COMS30023 / Cryptology

Problem Sheet 3 – Blockciphers and Standard Modes of Operation

Dr François Dupressoir*

2022/23

Introduction

This problem sheet first 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.

In the (optional) second section, we explore blockciphers a bit more in depth by considering classical designs and how they fail.

1 Modes of Operation

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

*Based on material by Dr. Martijn Stam, Dr David Bernhard and others

(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.

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?

   (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.

   (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 $\mathbb{A}_{\mathsf{(iv)ind}}$ making q queries, there exists an equally efficient adversary $\mathbb{B}_{\mathsf{(n)ind}}$ such that

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

   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 $|\mathcal{N}| \geq 2^{96}$.

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).

$\mathsf{CFB.Enc(E)}^n_k(m[1], \ldots, m[n])$
$c[0] \leftarrow n$
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 $c[1], \ldots, c[n]$

$\mathsf{OFB.Enc(E)}^n_k(m[1], \ldots, m[n])$
$X[0] \leftarrow n$
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.

   (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?

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

   (d) ⋆ ⋆ ⋆ Comment on their security.

2 Blockciphers (optional)

We will consider blockciphers for which $\mathcal{M} = \mathcal{C} = \{\mathtt{a}, \ldots, \mathtt{z}\}^9$ (that is, plaintexts are 9-letter strings).

1. (a) $\star$ Determine $|\mathcal{M}|$ and estimate, to one decimal, $\lg(|\mathcal{M}|)$.
   (b) $\star$ Determine $|\mathrm{Perm}(\mathcal{M})|$. What can you say about $\lg(|\mathrm{Perm}(\mathcal{M})|)$? ($\mathrm{Perm}(\mathcal{X})$ is the set of *permutations* of $\mathcal{X}$.)
   (c) $\star$ What do those quantities represent, and why might we be interested in them?

2. You may have heard of transposition (or *shuffling*) ciphers (for example, columnar transposition), that operate by changing the *position* of letters in a text. For example, one could use the following table to define a shuffle on nine positions.

in	1	2	3	4	5	6	7	8	9
out	4	6	1	5	3	7	9	2	8

   With this shuffle, `abcdefghi` would encipher to `cheadbfig`, so the first letter in the plaintext (`a`) goes to the fourth position in the ciphertext, and the first letter in the ciphertext (`c`) originates from the third position in the plaintext.
   (a) $\star$ Decipher `vlooibuys`.
   (b) $\star$ What is the keyspace $\mathcal{K}$ and how large are keys (in bits)?
   (c) $\star$ Find an adversary that distinguishes the shuffling cipher from a random permutation with an overwhelming advantage in a single query and minimal computation. (Calculate its advantage.)
   (d) $\star$ Suppose the adversary is more ambitious than simply distinguishing and wants to recover the key using a chosen plaintext attack. Explain how you would recover the key. Try to maximize the key recovery advantage while minimizing the number of queries and adversarial runtime.

3. Shuffling ciphers aren't very good. Another class of historical ciphers is known as *substitution* ciphers, where each letter of the alphabet is substituted by another one. For instance, one could use the following table to define the substitution.

in	`abcdefghijklmnopqrstuvwxyz`
out	`francoiszyxwvutqpmlkjhgedb`

   (a) $\star$ Decipher `atvqjkcml`.
   (b) $\star$ What is the keyspace $\mathcal{K}$ and how large are keys (in bits)?
   (c) $\star$ Find a distinguishing attach on the substitution cipher. Try to maximize the distinguishing advantage while minimizing the number of queries and adversarial runtime.
   (d) $\star\star$ Suppose the adversary is more ambitious and wants to recover the key using a chosen plaintext attack. Explain how you would recover the key. Try to maximize the key recovery advantage while minimizing the number of queries and adversarial runtime.

4. Shuffling once or substituting once is rubbish as an enciphering mechanism. But can we instead combine both operations, and iterate them a couple of times? Let's consider that we use P_k with $k \in \mathrm{Perm}(\{1, \ldots, 9\})$ to denote a shuffle, and S_k with $k \in \mathrm{Perm}(\{\mathtt{a}, \ldots, \mathtt{z}\})$ to denote a substitution. We can create an enciphering scheme E by composing shuffles and substitutions as follows.

