Known Plaintext Attack on AES-128

In this assignment, we work with the AES-128 algorithm, which we will use from a library. For Python, the recommended library is cryptography. For C/C++, libgcrypt or openssl. Sample programs for each are included in the files. You may also use any publicly available AES-128 implementation instead.

1. In the first part, the goal is to find a 20-bit key by a brute-force attack. Since AES uses a 128-bit key, a 24-bit key is fed into an expansion subroutine that produces a 128-bit long key. The subroutine ignores the last 4-bits of the short key so that the effective key length is 20. The key expansion subroutine is given to you as C and Python files.

   $$K_{short} \in \{0,1\}^{24}$$
   $$K_{AES} = expandKey(K_{short}) \in \{0,\ 1\}^{128}$$
   $$C = AES_{K_{AES}}(P),\ where\ 'P'\ is\ the\ plaintext\ and\ 'C'\ is\ the\ ciphertext.$$

   A set of four plaintexts and five ciphertexts are given to you as two text files, with the ciphertexts corresponding to the four given plaintexts and one secret plaintext. Use one of the plaintext-ciphertext pairs to find the key and hence find the secret plaintext as well.

2. In the second part, we do a meet-in-the-middle attack. Here the encryption is done using two 16-bit keys. Firstly, each key is expanded via a key expansion routine (given to you) to obtain two long (128-bit) keys. Then the message is encrypted with the first long key, and then again with the second long key, to obtain the ciphertext. Each AES operation is treated as a black box, but the two-key AES structure as a whole is not a black box, which means you can get the intermediate encryption and hence it allows for the meet-in-the-middle approach. A set of four plaintexts and five ciphertexts is given as input and the goal is the same as in part 1.

$$K_{short-1} \in \{0,1\}^{16}$$

$$K_{short-2} \in \{0,1\}^{16}$$

$$K_{AES-1} = expandKey(K_{short-1}) \in \{0,\ 1\}^{128}$$

$$K_{AES-2} = expandKey(K_{short-2}) \in \{0,\ 1\}^{128}$$

$$C = AES_{K_{AES-2}}(AES_{K_{AES-1}}(P)),$$

$$where\ 'P'\ is\ the\ plaintext\ and\ 'C'\ is\ the\ ciphertext$$

Note: The initialization vector (IV) used in both AES and 2-Key AES is a zero vector.

Deliverables:

1) Your program file should be named as (Prog_Asgn_3_1_<Roll_No>.cpp and Prog_Asgn_3_2_<Roll_No>.cpp) or (Prog_Asgn_3_1_<Roll_No>.py and Prog_Asgn_3_2_<Roll_No>.py)
2) ReadMe.
3) Combine your program files and ReadMe into a zip archive file and name it as Prog_Asgn_3_<Roll_No>.zip. Upload this zip file.