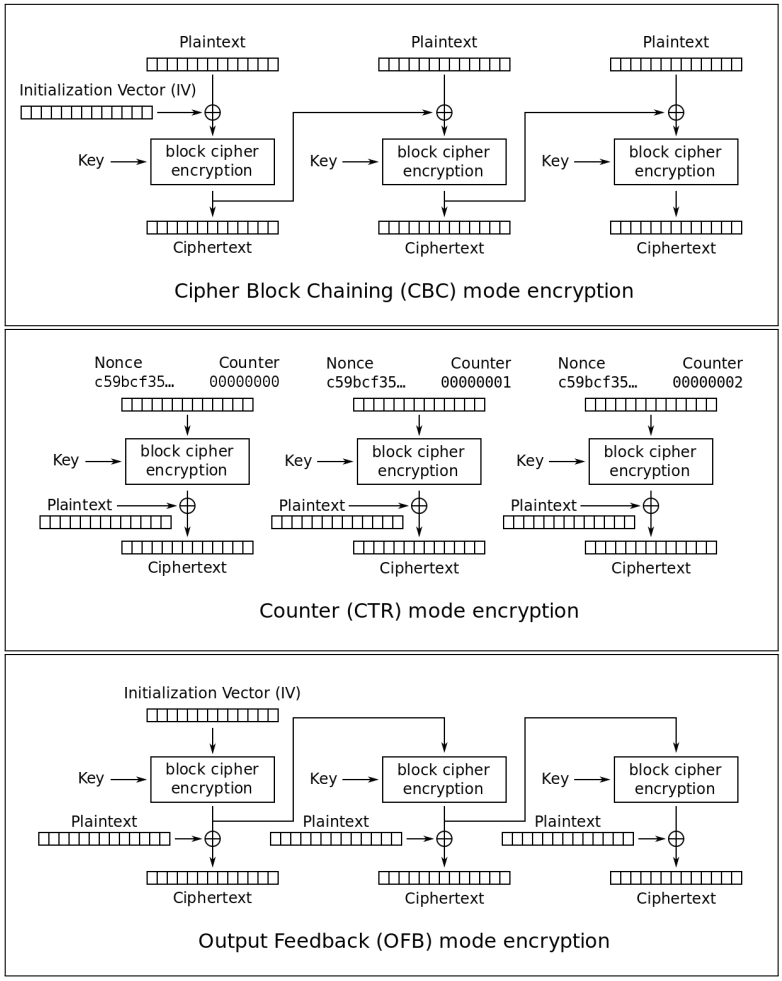
1 Decryption

Consider these handy diagrams from Wikipedia:



(CTR mode is a bit different from class because it considers the random part of the IV/nonce separate from the counter. Both of them are fairly similar, so feel free to use either version.)

1.1

Draw the world's prettiest decryption diagrams for CTR, CBC, and CFB.

2

Let m be a message consisting of k blocks (say $k = 100$). Alice encrypts m using CBC mode and transmits the resulting ciphertext c to Bob. Due to a hardware error, ciphertext block number $k/2$ 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?
- Answer the same question for randomized counter mode.
- Answer the same question for OFB mode.

(Don't just give a careless answer. Use your diagrams, and make sure can write a correct, written justification for your answer

3

Try to come up with your own block cipher.

Either break your system by showing an attack on semantic security, or try to show that it is secure if the PRF is secure.

(We haven't exactly defined how to do that, but you would start by showing that every input into a PRF is unique. That is, you never compute $E(k, a)$ for the same k and a in different places.)

4 Nonce IV

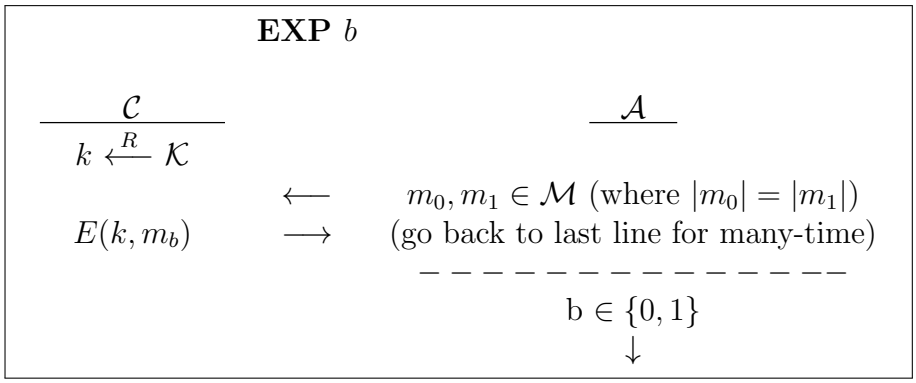
Suppose we choose the IV for counter mode as follows:

- The first 96 bits are $\xleftarrow{R} \{0,1\}^{96}$
- The final 32 bits are all 0.

For this problem, assume that every message m has $|m| \leq 2^{32}$ (including padding; see problem 5).

How many messages can we encrypt before we can break semantic security in the many-time CPA game?

CPA Security Game:


EXP b
C
A
k ←R K
m0, m1 ∈ M (where |m0| = |m1|)
E(k, mb)
(go back to last line for many-time)
b ∈ {0, 1}

5 Padding

So far, we've only been able to encrypt messages that are a multiple of 128 bits (the AES block size).

We usually use a *padding function* to turn every message into one that splits into blocks evenly. We can encrypt/decrypt that padded message, and remove the padding to get back the original message.

5.1

Consider the following function $pad(m)$:

- Write m in binary.
- Add a 1 at the end.
- Add 0 at the end until the message divides evenly into the block size.

5.1.1

Write the function *unpad* such that $unpad(pad(m)) = m$.

5.1.2

How long is $pad(m)$ in terms of $|m|$?

(Recall: $|m|$ is the length of m. If $m \in \{0,1\}^n$, then $|m| = n$.)

5.1.3

Prove that every secure block padding scheme has at least one message m whose padded version $pad(m)$ is at least one entire block longer.

5.1.4

What properties must every padding scheme have?

5.1.5

Try to come up with another padding scheme. Show that it works for any message in $\{0,1\}^n$ (for any $n \in \mathbb{Z}$).

5.2

Find a way to encrypt messages using OFB and CTR such that the resulting encryption is not any longer (apart from the IV).

5.3 Ciphertext Stealing

Don't try this until you've solved all the other problem:

Find a way to encrypt messages using CBC such that the resulting encryption is not any longer (apart from the IV).

(This one is very hard, and has a creative, non-obvious solution. It's really cool, though.)