Kg
$k_1 \leftarrow_\$ \mathrm{Perm}(\{1,\ldots,9\})$
$k_2 \leftarrow_\$ \mathrm{Perm}(\{\mathtt{a},\ldots,\mathtt{z}\})$
$k_3 \leftarrow_\$ \mathrm{Perm}(\{1,\ldots,9\})$
$k_4 \leftarrow_\$ \mathrm{Perm}(\{\mathtt{a},\ldots,\mathtt{z}\})$
return (k_1, k_2, k_3, k_4)

$\mathsf{E}_{(k_1,k_2,k_3,k_4)}(m)$
$c \leftarrow \mathsf{P}_{k_1}(\mathsf{S}_{k_2}(\mathsf{P}_{k_3}(\mathsf{S}_{k_4}(m))))$
return c

(a) ⋆ Explain how deciphering works.

(b) ⋆⋆ Argue (don't prove) that the repetition in the enciphering scheme is pointless, so we can consider only a two-key scheme $\mathsf{E}_{k_5,k_6} = \mathsf{P}_{k_5} \circ \mathsf{S}_{k_6}$ instead, without loss of generality (or security).

(c) ⋆⋆ Come up with a distinguishing attack.

(d) ⋆⋆⋆ Sticking to the simplification from (b), describe a key recovery attack under chosen plaintext attack that has advantage 1. You don't have to try to minimize the number of queries, but try to avoid exhaustive search, while still being guaranteed to recover the key (k_5, k_6).

5. Shuffling and substitution on their own, and taken together, are simply not good enough. We throw another ingredient into the mix. The Vigenère cipher is a generalization of Caesar's cipher, where letters of the alphabet are added together, identifying a with 1, b with 2, all the way up to identifying z with 0. Additions are done modulo 26. Given two words of the same length, we can add them letter by letter.

Kg
$k \leftarrow_\$ \{\mathtt{a},\ldots,\mathtt{z}\}^9$
return k

$\mathsf{V}_k(m)$
$c \leftarrow m + k$
return c

(a) ⋆ Use Shannon's theorem to demonstrate that Vigenère's scheme is perfectly secret.

As mentioned, we'd like to combine the Vigenère cipher with shuffles and substitutions. The hope is that having three different mechanisms in play will work better than only the two. We first consider whether repetition helps, or whether it is as pointless as with substitutions and shuffles.

(b) ⋆⋆⋆ Argue that when combining Vigenère with shuffles, repetition is pointless. That is, for all k_1, k_2, k_3, k_4, there exist k_5 and k_6 such that

$$\mathsf{P}_{k_5} \circ \mathsf{V}_{k_6} = \mathsf{P}_{k_1} \circ \mathsf{V}_{k_2} \circ \mathsf{P}_{k_3} \circ \mathsf{V}_{k_4}$$

(c) ⋆⋆ When combining Vigenère with substitutions, repetitions *do* in fact add complexity. However, you can still find a distinguishing attack. Do so.

(d) ⋆ ⋆ ⋆ Suppose that a cipher consists of ten repetitions, or rounds, each consisting of a substitution followed by Vigenère—all with independent keys. Describe an efficient chosen plaintext attack that recovers a complete description of the keyed encryption and decryption algorithms. (As lookup tables or functions—this is easier than recovering all 20 subkeys!)
(Hint 1: Which letters of the plaintext does the ith letter of the ciphertext depend on? — Answering this will help you understand how you can recover an algorithm to decrypt without recovering the entire key.)
(Hint 2: 26 chosen plaintexts will suffice.)

COMS30023 / Cryptology

Problem Sheet 4 – Authentication and Authenticated Encryption

Dr François Dupressoir*

2022-23

Introduction

In this work sheet, we investigate message authentication codes and how they relate to modes of operation. We'll also look at some of the complications that arise when you want a MAC that works for variable length messages instead of messages of a fixed length only.

We then discuss some interesting consequences of authenticated encryption.

