Homework 1: Encryption and Modes of Operation

Submission policy. Submit your answers on Blackboard by 11:59pm on **Friday**, Feb. 15, 2019. Late submissions will be penalized according to course policy. Your need to TYPESET your answers and your submission MUST include the following information:

1. A pdf file named as **LastnameHW1.pdf** with with all the answers to the theoretical problems and theoretical parts of the Project Part. At the end of the .PDF you must include a list of references used (online material, course nodes, textbooks, wikipedia, etc.)
2. A compressed file including all the files needed for the Project part named as **LastnameHW1.zip**.
3. Your name should be included in EVERY single file submitted.

Adminstration. This assignment will be graded and administered by Panagiotis Chatzigiannis.

Structure. Every ISA 656 consists of two parts. The theory part and the practical part. In this homework each part worths 50 points. HW1 accounts for 8% of your final grade.

Exercise 1. Piazza [2 points] The first step of this assignment (which you get points for!) is to sign up on Piazza and understand the course policy on communications:

> If you have a question about the course you should: (a) Come to office hours, OR (b) Post on Piazza. Do not use private posts/emails to ask technical questions. The rest of the class is probably also interested in your question, so make it public!

Exercise 2. Classic Ciphers [12 points] You know that your friend always uses one of the two following passwords: "abcd" or "kdmf". You get to see an encryption of your friend's password, can you find what was it if (WITHOUT assuming any type of brute-forcing, your attack should work by observation or pen and paper calculations.):

(a.) The encryption was done using the Caesar's (or shift) cipher?

(b.) The encryption was done using Vigener cipher with key length 2?

(c.) The encryption was done using Vigener cipher with key length 3?

(d.) The encryption was done using Vigener cipher with key length 4?

Exercise 3. Computational Security [13 points] Let $E_K(\cdot)$ and $D_K(\cdot)$ be the encryption and decryption algorithms of a symmetric key cryptosystem with ℓ-bit keys and n-bit plaintexts and ciphertexts. We create a new symmetric key cryptosystem that extends the size of the keys to 2ℓ-bits by applying twice $E_K(\cdot)$ and $D_K(\cdot)$:

$E'_{(K_1,K_2)}(M) = E_{K_2}(E_{K_1}(M))$
$D'_{(K_1,K_2)}(C) = D_{K_1}(D_{K_2}(C))$

1. **[1 point]** What is the size of plaintexts and the size of ciphertexts for the new cryptosystem?

An adversary wants to perform a brute-force attack on the cryptosystem suggested above in order to obtain keys K_1 and K_2. His only information is one valid pair of plaintext M and ciphertext C. Also, the adversary has access to a single processor core and all computations performed to aid the brute-force are counted as part of the run-time of the brute-force. For simplicity, assume that there are no spurious keys and that all cryptographic operations take constant time.

2. **[4 points]** Suppose that the adversary has only n-bit memory. Describe the brute force attack in pseudo-code and estimate the number of encryption and decryption operations performed in addition to the overall runtime of the algorithm.

3. **[4 points]** Suppose now that the adversary has $n \cdot 2^{\ell}$ memory (here the adversary is basically given exponential space, which is something that should never happen in real life). Describe, again in pseudo-code, a more efficient way to perform the attack and estimate the number of encryption and decryption operations performed and overall runtime.

4. **[4 points]** In response to increases in computational power, the Data Encryption Standard (DES), a legacy symmetric cryptosystem, was extended to support longer keys without designing a new algorithm. This was accomplished by developing Triple DES (3DES), which encrypts the plaintext three times with three different keys, with decryption applying the same keys in reverse order (the same as previously described scheme, but with one more key). Given again $n \cdot 2^{\ell}$ space, describe how to modify the attack of the previous part to break 3DES. Again, estimate the number of decryption and encryption operations performed along with the overall runtime.

Exercise 4. CPA Security [10 points] Consider the following notation: the symbol $||$ denotes string concatenation, the symbol $\oplus$ is the bit-wise XOR operator and the symbol $|m| = n$ denotes that the bitstring m has length n.

