MATH 4440/5440 Assessment, Module 1 (Classical Cryptography, Introduction to Modular Arithmetic)

Katherine Stange, CU Boulder, Fall 2020

Honor Code Rules

Assessments are open book, but are to be completed on your own without collaboration. To be specific, you may use your course notes, textbook, course website resources, course videos. You may not use the internet beyond the course websites. You may not ask anyone else for help (except your professor), including other humans, or posting/entering your question or any part of it into the internet. You may not share the questions or answers with anyone else. You may not use calculators (even from the course websites) unless explicitly permitted in the question. You may use the Crypto Tools Sheet (the Vigenére square and multiplication table, included here as a final page) for the entire assessment.

Have you read, understood, and followed the honor code rules above?

$$YES \quad / \quad NO$$

Some instructions on formatting.

You may use the accompanying LaTeX source document to produce LaTeX'ed answers. You may typeset answers separately. You may print the pages and solve the questions on them by hand. You may handwrite answers on separate sheets. You may upload PDF or image files (JPG or PNG). No matter what you do, just make sure it is clearly and easily legible before you upload it to canvas.

1 Question 1

This question asks for some encryptions and decryptions and similar. You must do these by hand, not using any online calculators (even from our website). You may use the Crypto Tools Sheet (the Vigenére square and multiplication table). For full credit, show your work and explain the method. You don't need to explain why the method works. And you don't need to write a novel, but, for example, if you are adding two things modulo 26, say what you are adding!

1. (10 points) Encrypt the following using Vigenère cipher, with key SHERLOCK:

   THE GAME IS AFOOT.

   You should remove/ignore the spaces.

2. (10 points) Here is a key for an ADFGVX cipher: the Polybius square shown, and column key 623541.

	A	D	F	G	V	X
A	a	o	4	h	f	8
D	n	q	1	r	p	d
F	k	m	3	c	z	9
G	j	l	v	b	6	0
V	5	s	i	t	x	u
X	e	w	7	y	2	g

Decrypt the ciphertext

AAD XGA DGA GXA FAG XDG ADV DAV VVG AXF DVD VVX DGG ADF FFD VXA VDA XGV

(Note: the spacing is just there to help you not skip letters by accident.)

3. (10 points) Decrypt the Hill ciphertext XOYUJW, which was encrypted using the encryption matrix

$$\begin{pmatrix} 3 & 7 \\ 8 & 3 \end{pmatrix}.$$

4. (10 points) Find the one-time pad key which will decrypt the ciphertext XYZA to the plaintext GOOD.

Solutions.
These solutions don't show work but do show the final result. Work is needed for full credit.

1. LOIXLAGSKHJFZH
2. TOAGREATMINDNOTHINGISLITTLE (Another Sherlock Holmes quote)
3. PICKLE
4. RKLX

2 Question 2

1. (10 points) Compute $30000000001 \cdot 17^3 + 56 \pmod 5$. Do this by hand and show your steps (it should not be laborious).
2. (10 points) Determine $(\mathbb{Z}/6\mathbb{Z})^*$ (the unit group). You may not use any calculators. Work from first principles (don't state a theorem for justification unless proving it). Show your work.

Solutions.

1. Here's one possible method (there are others):

$$30000000001 \cdot 17^3 + 56 \equiv 1 \cdot 2^3 + 1 \equiv 8 + 1 \equiv 4 \pmod 5$$

2. One method is to write out a multiplication table for $\mathbb{Z}/6\mathbb{Z}$ and examine the columns. Later in the course we will meet more sophisticated methods. The answer is $\{1, 5\}$.

3 Question 3

1. (5 points) State the keyspace for affine cipher.
2. (5 points) Explain why $(\alpha, \beta) = (2, 10)$ is not in the keyspace.
3. (10 points) Suppose we tried to do an affine cipher with the key $(\alpha, \beta) = (2, 10)$, even though it is not a valid key. Give an example of two plaintexts which encrypt to the same ciphertext under this system. State the two plaintexts and the resulting ciphertext.
4. (10 points) How many such pairs of plaintexts of length n are there? In other words, how many pairs of plaintexts p and p' of length n are there whose corresponding ciphertexts c and c' satisfy $c = c'$? Justify (I will consider anything you've written in the previous part as part of the justification, and you may receive partial credit for that even if your numerical answer is incorrect.)
5. (10 points) Suppose we use key $(\alpha, \beta) = (5, 3)$ for affine cipher (this is a valid key). Prove that every finite string of characters modulo 26 is the ciphertext to exactly one plaintext (no less and no more).

Solution.