Authentication Modes of Operation

1. Existential unforgeability implies universal unforgeability (both under chosen message attacks). In this question we will write a reduction to prove this statement.
   (a) $\star$ Figure out the logic of the reduction. Which of the two options below do we need to show and can you explain why?
      i. For every adversary $\mathbb{A}_{\mathsf{euf\text{-}cma}}$ that runs in time at most t and makes at most q queries to its Tag oracle there exists a reduction $\mathbb{B}_{\mathsf{uuf\text{-}cma}}$ that runs in time at most t' and makes at most q' queries to its Tag oracle (where t' and q' are reasonably related to t and q) such that:
      $$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathrm{MAC}}(\mathbb{A}_{\mathsf{euf\text{-}cma}}) \leq \mathsf{Adv}^{\mathsf{uuf\text{-}cma}}_{\mathrm{MAC}}(\mathbb{B}_{\mathsf{uuf\text{-}cma}})$$
      ii. For every adversary $\mathbb{A}_{\mathsf{uuf\text{-}cma}}$ that runs in time at most t and makes at most q queries to its Tag oracle there exists a reduction $\mathbb{B}_{\mathsf{euf\text{-}cma}}$ that runs in time at most t' and makes at most q' queries to its Tag oracle (where t' and q' are reasonably related to t and q) such that:
      $$\mathsf{Adv}^{\mathsf{uuf\text{-}cma}}_{\mathrm{MAC}}(\mathbb{A}_{\mathsf{uuf\text{-}cma}}) \leq \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathrm{MAC}}(\mathbb{B}_{\mathsf{euf\text{-}cma}})$$
   (b) $\star\star$ Describe (in words), draw (as a diagram), or write down (as code) the reduction.
   (c) $\star\star$ Analyse the runtime, queries and advantage of your reduction $\mathbb{B}_{\mathsf{euf\text{-}cma}}$ compared to those of $\mathbb{A}_{\mathsf{uuf\text{-}cma}}$.
2. Let E_k be the enciphering algorithm of a blockcipher with key k and block length ℓ. Let b be some positive integer. We will create a MAC for message space $\mathcal{M} = \{0,1\}^{b \cdot \ell}$. As usual, we represent a message $m \in \mathcal{M}$ as a sequence $m[1], \ldots, m[b]$ of plaintext blocks (of bits each).

*Based on notes by Dr. Martijn Stam, Dr David Bernhard and others

$\mathsf{Exp}^{\text{weak-ow-cca}}_{\text{Enc}}(\mathbb{A})$

$k \leftarrow_{\$} \mathsf{Kg}$
$n^* \leftarrow_{\$} \mathcal{N}$
$m^* \leftarrow_{\$} \mathcal{M}$
$c^* \leftarrow \mathsf{Enc}_k^{n^*}(m^*)$
$\widehat{m} \leftarrow_{\$} \mathbb{A}^{\mathcal{D}(\cdot,\cdot)}(n^*, c^*)$

$\mathcal{D}(n, c)$

require $(n, c) \neq (n^*, c^*)$

$m \leftarrow \mathsf{Dec}_k^n(c)$
return m

$$\mathsf{Adv}^{\text{weak-ow-cca}}_{\text{Enc}}(\mathbb{A}) = \Pr\left[\mathsf{Exp}^{\text{weak-ow-cca}}_{\text{Enc}}(\mathbb{A}) : \widehat{m} = m^*\right]$$

Figure 1: A weak notion of one-wayness against chosen ciphertext attacks.

(a) ⋆ Recall how CBC-MAC works.

(b) ⋆⋆ Write down a description for a hypothetical CFB-MAC (see the description of CFB mode in the previous work sheet), that works by encrypting $m[2], \ldots, m[b]$ in CFB mode with nonce $m[1]$ to obtain $c'[2], \ldots, c'[b]$, and returning as tag the enciphering $\mathsf{E}_k(c'[b]$ of the last block of ciphertext.

(c) ⋆⋆⋆ Argue that CBC-MAC and CFB-MAC as just defined are equivalent. (Given the same input, they produce the same output.)

3. For CBC-MAC, we mentioned that the "vanilla" version is secure only for messages of fixed length $b \cdot \ell$. If we want to use CBC-MAC on a variable number of blocks, some form of post-processing is needed, or else the mode is insecure. We now demonstrate this insecurity, which gives rise to a *length extension* attack.

   Consider an adversary that asks for the tag on the one block message $m[1] = 0^\ell$, receiving tag t in return. The claim is that the adversary can now create, without any additional Tag queries and with probability 1, a forgery for the two-block message with $\widehat{m} = 0^\ell \| t$.

   (a) ⋆⋆ What is the forged tag for $\widehat{m}$, and why is it valid?

   (b) ⋆ Is this length extension attack an existential or universal forgery? Is it a passive or chosen message attack?

