Homework Number: HW04

Name: Andrew Wu

ECN Login: wu1795

Due Date: 2/13/24

Encrypted Output:

3ba1ab4b7fe412ca26c7a25cff913d1b748da805c97c83554d9e9cf5b12243ff03a8c6b6dcb
c520750a14df9b646fa480d1e64cc2e9174a23dbed6aad77144350ff768093cf7571852a26ffa36fe4
7652a546acf9d4bc1ad395a92553b4b7e0a5a7811d7b95d95cacc117e344ac093da247168cd4bbbd
a5bc2866fd044c8ca18ecd2b6a78bfe19520f22b7fa12862132e32ee78c5e4200166c40f1a93f9b08c
5f67b9bde38d34ed34bd03183a529a5a62d81b1cf084832fcb9139a51100a04c7c631d3fbfa5bb9b8
cbe970f02213ab07d3e179313142865fb8b022241552567964250cfa2aa97c59223d30a2a7da8974
d0f6c34f4f46ed6cab53e483f95d4ed157bb78ce078a88397c9d656830fadd080d729ac7428a6ca3c
17ad67d0cf16d35a8ecb35cd818a380309332c4cc29d00b6fe542b67724295b49804b2122b5b24e6
f09e22451bb77c6876d51b7294b405dcff0cdc83754538442fcc766bfe4fac839e932f757aebbe7f43
c87d08249c6ef50d9adefa8eca175785ba0dbc31e2e61ba32a75f596894ea736bcea8f351d3c45745
39e7ad760c4a0c4b252e2dbc859c4b0a6b44fbf29b3fa7fddeace3855c675130ef65d4fa7f8125d457
5f329cc93d75d14fdcb1419678cae4d686d4b72f56ac4d7974e3b1f1bbb3776dda5db94b7d2ef1f73
f96f7b24378a1e299271006cd478bd84fe7a24c67794e663668c918bdb65097099351e1ebf6e7d11
48754f1051d33156e4fb7e96cce8f976f6a0ad71d12b10d1b43458c02002bf1fc14c9c63e9033dfdc
bc9baae76efc8e12a850fdd21ead4e9b14fb359a27fc4943b0d76714

Decrypted Output:

Newly re-signed McLaren driver Lando Norris is confident that the team will be in the mix for race victories in 2024, but the Briton feels he may have to wait a little longer for a championship challenge. McLaren caught the eye last season by going from struggling to score points to regularly fighting for podiums, with highly effective upgrades being implemented following a technical reshuffle. Norris came close to scoring McLaren's first Grand Prix win since 2021 on several occasions, taking six P2 finishes, while team mate Oscar Piastri managed to triumph in the Qatar Sprint Race.

In my AES.py file, I have split the file based on the functions being used for encryption or decryption, with the adapted lecture code at the bottom. For encryption, I read the input by blocks of 128 bits and pad them when necessary. I then slice the blocks to fill the initial state array, and XORing the initial state array with the first four words of the key schedule, derived from using the gen_key_schedule_256 function. I then use a for loop to push the state array through the functions subbytes, shiftrows, mixcolumns, and addroundkey, omitting the mixcolumns function on the 14^{th}/last round. For the subbytes function, I push each element of statearray through the gen_subbytes_table function that finds the multiplicative inverse of each

element, XORs the element with 4 shifted versions of itself, and the special constant. For the shiftrows function, I shift the second, third, and fourth rows by one, two, and three bytes to the left, respectively. For the mixcolumns function, I multiply each element in statearray by 2, XOR them with 3 times the next element and the remaining two elements columnwise. For the addroundkey function. I slice the corresponding words from the key schedule and XOR them with statearray. I then flatten the resulting statearray into a singular bitvector and write the output in hex. For decryption, I open, read, and create blocks of one byte from the input encrypted file, and create the same key schedule as encryption. I then slice the input to get elements to fill the initial state array, just like in encryption, and conduct the initial XOR with the last four words of the key schedule. I then do the reverse round order conducted in encryption, pushing the state array through invshiftrows, invsubbytes, addroundkey, and invmixcolumns, with the last round (round 1) omitting the invmixcolumns functions. The addroundkey function is the same as in encryption, with the invshiftrows shifting right instead of left, invsubbytes doing the reverse calculation as subbytes, and invmixcolumns doing the reverse mixing as mixcolumns. I then flatten the corresponding state array and write the output in ASCII.