Exercise 6

Question 1

Let m be a message consisting of ℓ AES blocks (say $\ell = 100$). Alice encrypts m using CBC mode and transmits the resulting ciphertext to Bob. Due to a network error, ciphertext block number $\ell/2$ is corrupted during transmission. All other ciphertext blocks are transmitted and received correctly. Once Bob decrypts the received ciphertext, how many plaintext blocks will be corrupted?

Question 2

Let m be a message consisting of ℓ AES blocks (say $\ell = 100$). Alice encrypts m using randomized counter mode and transmits the resulting ciphertext to Bob. Due to a network error, ciphertext block number $\ell/2$ is corrupted during transmission. All other ciphertext blocks are transmitted and received correctly. Once Bob decrypts the received ciphertext, how many plaintext blocks will be corrupted?

Question 3

Recall that encryption systems do not fully hide the length of transmitted messages. Leaking the length of web requests has been used to eavesdrop on encrypted HTTPS traffic to a number of web sites, such as tax preparation sites, Google searches, and healthcare sites. Suppose an attacker intercepts a packet where he knows that the packet payload is encrypted using AES in CBC mode with a random IV. The encrypted packet payload is 128 bytes. Which of the following messages is plausibly the decryption of the payload:

1. `The most direct computation would be for the enemy to try all 2^r possible keys, one by one.`
2. `If qualified opinions incline to believe in the exponential conjecture, then I think we cannot afford not to make use of it.`
3. `We see immediately that one needs little information to begin to break down the process.`
4. `In this letter I make some remarks on a general principle relevant to enciphering in general and my machine.`

(*a*) *NybbleCrypt* is a block cipher optimized for use in exam questions. It has a block size of 4 bits and a key length of 64 bits. Each block can be written as a single hexadecimal digit, for example $\texttt{5} \oplus \texttt{9} = \texttt{c}$.

(*i*) The *NybbleCrypt* encryption function for a particular key K is given in the following table:

m	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$E_K(m)$	c	8	2	7	d	0	6	1	a	e	f	4	b	9	5	3

Decrypt the following messages, which were encrypted using E_K under the following modes of operation, respectively:

(A) ECB mode: `c994f88` [2 marks]

(B) CBC mode: `b144f` [3 marks]

(C) OFB mode: `eae26` [3 marks]

Problems

5.1. Consider the storage of data in encrypted form in a large database using AES. One record has a size of 16 bytes. Assume that the records are not related to one another. Which mode would be best suited and why?

5.2. We consider known-plaintext attacks on block ciphers by means of an exhaustive key search where the key is k bits long. The block length counts n bits with $n > k$.

1. How many plaintexts and ciphertexts are needed to successfully break a block cipher running in ECB mode? How many steps are done in the worst case?
2. Assume that the initialization vector IV for running the considered block cipher in CBC mode is known. How many plaintexts and ciphertexts are now needed to break the cipher by performing an exhaustive key search? How many steps need now maximally be done? Briefly describe the attack.
3. How many plaintexts and ciphertexts are necessary, if you do *not* know the IV?
4. Is breaking a block cipher in CBC mode by means of an exhaustive key search considerably more difficult than breaking an ECB mode block cipher?

5.3. In a company, all files which are sent on the network are automatically encrypted by using AES-128 in CBC mode. A fixed key is used, and the IV is changed once per day. The network encryption is file-based, so that the IV is used at the beginning of every file.

You managed to spy out the fixed AES-128 key, but do not know the recent IV. Today, you were able to eavesdrop two different files, one with unidentified content and one which is known to be an automatically generated temporary file and only contains the value `0xFF`. Briefly describe how it is possible to obtain the unknown initialization vector and how you are able to determine the content of the unknown file.

5.4. Keeping the IV secret in OFB mode does not make an exhaustive key search more complex. Describe how we can perform a brute-force attack with unknown IV. What are the requirements regarding plaintext and ciphertext?

5.5. Describe how the OFB mode can be attacked if the IV is *not* different for each execution of the encryption operation.

5.6. Propose an OFB mode scheme which encrypts one byte of plaintext at a time, e.g., for encrypting key strokes from a remote keyboard. The block cipher used is AES. Perform one block cipher operation for every new plaintext byte. Draw a block diagram of your scheme and pay particular attention to the bit lengths used in your diagram (cf. the descripton of a byte mode at the end of Sect. 5.1.4).

5.7. As is so often true in cryptography, it is easy to weaken a seemingly strong scheme by small modifications. Assume a variant of the OFB mode by which we only feed back the 8 most significant bits of the cipher output. We use AES and fill the remaining 120 input bits to the cipher with 0s.

1. Draw a block diagram of the scheme.
2. Why is this scheme weak if we encrypt moderately large blocks of plaintext, say 100 kByte? What is the maximum number of known plaintexts an attacker needs to completely break the scheme?
3. Let the feedback byte be denoted by FB. Does the scheme become cryptographically stronger if we feedback the 128-bit value $FB, FB, \ldots, FB$ to the input (i.e., we copy the feedback byte 16 times and use it as AES input)?

5.8. In the text, a variant of the CFB mode is proposed which encrypts individual bytes. Draw a block diagram for this mode when using AES as block cipher. Indicate the width (in bit) of each line in your diagram.

