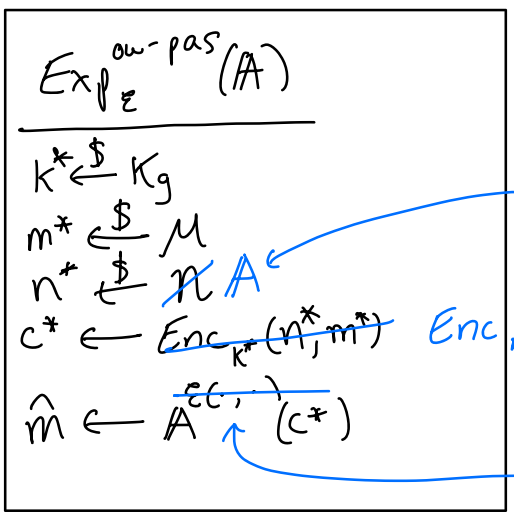
This problem sheet looks at exercising your mode of operations and reductions muscle. We'll look at different definitions of security—focusing mostly on the adversary's ability to fiddle with nonces—and see what we can see.

1. During the lectures, we skipped a one-way notion for nonce-based encryption. However, the notion is useful to describe certain attacks (as we will do below). In this question, we will develop a suitable OW-CPA notion. For simplicity, we will assume that $\mathcal{M} = \{0,1\}^\ell$ for some $\ell > 0$.

   (a) ⋆ Let's first concentrate on the adversary's goal, namely the "OW" part of OW-CPA security. How would you model a OW-PAS adversary against a nonce-based encryption scheme?

one-wayness = can't recover plaintext from ciphertext



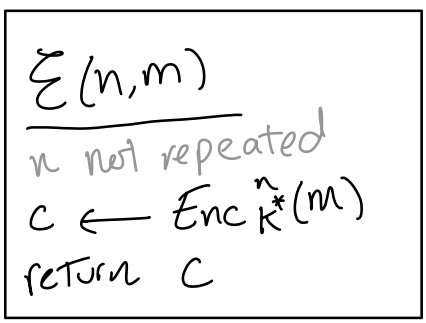
adversary gets to choose nonce for OW attack - good to know

from convention

OW-PAS = passive attacks: adversary has no oracle (missed the PAS)

   (b) ⋆ The next step is to add the adversary's power, namely the "CPA" part of OW-CPA security. Which oracle do you need to add in this case?

- cpa = chosen plaintext attack
- oracle can be supplied plaintext and (unique) nonce; returns the outputted ciphertext



✓

2. The idea of nonces is that they are unique. What happens when they are not?

   (a) ⋆ Consider AES in counter mode and suppose an adversary sees two ciphertexts of the same length, both created using the same nonce, say $n = 0^{64}$. What can the adversary learn about the plaintexts?

If the plaintexts are m_1 and m_2, the adversary can determine $m_1 \oplus m_2$.
Given the same nonce and key, the block cipher will give the same output for any block i, say x_i.
Denote $x = x_1 \| \cdots \| x_b$ as the output of the block cipher, if there are b blocks in the plaintext. ✓
Thus
$$c_1 = x \oplus m_1 \quad \text{and} \quad c_2 = x \oplus m_2.$$
It follows that
$$c_1 \oplus c_2 = (x \oplus m_1 \oplus x \oplus m_2)$$
$$= (x \oplus x) \oplus (m_1 \oplus m_2)$$
$$= m_1 \oplus m_2.$$

   (b) ⋆ Show that counter mode is not OW–CPA if nonces can be repeated by the adversary.

recall: OW-CPA = one-way chosen-plaintext attack
↖ defend against full recovery of plaintext

Using our result from part (a): if the adversary wants to recover a plaintext m encrypted using a nonce n from the ciphertext c, they need only to encrypt any new message m' into a ciphertext c', using the same nonce. Then by part (a), we have $m = c \oplus c' \oplus m'$. ✓

(can specify $m \in \{0\}^n$; then $m = c \oplus c'$)

3. Historically, many different modes other than CTR have been proposed. One of the most popular modes is so-called cipher-block-chaining (CBC), which we'll look at in this question.

   (a) ⋆ CBC mode is insecure when nonces are reused. Imagine an adversary trying to distinguish between the real and the ideal world by asking for encryptions of $(0^n, 0^n 1^n)$ and $(0^n, 0^n 0^n)$. How would it then distinguish based on the resulting ciphertexts?

Since the same IV is used, and the first block of the plaintext is the same, the first block of the ciphertext should also be identical (given by $Enc_K(0^n \oplus 0^n) = Enc_K(0^n)$).

