Exhaustive key search algorithm:

The basic strategy of attempting every possible key in turn until the correct key is determined is known as exhaustive key search, or brute-force search. It may be essential to have both plaintext and its corresponding ciphertext to identify the proper key, or ciphertext alone may be enough if the plaintext has some recognizable characteristic. Exhaustive key search can be used on any cipher, and a flaw in the cipher's key schedule can occasionally aid the efficiency of an exhaustive key search attack.

If the length of the key is known it makes it easy for an attacker to crack the key.

In the given challenge, I have used the brute force approach to identify the key used to encrypt the message.

Steps:

1. First, read the message, ciphertext, and nonce from the files.
2. Pass these three values as arguments to the function key_finder.
3. Consider the keyspace of length 2^24 (given 24 bits) and iterate through the loop to generate the keys with first 13 bytes fixed followed by the bytes generated within the keyspace the time complexity is of O(n) where n is the keyspace length 2^24.
4. Use the AES.new to generate the ciphertext with the generated key and nonce.
5. And decrypt the ciphertext and check with the original message bytes
6. If the decrypted ciphertext matches the original message bytes then return the key.
7. Iterate the process for all the messages and ciphertext