5.9. We are using AES in counter mode for encrypting a hard disk with 1 TB of capacity. What is the maximum length of the IV?

5.10. Sometimes error propagation is an issue when choosing a mode of operation in practice. In order to analyze the propagation of errors, let us assume a bit error (i.e., a substitution of a "0" bit by a "1" bit or vice versa) in a ciphertext block y_i.

1. Assume an error occurs during the transmission in one block of ciphertext, let's say y_i. Which cleartext blocks are affected on Bob's side when using the ECB mode?
2. Again, assume block y_i contains an error introduced during transmission. Which cleartext blocks are affected on Bob's side when using the CBC mode?
3. Suppose there is an error in the cleartext x_i on Alice's side. Which cleartext blocks are affected on Bob's side when using the CBC mode?
4. Assume a single bit error occurs in the transmission of a ciphertext character in 8-bit CFB mode. How far does the error propagate? Describe exactly *how* each block is affected.
5. Prepare an overview of the effect of bit errors in a ciphertext block for the modes ECB, CBC, CFB, OFB and CTR. Differentiate between random bit errors and specific bit errors when decrypting y_i.

5.11. Besides simple bit errors, the deletion or insertion of a bit yields even more severe effects since the synchronization of blocks is disrupted. In most cases, the decryption of subsequent blocks will be incorrect. A special case is the CFB mode with a feedback width of 1 bit. Show that the synchronization is automatically restored after $\kappa + 1$ steps, where κ is the block size of the block cipher.

5.12. We now analyze the security of DES double encryption (2DES) by doing a cost-estimate:

$$2DES(x) = DES_{K_2}(DES_{K_1}(x))$$

1. First, let us assume a pure key search without any memory usage. For this purpose, the whole key space spanned by K_1 and K_2 has to be searched. How much does a key-search machine for breaking 2DES (worst case) in 1 week cost?
   In this case, assume ASICs which can perform 10^7 keys per second at a cost of \$5 per IC. Furthermore, assume an overhead of 50% for building the key search machine.

2. Let us now consider the meet-in-the-middle (or time-memory tradeoff) attack, in which we can use memory. Answer the following questions:
   - How many entries have to be stored?
   - How many bytes (not bits!) have to be stored for each entry?
   - How costly is a key search in one week? Please note that the key space has to be searched before filling up the memory completely. Then we can begin to search the key space of the second key. Assume the same hardware for both key spaces.

   For a rough cost estimate, assume the following costs for hard disk space: \$8/10 GByte, where 1 GByte = 10^9 Byte.
3. Assuming Moore's Law, when do the costs move below \$1 million?

5.13. Imagine that aliens — rather than abducting earthlings and performing strange experiments on them — drop a computer on planet Earth that is particularly suited for AES key searches. In fact, it is so powerful that we can search through 128, 192 and 256 key bits in a matter of days. Provide guidelines for the number of plaintext–ciphertext pairs the aliens need so that they can rule out false keys with a reasonable likelihood. (**Remark**: Since the existence of both aliens and human-built computers for such key lengths seem extremely unlikely at the time of writing, this problem is pure science fiction.)

5.14. Given multiple plaintext–ciphertext pairs, your objective is to attack an encryption scheme based upon multiple encryptions.

1. You want to break an encryption system E, which makes use of triple AES-192 encryption (e.g. block length $n = 128$ bit, key size of $k = 192$ bit). How many tuples (x_i, y_i) with $y_i = e_K(x_i)$ do you need to level down the probability of finding a key K, which matches the condition $y_i = e_K(x_i)$ for one particular i, but fails for most other values of i (a so called *false positive*), to $Pr(K' \neq K) = 2^{-20}$?
2. What is the maximum key size of a block cipher that you could still effectively attack with an error probability of at most $Pr(K' \neq K) = 2^{-10} = 1/1024$, if this cipher always uses double encryption ($l = 2$) and has a block length of $n = 80$ bit?
3. Estimate the success probability, if you are provided with four plaintext–ciphertext blocks which are double encrypted using AES-256 ($n = 128$ bits, $k = 256$ bits). Please justify your results.

Note that this is a purely theoretical problem. Key spaces of size 2^{128} and beyond can not be brute-forced.

5.15. 3DES with three different keys can be broken with about 2^{2k} encryptions and 2^k memory cells, $k = 56$. Design the corresponding attack. How many pairs (x, y) should be available so that the probability to determine an incorrect key triple (k_1, k_2, k_3) is sufficiently low?

5.16. This is your chance to break a cryptosystem. As we know by now, cryptography is a tricky business. The following problem illustrates how easy it is to turn a strong scheme into a weak one with minor modifications.

We saw in this chapter that key whitening is a good technique for strengthening block ciphers against brute-force attacks. We now look at the following variant of key whitening against DES, which we'll call DESA:

$$DESA_{k,k_1}(x) = DES_k(x) \oplus k_1.$$

Even though the method looks similar to key whitening, it hardly adds to the security. Your task is to show that breaking the scheme is roughly as difficult as a brute-force attack against single DES. Assume you have a few pairs of plaintext–ciphertext.