Thus letting c_1 and c_2 be the encryptions of $(0^n, 0^n 1^n)$ and $(0^n, 0^n 0^n)$ respectively, we can define

$$A = \begin{cases} 1 & \text{if first } n \text{ bits of } c_1 \text{ and } c_2 \text{ are identical} \\ 0 & \text{otherwise} \end{cases}$$

Then $Pr[Exp_E^{real}(A) : b = 1] = 1$, so

$$\begin{aligned} Adv(A) &= 1 - Pr[Exp_E^{ideal}(A) : b = 1] \\ &= 1 - Pr[\text{first } n \text{ bits of 2 random ciphertexts are identical}] \\ &= 1 - \frac{1}{2^n}. \end{aligned}$$

(b) ⋆⋆ In fact, CBC mode is not even secure when nonces are unique. Can you come up with a distinguishing attack? Hint: two *adaptively* chosen messages suffice.

Choose nonces $n_1 = 0^n$ and $n_2 = 1^n$, and plaintexts $m_1 = 0^n$ and $m_2 = 1^n$.

Then the result of the xor step will be 0^n in both cases, meaning the output should be $c_1 = c_2 = Enc_K(0^n)$.

Thus we can define $\mathcal{A}$ as in part (a), and following the same logic, we find $Adv(\mathcal{A}) = 1 - \frac{1}{2^n}$.

✓

4. Given its insecurity, the popularity of CBC might at first be surprising. However, when the nonce is chosen *uniformly at random*, CBC is secure. In such a case the nonce is usually referred to as an *initialisation vector* (IV). An advantage of using a random value is that you do not need to worry about synchronizing across multiple devices; a disadvantage is that you rely on a good source of randomness (an expensive resource).

(a) ⋆ Imagine you use random values for the nonce and you encrypt q different messages. What is the probability that, by chance, you end up using the same nonce for two different messages?

If the nonce is n bits long, then the odds of a collision are approximately given by the birthday bound:

$$\frac{q(q-1)}{2\cdot|\mathcal{N}|} = \frac{q(q-1)}{2\cdot 2^n}$$

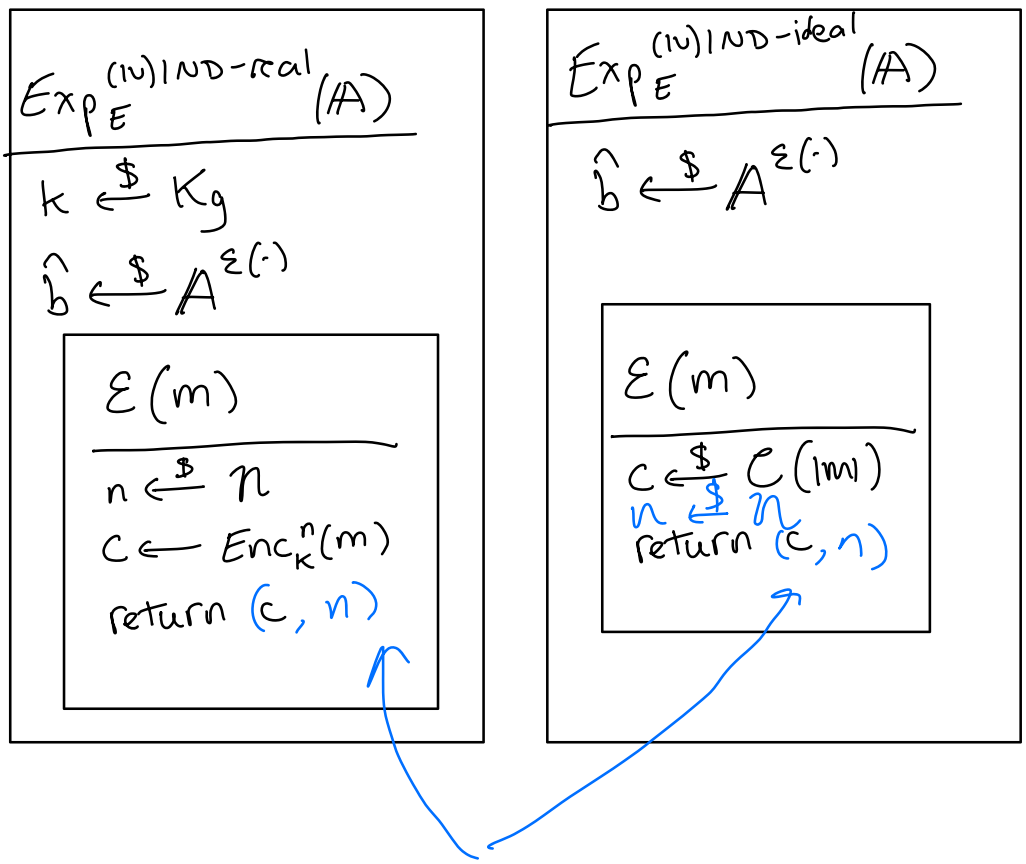
$$= \boxed{\frac{q(q-1)}{2^{n+1}}}$$

✓

(b) ⋆⋆ Draw the "real" and "ideal" experiments to define an (IV)IND advantage for the random-IV scenario? It is easiest to first describe the real world and then ensure that the ideal world matches, so its oracle takes in the same kind of inputs, producing the same kind of outputs, and rejecting the same queries. Remember Kerckhoffs and nonces.

