Problem 2 Task 3

a) What is the key space for the mini block cipher?
Answer: The mini block cipher uses a key size of 16 bits. The key space is the set of all possible keys that can be generated from those 16 bits. For a binary system, where each bit can be either 0 or 1, the size of the key space (number of possible keys) is 2^{16}, because each of the 16 positions can have 2 possible values.

b) Image the mini block cipher is executed twice to generate a cipher text. It is called double mini cipher block. We need a key in 32 bits. The first 16 to the first mini block cipher, the remaining 16 to the second mini block cipher. The meet in the middle attack is to match the state for the first encryption of mini block cipher and the second decryption mini block. How many operations are needed to such attack?
Answer:
For the meet-in-the-middle attack, you would:

1. Encrypt the plaintext with all 2^{16} possible keys and store the intermediate results.
2. Decrypt the ciphertext with all 2^{16} possible keys and also store these intermediate results.

Then we can search for matching intermediate values between these two sets. The computational effort for this is primarily in the encryption and decryption steps, each involving 2^{16} operations, leading to a total of 2×2^{16} = 131,0722 × 216 = 131,072 operations.

c) If we do exhaustive key search for the double mini block cipher, how many operations are needed?
Answer: An exhaustive key search for a double mini block cipher, where you have a 32-bit key split into two 16-bit keys, would involve trying every possible combination of both keys. Since each mini block cipher operation uses a 16-bit key, and you are combining two such operations, you would have to try 2^{32} key combinations to guarantee finding the correct key. This amounts to 2^{32} = 4,294,967,296232=4,294,967,296 operations, significantly more than the meet-in-the-middle attack.

d) What is the trade off in this attack?

Answer: The main trade-off in employing a meet-in-the-middle attack lies in balancing time (computational complexity) against memory (space complexity). For time complexity, The meet-in-the-middle attack significantly reduces the time complexity from 2^{32} (for an exhaustive search of a 32-bit key) to about 2^{17} operations when considering both encryption and decryption phases separately. However, this time efficiency comes at the cost of increased memory usage. You need to store 2^{16} intermediate states from both encryption and decryption phases to facilitate the comparison for matches. This can be substantial, depending on the storage requirements for each state.