IIT Kanpur
Department of CSE
APPLIED CRYPTOGRAPHY

Assignment 1, Semester 2025

Due Date: 13/6/2025

Introduction

This assignment is marked out of 200 marks, and will contribute to 20% final subject marks. It consists of two parts.

Part 1 [70 marks (7%)]: Answer the questions in this file. You are required to type your answer in a separate file and submit as PDF. Handwritten, scanned images, screenshots, and/or Microsoft Word submissions are not acceptable. Note that indicating the Question number and writing your answer are sufficient - do not copy and paste the questions again into your submission.

Part 2 [130 marks (13%)]: Finish the questions in the Jupyter notebook file directly. The Jupyter notebook can be opened offline by setting up the Jupyter Notebook on your local machine (`https://jupyter.org/install`), or online by using Google Colab. You **MUST** use Python 3. It should be fine for any sub-versions of Python 3. However, we recommend you test your implementation under Python 3.10.16, as we will run your submission under this environment. **Before your submission, please rename your Jupyter notebook as {YOUR_STUDENT_NUM}.**

** If you are using Google Colab, some images may not correctly display. You need to re-configure the path to those images. However, you can leave it. Instead, you can inspect the source code in each question body and manually open the corresponding image inside the *img* folder for view.

Part 1 Questions

1. One-time pad (OTP) [20 marks]

   The one-time pad is an amazingly simple, elegant encryption scheme that has theoretically unbreakable security.

   However, it is rarely used in practice. In this question, we will explore some of the reasons why it is not widely used.

   (a) Recall that the one-time pad encryption is defined as $c = p \oplus k$, where p is the plaintext, k is the key, c is the ciphertext, and $k, p, c \in \{0,1\}^n$. Suppose that the key is random, and the key is only used once. Is the one-time pad encryption scheme perfectly secure? Fill in the blanks [5 Marks]

Proof. Recall, we say that a cryptosystem is perfectly secret if

$$Pr(P = p|C = c) = \_\_\_\_\_.(expression)$$

for all possible plaintexts p and all possible ciphertexts c. In other words, by observing the ciphertext, the knowledge you learned about a given plaintext is *ZERO*.

Express $Pr(P = p|C = c)$ using Bayes's theorem, we get

$$Pr(P = p|C = c) = \_\_\_\_\_.(expression)$$

Recall that for a uniformly distributed key k:

$$Pr(C = c|P = p) = Pr(Enc(k, p) = c) = \_\_\_\_\_(expression)$$

$$= \_\_\_\_\_(probability\ expressed\ by\ n)$$

$$Pr(C = c) = \sum\nolimits_{p\in\{0,1\}^n} \_\_\_\_\_(expression)$$

$$= \_\_\_\_\_(probability\ expressed\ by\ n)$$

Simplify the Bayes's expression, we get the proof.

(b) Two-time pad: suppose the encryption algorithm works as Question 1(a), but the key is reused. Now suppose you have intercepted two ciphertexts:

$$c_1 = 00011000000101110000010100000000100001010$$

$$c_2 = 00000111000111010001101100000000100000001$$

You know that the two ciphertexts are encrypted using the same key and **EITHER** c_1 is an encryption of a string "value" and c_2 is an encryption of a string "email" **OR** c_1 is an encryption of a string "hello" and c_2 is an encryption of a string "world". Can you recover the plaintexts? If so, what are the plaintexts? If not, explain why. (All characters are encoded in ASCII code) [5 marks]

(c) There is nothing exclusively special about strings and XOR in the OTP scheme, arithimitic operations can also be used instead of the xor. Suppose n is a prime, we have a plaintext $p \in \mathbb{Z}_n$ and a key $k \in \mathbb{Z}_n^+$ that are both integers. We can encrypt the plaintext by computing $c = pk \bmod n$. [5 marks]

1. Show the corresponding decryption scheme.
2. Is this encryption scheme perfectly secure? Justify your answer.

(d) The daily-life communications are usually meaningful sentences. As a result, sometimes hackers do not even need a key to decrypt a ciphertext. We inherit the encryption scheme from 1(a), and you know *length of* k, $|k| = 28$, and a ciphertext

c=0x221C05471C0E00551B09151D4F171C550B4F164F1301011C1D000E04

6E20646F20666F7220796F752C2061736B207768617420796F

752063616E20646F20666F7220796F757220636F756E7472792E204A464B

Identify the vulnerability of this scheme and try to recover the plaintext and the key. Note that we use ASCII encoding for the plaintext, the ciphertext, and the key. [5 marks]

Hint: The decrypted plaintext is a famous quotation. The encryption scheme is not a block cipher.

The above questions give us a sense of why OTP is not widely used. First, for security, every key can only be used for once. If you are using this scheme to message your friends, you are supposed to manage tons of keys. Second, the key must be as long as the plaintext. Otherwise, we are at risk of reusing the key or even leaving part of the plaintext not encrypted. Third, the key must be truly random.

2. Affine Ciphers [15 marks]

   Consider the following version of a classical cipher where plaintext and ciphertext elements are from $\mathbb{Z}_{28}$. The encryption function, which maps any plaintext p to a ciphertext c, is given by

   $$c = E_{(a,b)}(p) = a(p + b) \bmod 28,$$

   where a and b are from the $\mathbb{Z}_{28}$.