   (c) ⋆⋆⋆ Could you generalize the attack? You might, for example, think about producing forgeries for longer messages; or letting the adversary choose the first message block.

   (d) ⋆⋆⋆ One possible form of postprocessing is *truncation*. That is, to create a $\tau < \ell$ bit tag, return the τ most significant bits of vanilla CBC-MAC's output. How does truncation affect EUF-CMA-security?

Authenticated Encryption

Consider the weak CCA-like one-way security experiment shown in Figure 1.

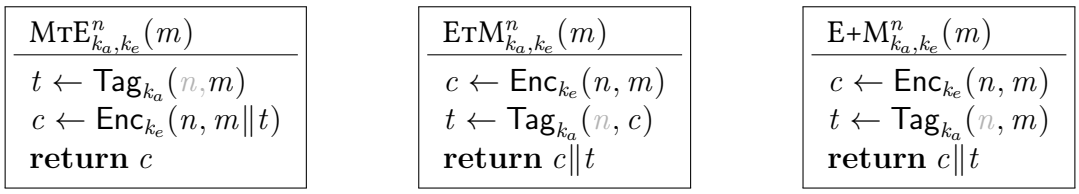


Figure 2: Generic composition without nonce authentication

▶ Note that the adversary here is weaker than the standard CCA adversary seen, for example, in (N)IND-CCA: the attacks we consider do not require the adversary to control the nonce (so we use random IVs), or to make encryption queries (so we do not give the adversary an encryption oracle). As a consequence, this notion is not very interesting for its own sake, and you don't have to remember it beyond the end of this problem sheet.

1. ⋆⋆ Show that CBC Mode is *not* weak-OW-CCA-secure. (Aim for "reasonable" time and query complexity, and an advantage as close as you can get it to 1.) You may assume that the challenge message consists of two blocks, without imposing a similar limit on the ciphertexts you can query the decryption oracle on.

2. ⋆⋆ Show that CTR Mode is *not* weak-OW-CCA-secure. You may assume that the challenge message consists of two blocks and try to come up with an attack that only request decryptions of two-block ciphertexts.

3. Figure 2 shows all three generic composition modes with the explicit nonce-authentication "greyed out". From the three types of generic composition, we saw that encrypt-then-mac (the middle panel) was the preferred option. For encrypt-then-mac, it is crucial that the nonce is not just used for encryption, but is also explicitly authenticated. In this question we will look how leaving out nonce authentication, affects the integrity of ciphertexts, and the overall security of the constructed encryption scheme.
   (a) ⋆ Consider Encrypt-then-Mac without nonce authentication. What does decryption for this mode look like? Specifically, how do you determine the validity of ciphertexts?
   (b) ⋆⋆ Consider a valid nonce–ciphertext pair (n^*, c^*). Can you come up with another *valid* nonce–ciphertext pair? (One that will successfully decrypt.)

COMS30023 / Cryptology

Symmetric Cryptography and Provable Security — Model Answers

Dr François Dupressoir*

2022/23

1 Modes of Operation (Week 4)

1b : OW-CPA can be defined as below.

$\mathsf{Exp}^{\text{ow-cpa}}_{\mathsf{Enc}}(\mathbb{A})$
$k \leftarrow_{\$} \mathsf{Kg}$
$n^* \leftarrow_{\$} \mathbb{A}^{\mathcal{E}_{\mathsf{cpa}}(\cdot,\cdot)}$
$m^* \leftarrow_{\$} \{0,1\}^{\ell}$
$c^* \leftarrow \mathsf{Enc}^{n^*}_k(m^*)$
$\widehat{m} \leftarrow_{\$} \mathbb{A}^{\mathcal{E}_{\mathsf{cpa}}(\cdot,\cdot)}(c^*)$

$\mathcal{E}_{\mathsf{cpa}}(n, m)$
$c \leftarrow \mathsf{Enc}^n_k(m)$
return c

$$\mathsf{Adv}^{\text{ow-cpa}}_{\mathsf{Enc}}(\mathbb{A}) = \Pr\left[\widehat{m} = m^*\right]$$

