# Laboratory work 7

Dávid Kromka

## Task 1

**Input:**

* **name**: A string representing a message (e.g., "David").

**Output:**

* **encrypted\_message**: A list of hexadecimal values obtained after applying the SubBytes transformation using the Rijndael S-box.

**Procedure:**

1. Initialize an empty list **ascii\_name**.
2. Convert the characters in the **name** string to their corresponding ASCII values and store them in the **ascii\_name** list. For each character **char** in **name**:
   * Compute the ASCII value of **char** using the **ord(char)** function.
   * Append the ASCII value to the **ascii\_name** list.
3. Initialize an empty list **hex\_name**.
4. Convert each ASCII value in the **ascii\_name** list to its hexadecimal representation and store the hexadecimal values in the **hex\_name** list. For each ASCII value **val** in **ascii\_name**:
   * Convert **val** to its hexadecimal representation, omitting the "0x" prefix, and append the result to the **hex\_name** list.
5. Initialize an empty list **encrypted\_message**.
6. Perform SubBytes conversion using the Rijndael S-box:
   * For each hexadecimal byte **byte** in the **hex\_name** list:
     + Extract the first and second hexadecimal digits of **byte** and convert them to integers.
     + Use the obtained integers as indices to access the Rijndael S-box and obtain the corresponding substituted byte value.
     + Append the substituted byte value to the **encrypted\_message** list.
7. The **encrypted\_message** list now contains the result of applying the SubBytes transformation to the input message, as specified by the Rijndael S-box.

**Output:**

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

## Task 2

**Input:**

* **entries**: A list of hexadecimal strings.

**Output:**

* **decimal\_numbers**: A list of decimal integers obtained after applying the SubBytes transformation using the Rijndael S-box.
* **hex\_result**: A list of hexadecimal values obtained after applying the SubBytes transformation using the Rijndael S-box.

**Procedure:**

1. **Direct SubBytes Conversion:**
   * Initialize an empty list **hex\_result**.
   * Perform SubBytes conversion using the Rijndael S-box for each hexadecimal byte **byte** in the **entries** list:
     + Extract the first and second hexadecimal digits of **byte** and convert them to integers.
     + Use the obtained integers as indices to access the Rijndael S-box and obtain the corresponding substituted byte value.
     + Append the substituted byte value to the **hex\_result** list.
2. Convert each byte in the **hex\_result** list to its decimal representation and store the decimal values in the **decimal\_numbers** list. For each byte **byte** in **hex\_result**:
   * Convert **byte** to its decimal representation using the **int(byte, 16)** function.
   * Append the decimal value to the **decimal\_numbers** list.
3. Print the results of the direct SubBytes conversion:
   * Original entries in hexadecimal representation (**entries**).
   * Resulting entries in decimal representation (**decimal\_numbers**).
4. **Reverse SubBytes Conversion:**
   * Initialize an empty list **hex\_result**.
   * Perform SubBytes conversion using the reverse Rijndael S-box for each hexadecimal byte **byte** in the **entries** list:
     + Extract the first and second hexadecimal digits of **byte** and convert them to integers.
     + Use the obtained integers as indices to access the reverse Rijndael S-box and obtain the corresponding substituted byte value.
     + Append the substituted byte value to the **hex\_result** list.
5. Convert each byte in the **hex\_result** list to its decimal representation and store the decimal values in the **decimal\_numbers** list. For each byte **byte** in **hex\_result**:
   * Convert **byte** to its decimal representation using the **int(byte, 16)** function.
   * Append the decimal value to the **decimal\_numbers** list.
6. Print the results of the reverse SubBytes conversion:
   * Original entries in hexadecimal representation (**entries**).
   * Resulting entries in decimal representation (**decimal\_numbers**).
7. The **decimal\_numbers** list now contains the result of applying the SubBytes transformation using both the forward and reverse Rijndael S-boxes.

**Output:**

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

## Task 3

**Input:**

* Two integers **a** and **b** to be multiplied in the Galois field.

**Output:**

* **result**: The result of the multiplication operation in the Galois field, calculated as **(a \* b) % p**, where **p** is the prime number defining the Galois field.

**Procedure:**

1. Initialize the prime number for the Galois field:
   * **p = 11**.
2. Perform the multiplication operation in the Galois field:
   * Multiply the integers **a** and **b**.
   * Calculate the result as **(a \* b) % p**.
3. The **result** variable now holds the result of the multiplication operation in the Galois field.

**Output:**

****

## Task 4

**Input:**

* Coordinates of the original triangle vertices: A(0, 0), B(0, 100), C(200, 0)
* Transformation matrix **M = [[2, 0, 0], [0, 2, 0], [0, 0, 0]]**
* Translation vector **V = [0, -100, 0]**

**Output:**

* Transformed coordinates of the triangle vertices after applying the affine transformation.

**Procedure:**

1. **Define the Original Coordinates:**
   * Define the original coordinates of the triangle vertices:
     + A(0, 0, 1)
     + B(0, 100, 1)
     + C(200, 0, 1)
2. **Define Transformation Matrix and Translation Vector:**
   * Define the transformation matrix **M** as **[[2, 0, 0], [0, 2, 0], [0, 0, 0]]**.
   * Define the translation vector **V** as **[0, -100, 0]**.
3. **Apply Affine Transformation to Each Point:**
   * Calculate the transformed coordinates of each vertex using the formula:
     + For each point P, calculate **P\_transformed = M \* P + V**.
     + For point A:
       - **A\_transformed = M \* [0, 0, 1] + [0, -100, 0]**.
     + For point B:
       - **B\_transformed = M \* [0, 100, 1] + [0, -100, 0]**.
     + For point C:
       - **C\_transformed = M \* [200, 0, 1] + [0, -100, 0]**.
4. **Output:**
   * Print the transformed coordinates of points A, B, and C.

**Output:**

Obrázok, na ktorom je text, písmo, snímka obrazovky

Automaticky generovaný popis