This implementation provides a complete AES encryption and decryption system using object-oriented Python. The code processes data in 128-bit blocks using a state array structure, where each block undergoes 14 rounds of transformation. Each round applies SubBytes using lookup tables generated from GF(2^8) operations, performs ShiftRows by circular rotation, executes MixColumns through matrix multiplication in GF(2^8), and adds the round key. The final round omits the MixColumns step as specified.

The implementation successfully processes the input by first converting it to BitVectors for efficient finite field operations. The encryption output is saved as a hex string, and when decrypted, correctly recovers the original text. As expected, the decrypted output contains trailing NULL bytes for padding the last block to the required 128-bit size, which is verified by the fact that the only difference between the original and decrypted files is the specified padding bytes at the end