Question 1. (25/100)

Task 1 (6 Marks):

(1) $10001101 \land 01110110 = 00000100$
(2) $10111001 \lor 01100111 = 11111111$
(3) $11100011 \oplus 01110111 = 10010100$

Task 2 (6 Marks):

(1) Confidentiality: The personal information should not be leaked to the unauthorized entities.
(2) Integrity: The personal information should be able to be modified by only the user himself/herself.
(3) Availability: The authorized entities should be able to access the personal information when they need it.

(or other reasonable answers)

Task 3 (7 Marks):

Use an input with more than 16 bytes, which is with the length of 24 bytes.

The first 16 bytes can be any data. Then the following 4 bytes can also be any data. The last 4 bytes are the address of the function `special_action()`, which is 0x000025C2.

Thus, the overall input is [any data with 20 bytes]+[0x000025C2].

The overall input will override the `retaddr` in the memory. When the computer finishes the execution of `authorize()`, it will go to the address `retaddr` for the code execution. Since we override the address by 0x000025C2, the computer will execute the code in `special_action()` directly without checking the password.

Task 4 (6 Marks):

The user trust CA 2, so it only trusts the derivatives of CA 2, which are Endpoint 1. It does not trust other entities in this figure.

Thus, this user does not trust Endpoint 2, but this user trusts Endpoint 1.

Question 2. (25/100)

Task 1 (4 Marks):

Authentication is to determine whether the user is who he/she claims to be. (i.e., who the user is)

Authorization is to check whether the user has the right to do something. (i.e., what the user can do)

Authentication is the previous step of Authorization.

Task 2 (6 Marks):

If Eve is a passive attacker, it can only read the message transmitted from Alice to Bob.

If Eve is an active attacker, it can perform more active attacks besides reading.

Examples:

In the passive scenario, Eve can read the message, which breaks confidentiality.

In the active scenario, Eve can modify the message, which breaks integrity.

(or other reasonable answers)

Task 3 (10 Marks):

Convert the plaintext string to its numerical representation, perform the encryption $Enc(7, m) = (m + 7) \ mod \ 26$ on each character, and convert the ciphertext numbers back to their characters respectively.

1. P is 15.
   a. $c_1 = (15 + 7) \ mod \ 26 = 22.$
   b. The corresponding character is W.
2. O is 14.
   a. $c_2 = (14 + 7) \ mod \ 26 = 21.$
   b. The corresponding character is V.
3. L is 11.
   a. $c_3 = (11 + 7) \ mod \ 26 = 18.$
   b. The corresponding character is S.
4. Y is 24.
   a. $c_4 = (24 + 7) \ mod \ 26 = 5.$
   b. The corresponding character is F.
5. U is 20.

    a. $c_5 = (20 + 7)\ mod\ 26 = 1$.
    b. The corresponding character is *B*.
6. *C* is 2.
    a. $c_6 = (2 + 7)\ mod\ 26 = 9$.
    b. The corresponding character is *J*.
7. *O* is 14.
    a. $c_7 = (14 + 7)\ mod\ 26 = 21$.
    b. The corresponding character is *V*.
8. *M* is 12.
    a. $c_8 = (12 + 7)\ mod\ 26 = 19$.
    b. The corresponding character is *T*.
9. *P* is 15.
    a. $c_9 = (15 + 7)\ mod\ 26 = 22$.
    b. The corresponding character is *W*.

Thus, the ciphertext is WVSFBJVTW.

Task 4 (5 Marks):

A truly random number is generated by the hardware, which is from physical activities.

A pseudorandom number is generated by the pseudorandom functions, which is calculated from a seed.

Question 3. (25/100)

In the last round,

ByteSub:

The result calculated from the S-box is:

$$\begin{bmatrix} 67 & af & 4c & 71 \\ a2 & 84 & 98 & c6 \\ d3 & 6d & 65 & 66 \\ 74 & 95 & 35 & 25 \end{bmatrix}$$

ShiftRows:

The result of ShiftRows is:

$$\begin{bmatrix} 67 & af & 4c & 71 \\ 84 & 98 & c6 & a2 \\ 65 & 66 & d3 & 6d \\ 25 & 74 & 95 & 35 \end{bmatrix}$$

AddRoundKey:

The result of *AddRoundKey* is:

$$\begin{bmatrix} 67 & af & 4c & 71 \\ 84 & 98 & c6 & a2 \\ 65 & 66 & d3 & 6d \\ 25 & 74 & 95 & 35 \end{bmatrix} \oplus \begin{bmatrix} b6 & 8f & 48 & 23 \\ fd & 9f & 0a & f6 \\ cf & 78 & 6b & 12 \\ 6b & 96 & 23 & e9 \end{bmatrix} = \begin{bmatrix} d1 & 20 & 04 & 52 \\ 79 & 07 & cc & 54 \\ aa & 1e & b8 & 7f \\ 4e & e2 & b6 & dc \end{bmatrix}$$

The result of the last round is:

$$\begin{bmatrix} d1 & 20 & 04 & 52 \\ 79 & 07 & cc & 54 \\ aa & 1e & b8 & 7f \\ 4e & e2 & b6 & dc \end{bmatrix}$$

Question 4. (25/100)

Task 1 (5 Marks):

The number of trials: $\sqrt{2^{200}} = 2^{100}$. Since $2^{100} < 2^{112}$ which is the security requirement from NIST, this function does not ensure the strong collision resistance.

Task 2 (5 Marks):

The collision exists in the hash function, because the size of inputs is much bigger than the size of outputs. Since one input must be mapped to an output, it must be multiple inputs whose outputs are the same. Thus, collision happens.

Task 3 (4 Marks):

MD5 is deprecated because its strong collision resistance has been broken.

Task 4 (6 Marks):

Given a message with the length of 29 bytes, we need to pad 3 bytes to let it be divisible if the block size is 4 bytes.

According to ZeroLength padding, it pads with zeros except for the last byte which is equal to the number of padding bytes. Thus, the padded data is 0x00 0x00 0x03.

Task 5 (5 Marks):

$MAC = SHA3(k||m)$ or $MAC = SHA3(m||k)$.