1. **[5 points]** Consider the following encryption scheme: to encrypt a message m of length n with a key k of size $n/2$ set the ciphertext $c = x \oplus F_k(x)$, where $F_k()$ is a length preserving pseudorandom function. Prove that this is not a CPA secure encryption scheme (i.e., give a counterexample that shows that it does not satisfy the definition discussed in class).

2. **[5 points]** Let $H()$ be a collision resistant hash function H that maps $2n$-bit strings to n bit strings. The key is a random bitstring $k \in \{0,1\}^n$. To encrypt a message m of length n, choose a random $2n$-bit string r and output the ciphertext

$$r||(H(r) \oplus m \oplus k)$$

Prove that this is not a CPA secure encryption scheme (i.e., give a counterexample that shows that it does not satisfy the definition discussed in class).

Exercise 5. [13 points] Recall the definition of a block cipher we discussed in class: it is an algorithm that implements a function $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ (i.e. it takes as input a key of size k bits and a plaintext of size n bits and outputs a ciphertext of size n bits), such that for all possible secret key values $K \in \{0,1\}^k$, the function $E_K : \{0,1\}^n \rightarrow \{0,1\}^n$ is a permutation, or in other words the function is one-to-one.

1. **[3 points]** Is the following image a description of a valid block cipher? Rows and columns correspond to keys and plaintexts, respectively, and the table entry for row K and column X is the output of the algorithm, i.e. the corresponding ciphertext. Justify your answer. (Note that we do not ask whether it is a *secure* block cipher).

K / X	000	001	010	011	100	101	110	111
00	000	101	010	111	011	110	001	100
01	111	010	110	000	011	101	100	001
10	101	000	001	011	110	010	111	100
11	010	100	001	101	011	000	111	110

2. **[3 points]** The function $E' : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ which is defined as $E'_K(X) = X$ is a valid block cipher. Explain why. (As before, note that we do not ask whether it is a *secure* block cipher).

3. **[3 points]** The function $E'' : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ which is defined as $E''_K(X) = \bar{X} \oplus K$ (where $\bar{X}$ is obtained by flipping all the bits of X, i.e. if $X = 01001$ then $\bar{X} = 10110$) is a valid block cipher. Explain why. (As before, note that we do not ask whether it is a *secure* block cipher).

4. **[4 points]** Let us now think about block cipher security. Do you think that E' and E'' (as described above) are *secure* block ciphers for the case when $n = 128$? Justify your answer!

Programming assignment

In this lab, you will investigate the strengths and weaknesses of various block cipher modes of operation.

Objectives:

- Get familiar with how to use different modes of operation via a python library.
- Understand the importance of CPA security in practice.
- Attack insecure modes of operation.

Setup:

We strongly recommend working at a Linux or Mac OS machine for this project. If you do not have access to one please consider using a Virtual Machine.

For this project you will need to code in Python. The given code is compatible with Python2. You will also have to use the `pycrypto` library. For setup instructions look here: https://techtutorialsx.com/2018/04/09/python-pycrypto-using-aes-128-in-ecb-mode/

For this assignment **you need to download** the files located here: https://drive.google.com/file/d/1CQhHrb85pinlmyzhzMiK6-euWAa9m1ML/view?usp=sharing

Part 1. Getting Familiar with Modes of Operation

A mode of operation is a way of using block ciphers like AES, which only encrypt a fixed number of bits, to encrypt arbitrary-length messages. You can experiment with AES and modes of operation in Python.

```
THIS_MODE = AES.MODE_ECB
KEY_SIZE = AES.key_size[0]
key = Random.new().read(KEY_SIZE)
iv = Random.new().read(AES.block_size)
message = ’hello world’
cipher = AES.new(key, THIS_MODE, iv)
ciphertext = cipher.encrypt(pad(message))
```

We suggest you use the following pad and unpad functions to pad your message with 0’s to the appropriate length before encrypting, and remove the zeros upon decryption:

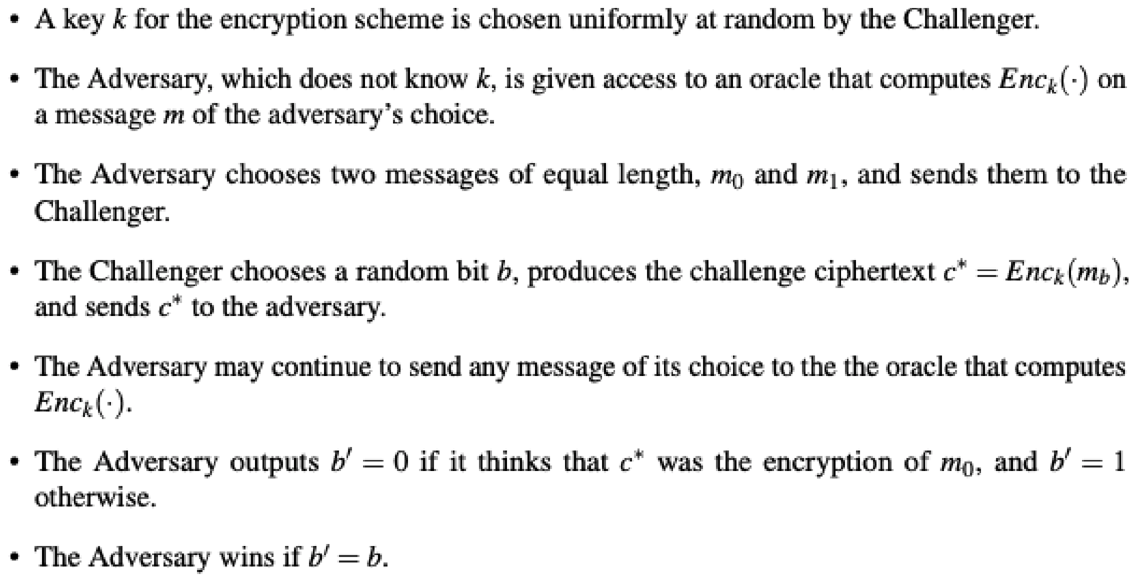
```
def pad(s):
    """Takes in a string, and returns that string with zeros appended.
    Padding should be performed prior to encryption.
    """
    return s + b"\0" * (AES.block_size - len(s) % AES.block_size)

def unpad(s):
    """Removes trailing zeros from a string.
    Unpadding should be performed after decryption.
    """
    return s.rstrip(b"\0")
```

Part 2. Electronic Code Book (ECB) Mode

The ECB mode essentially encrypts each block of the ciphertext independently.

In file `tux6.ppm`, you will find an image. (PPM binary format is a simple image format where the pixels were represented as uncompressed bytes.) Encrypt that image using Electronic Code Book (ECB) mode. Encrypt the same image using Cipher Block Chaining (CBC) mode. (When encrypting, remove the three header lines from tux6.ppm, and prepend them unencrypted to the ciphertext.)

Recall the definition of CPA security for a symmetric encryption scheme. When playing the following game with a Challenger, no efficient Adversary should be able to win with probability much more than half:



- A key k for the encryption scheme is chosen uniformly at random by the Challenger.
- The Adversary, which does not know k, is given access to an oracle that computes $Enc_k(\cdot)$ on a message m of the adversary's choice.
- The Adversary chooses two messages of equal length, m_0 and m_1, and sends them to the Challenger.
- The Challenger chooses a random bit b, produces the challenge ciphertext $c^* = Enc_k(m_b)$, and sends c^* to the adversary.
- The Adversary may continue to send any message of its choice to the the oracle that computes $Enc_k(\cdot)$.
- The Adversary outputs $b' = 0$ if it thinks that c^* was the encryption of m_0, and $b' = 1$ otherwise.
- The Adversary wins if $b' = b$.

Prove that AES in ECB mode cannot satisfy the definition of CPA-secure encryption. In other words, present an algorithm for an Adversary that wins the game described above.

What to submit

1. A file called ECB_image.ppm, with the image encrypted using Electronic Code Book mode.
2. A file called CBC_image.ppm, with the image encrypted using Cipher Block Chaining mode.
3. A file called ECB_writeup.txt, with a proof that AES in ECB mode is not CPA-secure.