1. The keyspace is $(\mathbb{Z}/26\mathbb{Z})^* \times (\mathbb{Z}/26\mathbb{Z})$.
2. Since α is not invertible, it is not in $(\mathbb{Z}/26\mathbb{Z})^*$.
3. The encryption map is $p \mapsto 2p + 10 \pmod{26}$. In order to have two plaintexts encrypting to the same ciphertext, we must solve
$$2p_1 + 10 \equiv 2p_2 + 10 \pmod{26}.$$
This becomes
$$2(p_1 - p_2) \equiv 0 \pmod{26}.$$
Using the multiplication-by-2 column of the multiplication table, we can compute
$$p_2 - p_2 \equiv 0, 13 \pmod{26}.$$
Therefore, any pair of plaintext letters which differ by 13 will encrypt to the same ciphertext letter. (Conversely, if the letters map to the same ciphertext letter, they are the same or differ by 13.) An example is A (0) and N (13), which both encrypt to K (10). There are many other correct answers.
4. This problem was unintentionally harder than expected, so I made the denominator of the exam 110 instead of 115; in a sense this is half bonus. The justification depends upon the justifications in the previous part. There are 13 such length-one plaintext pairs, since all 26 plaintexts are paired: k is paired with $k+13$. The complication comes when we have n characters.
One approach is to think of the map from plaintexts to valid ciphertexts:
$$e : \mathcal{P} \to \mathcal{C}$$
We have seen that there are only 13^n valid ciphertexts of length n (since there are 13 possible letters in each position), so
$$|\mathcal{C}| = 13^n.$$

We also know the size of the space of plaintexts, since it allows every alphabetic string:

$$|\mathcal{P}| = 26^n.$$

My next goal is to calculate the pre-image of every ciphertext. If p is a plaintext that maps to a ciphertext c, then I can form all the other plaintexts that map to the same ciphertext by changing or not changing each letter. Since that's a binary choice for each letter, and the choice of not changing anything leaves p alone, we find that there are 2^n possible plaintexts that map to the ciphertext c. (Quick check: $2^n \cdot 13^n = 26^n$, verifying that, adding up the sizes of all the preimages, we get the entire plaintext space.)

Ok, so now how many pairs map to the same place? Well if 2^n things map to c, there are $\binom{2^n}{2}$ pairs mapping to c. We have to multiply this by the number of possible c. The final answer is

$$\binom{2^n}{2} 13^n.$$

5. Let c be a candidate ciphertext of length one, i.e. an element of $\mathbb{Z}/26\mathbb{Z}$. The corresponding plaintext letters p are all solutions to the equation $5p + 3 \equiv c \pmod{26}$, which is the same as $5p \equiv c - 3 \pmod{26}$. As multiplication by 5 is a bijection (the table verifies this), there is exactly one solution to this equation. Therefore we have proven the required statement for ciphertexts of length exactly one. For a longer string, the observation above holds letter-by-letter, giving exactly one solution to the entire ciphertext.

4 Question 4

Suppose I have a randomly chosen english text, call it T, of length 10000 characters. It may be obtained, for example, by taking a random excerpt of Charles Dickens (or another standard english author). I encrypt the text using Caesar cipher using each key i (for i from 0 to 25), to obtain 26 texts, T_i. To reiterate, the texts T_i, for i from 0 to 25, are all the possible Caesar encryptions of T.

Suppose I also have a Caesar ciphertext of length 10000, call it C, which you know is an encryption of english plaintext. The corresponding plaintext is not related to T in any particular way.

Suppose you are only given the *coincidence counts* between C and each T_i for all $0 \leq i \leq 25$. (By *coincidence count* I mean the number of positions at which C and T_i agree.) That is, you are given 26 integers.

1. (5 points) What pattern will you see in the coincidence counts and how will this pattern alone allow you to guess the key used for the Caesar ciphertext C?

2. (10 points) Why does this work? Give a carefully explained justification from first principles (i.e. depending only on basic concepts in probability and the geometry of vectors).

Solution.

1. The count will be noticeably higher for i equal to the Caesar key shift of ciphertext C. For example, if C was encrypted with a key of 3, then the coincidence count will be particularly high for C with T_3. Thus you should take the highest coincidence count and guess that that i is the key.

2. Write $\mathbf{v}_n$ for the standard english frequencey distribution, shifted by n. This is a vector with 26 entries, where the j-th entry represents the probability of the j-th letter of the alphabet. The letters of C follow a the

distribution $\mathbf{v}_k$. The letters of T_i follow the distribution $\mathbf{v}_i$. To determine the probability of a coincidence between randomly chosen letters of C and T_i, we have

$$\begin{aligned}\text{Prob(letters agree)} &= \sum_{j=0}^{25} \text{Prob(letters are both the } j\text{-th letter of the alphabet)} \\ &= \sum_{j=0}^{25} (\mathbf{v}_k)_j (\mathbf{v}_i)_j \\ &= \mathbf{v}_k \cdot \mathbf{v}_i\end{aligned}$$

Therefore, the largest probability occurs when the dot product is maximized. As the vectors $\mathbf{v}_i$ all have the same entries in different orders, they are all the same length. Therefore,

$$\mathbf{v}_k \cdot \mathbf{v}_i = |\mathbf{v}_k||\mathbf{v}_i| \cos\theta = |\mathbf{v}_k|^2 \cos\theta$$

where θ is the angle between the vectors. This is maximal when they point in the same direction, i.e. $\mathbf{v}_k = \mathbf{v}_i$, i.e. $k = i$. Therefore the greatest probability of a coincidence occurs when $i = k$. The coincidence counts we are given will approximately follow the probabilities just calculated. With long texts (length 10000), this will very likely result in the highest count being when $k = i$.