   (a) Derive the decryption function for the scheme. Show your work. [5 marks]

   (b) A key is considered to be *trivial* if $c = p$ for all input p. How many non-trivial keys are possible for this scheme? [5 marks]

   (c) Assume there is a helper which can output the corresponding ciphertext for arbitrary plaintext you supply. Describe an efficient way to retrieve the key using this helper. [5 marks]

3. Cryptanalysis on Monoalphabetic Cipher [15 marks]

   In the lectures, we learned that a brute-force attack on a Monoalphabetic Cipher has a searching space of $N!$ where N is the size of the substitution list. However, such a simple substitution cipher is vulnerable to a language statistics analysis. In this question, you are asked to find the **decryption key** (the substitution list) for a given ciphertext encrypted using Monoalphabetic Cipher.

   You are given the following:

   i. A ciphertext file 'ciphetext.txt' which contains approximately 500 words encrypted from a plaintext using Monoalphabetic Cipher. The original plaintext is taken from an Enlish book. All characters in both plaintext and ciphertext are lower-cased.

ii. The knowledge that the encryption key contains 26 lower-cased letters in English alphabet ('a' to 'z') plus the blank space ' ', the Comma ',' and the Period '.'. Othe special characters (for example: numbers é - , etc.), in the plaintext are kept un-substituted.

For example, 'light-thinking' may be decrypted as 'xgebr-rbgkjgke'.

You are required to write detailed and reasonable analysis steps for inferring each letter of the key. Any solution without a reasonable explanation **WILL NOT** gain any marks for this question, even if the key is correct. If any of the letters never appear in the ciphertext, please note them in your solution.

This question explains why it is dangerous to leave a statistical relationship between the plaintext and ciphertext exposed while designing a cryptography system.

4. A practical content delivery encryption system. [20 marks]

A game company tends to protect game content delivery on PS/Switch/Xbox through DVDs. Here is one possible approach. Suppose there are at most a total of n consoles in the world (e.g. $n = 2^{32}$). We view these n consoles as the leaves of a binary tree with height $\log_2 n$. Every node v_j in this binary tree contains a key $k_j \in K$. These keys are kept secret from consumers and are fixed for all time. At manufacturing time every console is assigned a serial number $i \in \{0, \cdots, n-1\}$. Let S_i be the set of $1 + \log_2 n$ nodes along the path from the root of the binary tree to leaf number i. The manufacturer embeds in player number i the $1 + \log_2 n$ keys, which is its unique key associated with the keys in nodes in S_i. In this way, each console ships with $1 + \log_2 n$ keys embedded in it, and these keys are supposedly inaccessible to the end user. Ideally, a game m is encrypted as

$$Console \ := \underbrace{E(k_{root}, k)}_{header} || \underbrace{E(k, m)}_{body}$$

where $E(key, message)$ is an encryption scheme, $||$ denotes string concatenation, and $k \xleftarrow{R} \mathcal{K}$ is a fresh random key called a content key (you can think key k is fully random and unique for different m simply). Since all consoles have the key k_{root}, all consoles can decrypt the content m. We refer to $E(k_{root}, k)$ as the header and $E(k, m)$ as the body. In what follows the console header may contain multiple ciphertexts where each ciphertext is the encryption of the content k under some key k_j in the binary tree. That's because if some consoles are hacked, the industry can use keys $\{k_j\}$ in the binary tree to encrypt a newly released game to revoke access to this game of the hacked console. Let's see some examples.

(a) Let's say that $1 + \log_2 n$ keys embedded in console number i are exposed by hackers and disclosed to the public. Show that when a new game m is about to release (Baldur's Gate 3 for example), m can be encrypted by using a header containing $\log_2 n$ short ciphertexts so that all consoles can decrypt the game m

except for console number i. In effect, the industry disables the game for console number i. [10 marks]

[Hints] The header will have $\log_2 n$ ciphertexts where each ciphertext is the encryption of the content key k under certain $\log_2 n$ keys from the binary tree.

(b) Now suppose the keys embedded in s consoles, $I = \{i_0, \cdots, i_{s-1}\}$, which are exposed by hackers, where $s > 1$. At this time, the industry needs to ban all the consoles in the console set I from decrypting the game. Show a way that the industry can encrypt the contents of a new game using a header containing $O(s \log_2 n)$ short ciphertexts so that all the consoles can decrypt the game except for the console set I. [10 marks]

[Hints] What you just did is that all hacked devices can be revoked without affecting other consumers.

Part 2 Questions

Please finish the Part 2 questions in the Jupyter Notebook '2025asg1.ipynb'. Do not forget to rename the file with your student number before the submission.

Submission and Evaluation

- You must submit a PDF document for the part one, and the Jupyter notebook for the part two, via the Assignment 1 submission entry on the LMS by the due date. Handwritten, scanned images, screenshots, and/or Microsoft Word submissions are not acceptable — if you use Word, create a PDF version for submission.
- Late submission will be possible, but a late submission will attract a penalty of 10% per day (or part thereof). Requests for extensions on medical grounds will need to be supported by a medical certificate. Any request received less than 48 hours before the assessment date (or after the date) will generally not be accepted except in the most extreme circumstances.
- This assignment will be marked out of 200 marks, and will contribute to 20% of your total marks in this subject. Marks are primarily allocated for correctness of your thinking and clarity of your communication, rather than (only) the correct result without justification.
- We expect your work to be neat — parts of your submission that are difficult to read or decipher will be deemed incorrect. Make sure that you have enough time towards the end of the assignment to present your solutions carefully. Time you put in early will usually turn out to be more productive than a last-minute effort.
- You are reminded that your submission for this assignment is to be your own individual work. For many students, discussions with friends will form a natural part of the undertaking of the assignment work. However, it is still an individual task. You are welcome to discuss strategies to answer the questions, but not to share the work (even draft solutions) on social media or discussion board. It is University policy that cheating by students in any form is not permitted, and that work submitted for assessment purposes must be the independent work of the student concerned.

If you have any questions, you are welcome to reach out the teaching team.