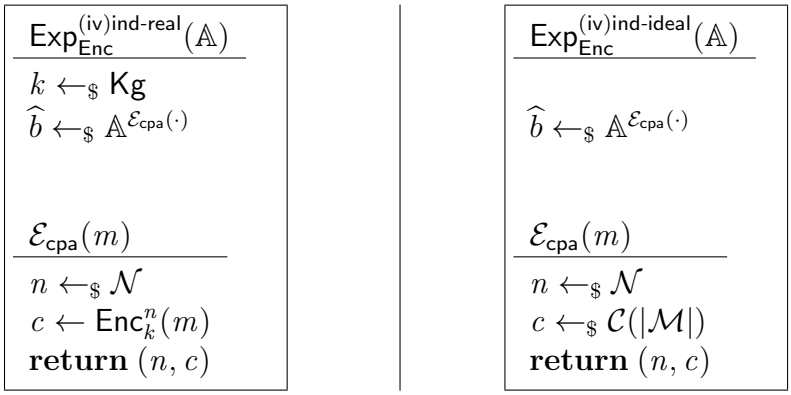
Discussion: We made a choice for the generation of the challenge nonce, to ask the adversary to choose it, and give the adversary access to the CPA oracle while doing so—this is the most powerful adversary model for OW-CPA. It would also have been a valid answer, given the creativity involved in the question, to either generate the challenge nonce at random, or ask the adversary to generate it *without* access to the CPA oracle. (These are all distinct notions! Try to prove implications between them.)

2a : As the nonces are identical, the same keystream will be generated and then xored to the messages. Thus we end up essentially using a one-time pad twice. By xoring the two ciphertexts, we cancel out the keystream contribution and reveal the xor of the two corresponding plaintexts.

2b : Suppose the challenge nonce–ciphertext pair is (n^*, c^*). Given this input, the adversary queries her CPA oracle on $\left(n^*, 0^{|c^*|}\right)$, receiving some ciphertext c in return. Then $\widehat{m} \leftarrow c^* \oplus c$ will equal the challenge message $m^* = \mathsf{Dec}^{n^*}_k(c^*)$, following the property of CTR mode discussed in 2a.

*Based on material by Dr. Martijn Stam, Dr David Bernhard and others

4b : (IV)IND security can be defined with the following experiments and advantage.



$$\mathsf{Adv}^{\mathsf{(iv)ind}}_{\mathsf{Enc}}(\mathbb{A}) = \Pr\left[\mathsf{Exp}^{\mathsf{(iv)ind\text{-}real}}_{\mathsf{Enc}}(\mathbb{A}) : \widehat{b} = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{(iv)ind\text{-}ideal}}_{\mathsf{Enc}}(\mathbb{A}) : \widehat{b} = 1\right]$$

4c : We want to prove that any nonce-based scheme that is IND secure is secure as an IV-based scheme. By reduction, we assume an adversary $\mathbb{A}_{\mathsf{(iv)ind}}$ against the (IV)IND security of the scheme, and show that there exists an adversary $\mathbb{B}_{\mathsf{(n)ind}}$ against the nonce-based IND security of the scheme, whose time and query complexities and advantage are not outlandishly far from those of $\mathbb{A}_{\mathsf{(iv)ind}}$.

When $\mathbb{B}_{\mathsf{(n)ind}}$ starts running, it simply runs $\mathbb{A}_{\mathsf{(iv)ind}}$. Whenever $\mathbb{A}_{\mathsf{(iv)ind}}$ make a query m to its CPA oracle, $\mathbb{B}_{\mathsf{(n)ind}}$ samples a nonce n uniformly at random, queries its own CPA oracle on input (n, m) to obtain a ciphertext c, and then outputs (n, c) to $\mathbb{A}_{\mathsf{(iv)ind}}$. If and when $\mathbb{A}_{\mathsf{(iv)ind}}$ terminates with some output $\widehat{b}$, $\mathbb{B}_{\mathsf{(n)ind}}$ terminates without output $\widehat{b}$.

We now analyse the reduction's time and query complexity. The running time of $\mathbb{B}_{\mathsf{(n)ind}}$ is the running time of $\mathbb{A}_{\mathsf{(iv)ind}}$ plus the cost of sampling q nonces—where q is the number of queries $\mathbb{A}_{\mathsf{(iv)ind}}$ makes to its CPA oracle. The reduction $\mathbb{B}(\mathsf{n})\mathsf{ind}$ makes exactly as many queries to its CPA oracle as $\mathbb{A}_{\mathsf{(iv)ind}}$ makes to its own.