Note that you have to add the header lines back to the encrypted images you submit so that the files can be opened with an image viewer.

Part 3. Cipher Block Chaining (CBC) Mode

Cipher Block Chaining works by XORing each ciphertext block with the next plaintext before encryption. For the first block, a random initialization vector (IV) is used instead. Note that the IV needs to be included as part of the ciphertext in order for decryption to be possible. It has been proven that CBC mode, unlike ECB mode, is CPA-secure if the block cipher encryption function is a pseudorandom permutation. However, if the IVs are incremented like a counter instead of being chosen at random, this breaks down, as you will show in this section.

In the given files, in CBC_ciphertext you will find a ciphertext. You know that it was produced using the initialization vector in CBC_iv1, and that the ciphertext encrypts the message in CBC_message1: “turn to page 01”. The sender isn’t using CBC mode correctly, they should be choosing random initialization vectors, but instead, they are incrementing the initialization vector like a counter, adding 1 to it for every message sent. A few messages later, they will be using the IV in CBC_iv2. What two messages can you specify as m_0 and m_1 in the CPA security game when it is time to use CB_iv2, such that you will be able to tell the difference between the two encryptions?

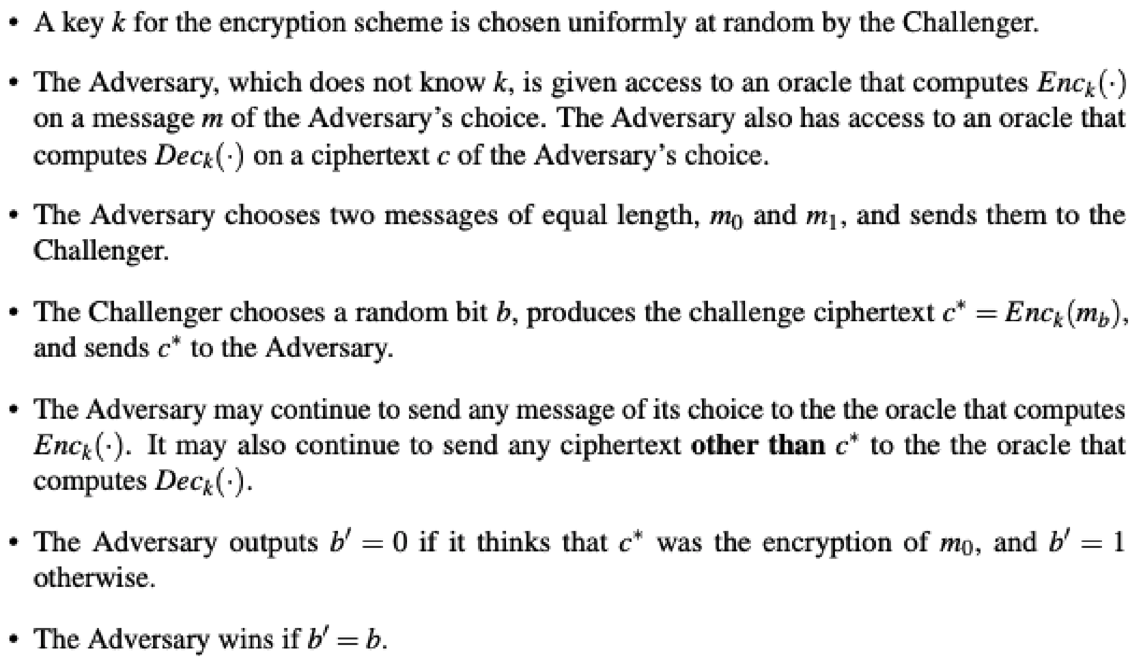
What to submit A file called CBC_writeup.txt describing m_0 and m_1, and explaining why this choice of message would allow you to break CPA-security

Part 4. Counter (CTR) Mode

In Counter mode a random IV is chosen (as in CBC mode), and IV, IV +1, IV +2, etc are fed through the block cipher encryption function to produce a stream of random looking bits to XOR the message with. It has been proven that counter mode is CPA-secure if the block cipher encryption function is a pseudorandom permutation. However, it is malleable, meaning that an adversary who sees a ciphertext can produce a different ciphertext encrypting a related message!