Recall: IND = indistinguishability
Kerckhoffs = everything known except key

Define $\mathcal{N}$ to be the set of all nonces/IVs.



oracles return nonces:
they're random, but adversary
can see them (Kerckhoff's)

(c) ⋆⋆⋆ Nonce-based security implies IV-based security, as long as the probability the randomly chosen IVs collide can be contained. A semi-formal statement is that for any nonce-based encryption scheme Enc and any adversary $\mathbb{A}_{(\mathsf{iv})\mathsf{ind}}$ making q queries, there exists an equally efficient adversary $\mathbb{B}_{(\mathsf{n})\mathsf{ind}}$ such that

$$\mathsf{Adv}^{(\mathsf{iv})\mathsf{ind}}_{\mathsf{Enc}}(\mathbb{A}_{(\mathsf{iv})\mathsf{ind}}) \leq \mathsf{Adv}^{(\mathsf{n})\mathsf{ind}}_{\mathsf{Enc}}(\mathbb{B}_{(\mathsf{n})\mathsf{ind}}) + q^2/|\mathcal{N}|$$

The consequence of the birthday bound $q^2/|\mathcal{N}|$ in the statement above, coupled with a desire to allow nonce-based schemes to be used with randomly chosen IVs, is that the nonce space must be large. To give a concrete benchmark, a recent lightweight competition required that $|\mathcal{N}| \geq 2^{96}$.

Let Enc be a nonce-based encryption scheme, and $\mathbb{A}$ be an adversary making q queries.
Then we define

$\underline{Exp^{(n)ind}_{E}(\mathbb{B}_{(n)ind})}$
$k \xleftarrow{\$} Kg$
$\hat{b} \xleftarrow{\$}$ $\mathbb{B}_{(n)ind}$
$\underline{\quad}$
$\hat{b} \xleftarrow{\$} \mathbb{A}^{(iv)ind}$
return $\hat{b}$

see feedback for sample answer

Not sure where the $q^2/|\mathcal{N}|$ comes from.

5. CFB and OFB are two other modes of operation. CFB stands for cipher feedback mode. OFB stands for output feedback mode. For both, the encryption routines are depicted below, CFB to the left and OFB to the right (where we've omitted the parsing and recombining of messages and ciphertexts into blocks and back).

$\mathsf{CFB.Enc}(\mathsf{E})^n_k(m[1],\ldots,m[n])$
$c[0] \leftarrow n$
for $i \in [1,\ldots,n]$
$\quad X[i] \leftarrow \mathsf{E}_k(c[i-1])$
$\quad c[i] \leftarrow m[i] \oplus X[i]$
return $c[1],\ldots,c[n]$

$\mathsf{OFB.Enc}(\mathsf{E})^n_k(m[1],\ldots,m[n])$
$X[0] \leftarrow n$
for $i \in [1,\ldots,n]$
$\quad X[i] \leftarrow \mathsf{E}_k(X[i-1])$
$\quad c[i] \leftarrow m[i] \oplus X[i]$
return $c[1],\ldots,c[n]$

For each of the two modes, answer or discuss the following:

(a) ⋆ Define the decryption algorithms.

✓

CFB.Dec (D)

$c[0] \leftarrow n$

for $i \in [1,\ldots,n]$

$\quad X[i] \leftarrow E_K(c[i-1])$

$\quad m[i] \leftarrow c[i] \oplus X[i]$

return $m[1],\ldots,m[n]$

OFB.Dec (D)

$X[0] \leftarrow n$

for $i \in [1,\ldots,n]$:

$\quad X[i] \leftarrow E_K(X[i-1])$

$\quad m[i] \leftarrow c[i] \oplus X[i]$

return $m[1],\ldots,m[n]$

(b) ⋆ If you had to compare with the other with the two modes we have seen so far (~~ECB,~~ CBC, CTR), which mode do you find most similar?

CBC, as it also uses an chaining between block encryptions, whereas CTR encrypts each block independtly (except for knowing the position within the whole message).

(see feedback for alternate answer)

(c) ⋆⋆ Comment on the efficiency. Are encryption or decryption parallelizable? Does decryption require the decipher functionality of the blockcipher?

Encryption is not parallizable for either algo. However, decryption can be parallelized for CFB (though not for OFB).

Neither mode requires the decipher functionality. ✓

Both algorithms run in linear time (assuming block cipher encryption takes constant time).

(d) ⋆⋆⋆ Comment on their security.

They are secure against the attack specified in 3b. since encrypting the nonces first means you can't influence the output by how you choose the nonce.

Not sure what else to compare it to.

thus not (N)IND secure