are correct.
    d) Bit error in the last bit of c_3: the plaintext bits $1 - 299$ are correct and the last bit is faulty.

22.

	ECB	CBC	CTR
Message expansion	no	extra IV	extra counter
Error propagation	no	yes	no
Pre-computation	no	no	yes
Parallelization	yes	only decryption	yes

23. We use the fact that CBC and CTR modes are malleable. In the CCA2 experiment, an adversary chooses any two unrelated plaintexts consisting each of one block. Suppose the oracle selects m and the challenge ciphertext is $c = c_0 \| c_1$, where c_0 is the initialization vector (CBC mode) or the counter (CTR mode).

    First, consider a cipher in CBC mode. Let $e_1 = 10 \ldots 0$ be zero except for the first bit. The adversary chooses the ciphertext $c' = (c_0 \oplus e_1) \parallel c_1$. Since $c \neq c'$, the oracle accepts c' in a chosen-ciphertext attack. The oracle computes

    $$m' = E_k^{-1}(c_1) \oplus c_0 \oplus e_1$$

    and gives m' to the adversary. But m' is equal to $m \oplus e_1$, so that the adversary can easily select the correct plaintext and win the CCA2 game.

    Now, consider the CTR mode. The adversary chooses $c' = c_0 \parallel (c_1 \oplus e_1)$, where $e_1 = 10 \ldots 0$ is as above. The oracle computes

    $$m' = E_k(c_0 + 1) \oplus c_1 \oplus e_1.$$

    The adversary gets the plaintext $m' = m \oplus e_1$ and can thus easily win the CCA2 game.