

Hw1 Solutions

CIS 4360 — Viet Tung Hoang — Spring 2025

Your Name Here!

Problem 1

Here I assume that you have installed Latex properly in your machine. You should compile this file under `pdflatex`.

Problem 2

One of the most important aspects of Latex is its math mode. Mathematical symbols should look like x or X_5 or e^t . Never write something like `x` in ordinary text mode—it looks terrible.

Problem 3

Occasionally you need to draw pictures. To do that, you first need to produce a picture file in pdf format—here my file is `ecb.pdf`—and then insert it in the latex file. To draw pictures, I use PowerPoint. You then can print the picture as a pdf file, and use some other tools to crop the image. The resulting picture is shown in Figure 1.

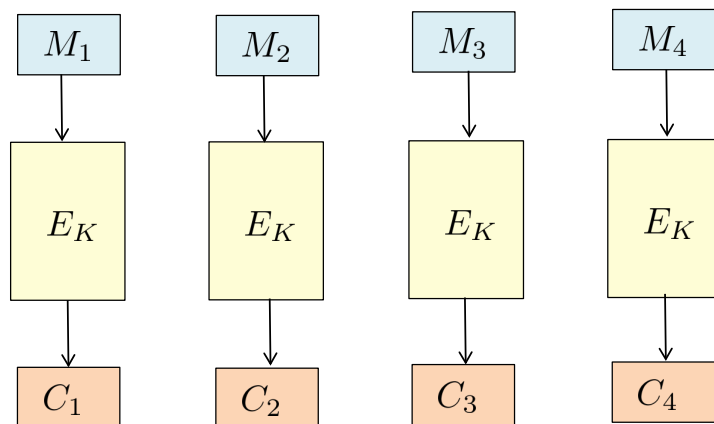


Figure 1: The ECB mode of encryption, illustrated for 4 blocks.

Problem 4

In this course we'll routinely use the following notation. For a finite set S , we write $x \leftarrow S$ to denote picking an element of S uniformly at random and assigning it to x , and we write $|S|$ to

denote the number of elements of S . We write 0^n to denote the all-zero string of n bits, and 1^n the all-one string of n bits. Let $\{0, 1\}^n$ be the set of all n -bit binary strings and $\{0, 1\}^*$ the set of all binary strings. For binary strings x and y , we write $|x|$ to denote the length of x , and $x||y$ the concatenation of x and y . If x and y also have the same length, we write $x \oplus y$ to denote their xor. We use \perp to denote a symbol that indicates invalidity; you can think of it as NULL.

Problem 5

In homework you'll be asked to give attacks. Below is how you should write an attack to break the left-or-right security notion of the ECB mode of encryption. Don't worry about the technical details; you'll learn them later. The takeaway lesson here is:

- You should write pseudocode to describe your attack, and accompany it with some English explanation.
- Always analyze your attack by calculating its *advantage*. This is a number from 0 to 1 that measures how likely your attack will succeed.
- In our class you'll learn lots of attack notions, and as a result, you are likely to mistake one for another. Always consult the notes to make sure that you are giving the correct form of attack.

We now construct an adversary A that breaks the left-or-right security notion of the ECB mode. The code of the adversary is given in Figure 2.

```

adversary  $A^{\text{ENC}(\cdot, \cdot)}$ 
Pick arbitrary distinct messages  $M_0, M_1$  of the same length
 $C_0 \leftarrow \text{ENC}(M_0, M_0)$ ;  $C_1 \leftarrow \text{ENC}(M_0, M_1)$ 
if  $C_0 = C_1$  then return 0 else return 1

```

Figure 2: The code of an adversary A breaking the left-or-right security of ECB.

Informally, the adversary A first picks two arbitrary messages M_0 and M_1 of the same length. It then queries $C_0 \leftarrow \text{ENC}(M_0, M_0)$ to get a ciphertext C_0 of M_0 , and then queries $C_1 \leftarrow \text{ENC}(M_0, M_1)$. The adversary will output 1 (meaning that it believes that it's in the right world) if and only if $C_0 \neq C_1$.

If we are in the left world then C_1 is a ciphertext of M_0 . Because ECB is deterministic, we must have $C_0 = C_1$, and in that case the chance that the adversary outputs 1 is 0. If we are in the right world, meaning that C_1 is a ciphertext of M_1 , then we must have $C_1 \neq C_0$ due to the fact that $M_0 \neq M_1$ and the ECB mode is perfectly invertible. Thus the chance that the adversary outputs 1 is 1. Hence the left-or-right advantage of the adversary is $1 - 0 = 1$.