In files CTR_ciphertext and CTR_iv, you can find an AES in CTR mode ciphertext and the corresponding initialization vector. You know that the ciphertext encrypts a four digit number n that is a multiple of 10. Find a ciphertext (using the same IV) encrypting $n+5$.

Next, recall the definition of CCA security for a symmetric encryption scheme. When playing the following game with a Challenger, no efficient Adversary should be able to win with probability much more than half:



- A key k for the encryption scheme is chosen uniformly at random by the Challenger.
- The Adversary, which does not know k, is given access to an oracle that computes $Enc_k(\cdot)$ on a message m of the Adversary's choice. The Adversary also has access to an oracle that computes $Dec_k(\cdot)$ on a ciphertext c of the Adversary's choice.
- The Adversary chooses two messages of equal length, m_0 and m_1, and sends them to the Challenger.
- The Challenger chooses a random bit b, produces the challenge ciphertext $c^* = Enc_k(m_b)$, and sends c^* to the Adversary.
- The Adversary may continue to send any message of its choice to the the oracle that computes $Enc_k(\cdot)$. It may also continue to send any ciphertext **other than** c^* to the the oracle that computes $Dec_k(\cdot)$.
- The Adversary outputs $b' = 0$ if it thinks that c^* was the encryption of m_0, and $b' = 1$ otherwise.
- The Adversary wins if $b' = b$.

Prove that AES in CTR mode cannot satisfy the definition of CCA2-secure encryption. In other words, present an algorithm for an Adversary that wins the game described above.

What to submit

1. A Python 2.x script named CTR_solution.py that:

   (a) Takes in two parameters: (a) The path to a file with a ciphertext, and (b) The path to a file with an initialization vector.

   (b) Prints the ciphertext encrypting the message in the original ciphertext +5, assuming that the original ciphertext encrypts a four-digit multiple of 10.

   This script should be callable as follows:

   ```
   ISA656TA$ python CTR_solution.py CTR_ciphertext CTR_iv
   The new ciphertext is 941b3b25eda87cac89af30f78e4cd32e
   ```

2. A file called CTR_writeup.txt that proves that AES in CTR mode cannot satisfy the definition of CCA secure encryption.

Part 5. Galois counter mode

Galois counter mode is a mode of operation that offers both confidentiality and integrity; that is, it is not vulnerable to attacks like the one you executed in CTR_solution.py. You can read more

about Galois counter mode on Wikipedia (https://en.wikipedia.org/wiki/Galois/Counter_Mode). You can also read a blog post explaining why GCM has recently become very popular https://blog.cloudflare.com/padding-oracles-and-the-decline-of-cbc-mode-ciphersuites/.
Explain in a few sentences what you think are the advantages of Galois over the modes of operation discussed above. Are there any disadvantages?

What to submit A file called Galois_writeup.txt that answers the question above.

Part 6. Modes of Encryption in the Wild

In Google chrome, you can see details about the security of a website by clicking on the lock icon to the left of the URL. If the site is secure, it should say "Secure Connection - Your information (for example, passwords or credit card numbers) is private when it is sent to this site." If you click on "Details" right under that statement, you will see some additional information, like what mode of operation is being used in TLS.
Choose 3 websites that offer secure connections and use the AES block cipher. What mode of operation are they using?

What to submit A file called websites.txt naming the websites and the mode of encryption they use.

Submission Checklist for programming part

- Part 2

  1. ECB_image.ppm
  2. CBC_image.ppm
  3. ECB_writeup.txt

- Part 3

  1. CBC_writeup.txt

- Part 4

  1. CTR_solution.py
  2. CTR_writeup.txt

- Part 5

  1. Galois_writeup.txt

- Part 5

1. websites.txt

Acknowledgment

The programming part of this assignment was developed at Boston University for CS558.