Finally, we analyse the reduction's advantage. Ignoring problems related to nonce unicity for the moment, we see (by inspection) that if $\mathbb{B}_{\mathsf{(n)ind}}$ is in the real world, then $\mathbb{A}_{\mathsf{(iv)ind}}$ is itself in its real-world experiment. Similarly, if $\mathbb{B}_{\mathsf{(n)ind}}$ is in the ideal world, then $\mathbb{A}_{\mathsf{(iv)ind}}$ is also in its ideal world. Therefore, *as long as* $\mathbb{B}_{\mathsf{(n)ind}}$ *does not make any queries with repeat nonces to its oracle*, we have

$$\mathsf{Adv}^{\mathsf{(n)ind}}_{\mathsf{Enc}}(\mathbb{B}_{\mathsf{(n)ind}}) = \mathsf{Adv}^{\mathsf{(iv)ind}}_{\mathsf{Enc}}(\mathbb{A}_{\mathsf{(iv)ind}})$$

However, $\mathbb{B}_{\mathsf{(n)ind}}$ draws its nonces at random, and may sample the same nonce twice. If this occurs, the (N)IND advantage of $\mathbb{B}_{\mathsf{(n)ind}}$ is not defined. As a result, the advantage of $\mathbb{B}_{\mathsf{(n)ind}}$ is lower than that of $\mathbb{A}_{\mathsf{(iv)ind}}$ by the probability of sampling twice the same nonce. By (a coarse approximation of) the birthday bound, we have

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

We conclude the proof by simple algebraic manipulation.

Discussion: The above gives one way of answering reduction question: describing the operation of the reduction in plain(-ish) English. Later on, we see another reduction which is described differently. There is no one way of describing a reduction; I have a preferred way, but I do my best not to let my bias affect my marking.

2 Authentication (Week 5; Section 1)

1b : The following shows the reduction $\mathbb{B}_{\mathsf{euf\text{-}cma}}$.

$\mathbb{B}^{\mathcal{T}_{\mathsf{cma}}(\cdot)}_{\mathsf{euf\text{-}cma}}$

$m^* \leftarrow_{\$} \mathcal{M}$

$\widehat{t} \leftarrow_{\$} \mathbb{A}^{\mathcal{T}_{\mathsf{cma}}}_{\mathsf{uuf\text{-}cma}}(m^*)$

return $(m^*, \widehat{t})$

1c : The runtime of $\mathbb{B}_{\mathsf{euf\text{-}cma}}$ is roughly that of $\mathbb{A}_{\mathsf{uuf\text{-}cma}}$ (plus the cost of sampling a challenge message).

Whenever $\mathbb{A}_{\mathsf{uuf\text{-}cma}}$ wins and outputs a valid $\widehat{t}$ for m^*, then $(\widehat{m}, \widehat{t})$ is a valid message-tag pair. Moreover, since $\mathbb{A}_{\mathsf{uuf\text{-}cma}}$ is prohibited from querying m^* to its CMA oracle and $\mathbb{B}_{\mathsf{euf\text{-}cma}}$ only used her own oracle to answer queries made by $\mathbb{A}_{\mathsf{uuf\text{-}cma}}$, the message $\widehat{m}$ is guaranteed to be fresh. We therefore have the desired inequality.

$$\mathsf{Adv}^{\mathsf{uuf\text{-}cma}}_{\mathrm{MAC}}(\mathbb{A}_{\mathsf{uuf\text{-}cma}}) \leq \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathrm{MAC}}(\mathbb{B}_{\mathsf{euf\text{-}cma}})$$

3 Authenticated Encryption (Week 5; Section 2)

2 : To show that some concrete (or semi-concrete) construction is not secure, we construct an adversary whose running time is reasonable (usually a constant number of queries and almost no computation) and whose advantage is very close to 1.

Since CTR encryption works by generating a keystream and using it to mask the plaintext, we can simply mask the ciphertext, receive a message, and then undo the mask. Conretely, let (n^*, c^*) be the challenge ciphertext, where we know that c^* is two blocks long. Select some non-zero two block mask Z and set $c \leftarrow c^* \oplus Z$. Since we have $c \neq c^*$, we can query the decryption oracle on (n^*, c) retrieving some plaintext m. Then as guess for the message, we can return $\widehat{m} \leftarrow m \oplus Z$, which will be correct with probability 1.

3b : The decryption of (n, c^*) for any $n \neq n^*$ will succeed. Even if this outputs random garbage, this constitutes a valid forgery.