Shweta Prasad February 19, 2021

CS 2362 Problem Set 1

Collaborators: *none*

Problem 1-1

(a)

Whether Anthony will in fact meet Ceasar depends on how patient and/or lucky he is. `rot 13` decryption suggests that the plaintext is "`RIVER`" while `rot 22` decryption suggests that the plaintext is "`ARENA`". Anthony will have to toss a coin to decide where to go if he doesn't jump the gun at `rot 13` and go to Tiber.

(b)

Second panel: "`programmer`", Last panel: "`everytimeyousaybacklashaspartofawebaddress onairidiealittle`";

Decryption done by shifting each character by 7 characters.
Code to go with this named `ceasar.py`.

(c)

Shift : 10 characters
Plaintext: "`youmustbespeedoflightbecausetimestopswhenilookatyouhappyvalentinesday`"
(*thanks*)

(d)

Shift : 21 characters
Plaintext: "`wikipediaisthebestthingeveranyoneintheworldcanwriteanythingtheywantabout anysubjectsoyouknowyouaregettingthebestpossibleinformation`"

(e)

Shift : 7 characters
Plaintext: "`olddebayanisagoodoneexceptformelonmelonmelonmelonmelon`"
(*sir,what?*)

Problem 1-2

Given: the affine funciton used to encrypt the plaintext is $17x + 5(\mod 26)$.

To decrypt an affine cipher of the form $c = \alpha x + \beta(\mod p)$, we use the function $D(c) = \alpha^{-1}(c - \beta)(\mod p)$. α and p must be co-prime, which in this case, they are. We calculate that the modular inverse of $17(\mod 26)$ is 23.

Using this method, the plaintext "`NULNFSVIX`" can be decrypted to read "`CHICANERY`", which indeed, it was.

Code to go with this in `affine.py`

Problem 1-3

Mallory will be able to discover what message Bob sent to Alice. He discovers that the two will play Scrabble on Friday.

Mallory has knowledge that Bob and Alice are using an affine cipher to encrypt their messages. He knows that the underlying language of the plaintext and ciphertext is English, and can knows that the encryption and decryption functions will be in the mod 26 world. He has also intercepted a plaintext−ciphertext pair by means of which he can calculate the key to the cipher, as follows:

Mallory lists the English alphabets in order and numbers them from $0 - 25$ respectively. He then uses the known plaintext to deduce what at least 2 distinct alphabets have been substituted by, and writes the corresponding numbers against them. `CHESS` is represented as `LARHH` in this system, and working out the representative numbers backwards, we can write

$$num(L) = num(C)\alpha + \beta \implies 11 = 2\alpha + \beta$$

$$num(R) = num(E)\alpha + \beta \implies 17 = 4\alpha + \beta$$

Solving for α and β, we get

$$\alpha = 3$$
$$\beta = 5$$

Decrypting in the same manner as in problem 1-2, Mallory will be able to see that the message Bob sent to Alice was “`LETUSPLAYSCRABBLE`”.

Code associated with this problem is the same as last problem, `affine.py`.

Problem 1-4

(a)

A	C	G	H	O	P	P	R	R	T	Y	Y
s	s	m	w	i	e	i	o	e	t	m	i
s	l	t	n	r	t	e	l	a	a	s	t
d	e	d	n	n	e	o	n	i	a	c	t
w	e	o	e	n	n	h	v	w	k	e	r
g	e	i	j	o	s	i	i	n	g	t	u
i	s	i	d	e	o	n	t	f	p	h	i
e	t	t	a	g	o	w	a	h	n	l	y

Above is the table before decrypting. To decrypt, arrange the columns so that the leading row reads `‘CRYTPTOGRAPHY’`. It will look something like this (apologies there was some kind of error when I was manually filling it in and was unable to find it in the hurry to submit, I hope you get the gist of my answer, because I *have* implemented it based on this algorithm and my implementation works)

C	R	Y	P	T	O	G	R	A	P	H	Y
s	o	m	e	t	i	m	e	s	i	w	i
l	l	s	t	a	r	t	a	s	e	n	t
e	n	c	e	a	n	d	i	d	o	n	t
e	v	e	n	k	n	o	w	w	h	e	r
e	i	t	i	s	g	o	i	n	g	i	j
u	s	t	h	o	p	e	i	f	i	n	d
i	t	a	l	o	n	g	t	h	e	w	a

(b)

Code has been submitted and appropriate naming has been done. The current decrypted text file contains the result for the ciphertext file given in the original zip file. If you run the encrypt function on the ciphertext in that file, you will get the correct answer, which is also working proof for the fact that the encrypt and decrypt functions are inverses of each other.

Problem 1-5

(a)

The given plaintext-ciphertext vectors can be shown as related to each other in the following manner:

$$MX = C \mod 26$$

where M is the matrix used for encryption, X is the plaintext vector and C is the ciphertext vector.

Given $A = 0, B = 1...$; Substituting the alphabets in the respective pairs of vectors henceforth:

$$X_1 = \begin{bmatrix} B \\ F \\ T \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 19 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} H \\ I \\ Y \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \\ 24 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} D \\ Q \\ V \end{bmatrix} = \begin{bmatrix} 3 \\ 16 \\ 21 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} Q \\ T \\ X \end{bmatrix} = \begin{bmatrix} 16 \\ 19 \\ 23 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} E \\ X \\ Z \end{bmatrix} = \begin{bmatrix} 4 \\ 23 \\ 25 \end{bmatrix}$$

$$C_3 = \begin{bmatrix} U \\ N \\ M \end{bmatrix} = \begin{bmatrix} 20 \\ 13 \\ 12 \end{bmatrix}$$

Let

$$M = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Using the known plaintext-ciphertext pairs, we can write

$$MX_1 = C_1$$
$$MX_2 = C_2$$
$$MX_3 = C_3$$

All calculations are in mod 26. Expanding the above systems into equations, we get the following 9 equations:

$$\begin{aligned} a + 5b + 19c &= 7 \\ d + 5e + 19f &= 8 \\ g + 5h + 19i &= 24 \\ \\ 3a + 16b + 21c &= 16 \\ 3d + 16e + 21f &= 19 \\ 3g + 16h + 21i &= 23 \\ \\ 4a + 23b + 25c &= 20 \\ 4d + 23e + 25f &= 13 \\ 4g + 23h + 25i &= 12 \end{aligned}$$

Which can be re-grouped as follows:

$$\begin{aligned}a + 5b + 19c &= 7\\ 3a + 16b + 21c &= 16\\ 4a + 23b + 25c &= 20\\ \\ d + 5e + 19f &= 8\\ 3d + 16e + 21f &= 19\\ 4d + 23e + 25f &= 13\\ \\ g + 5h + 19i &= 24\\ 3g + 16h + 21i &= 23\\ 4g + 23h + 25i &= 12\end{aligned}$$

Solving these equations, we get

$$\begin{bmatrix}a\\b\\c\end{bmatrix} = \begin{bmatrix}3\\9\\17\end{bmatrix};\begin{bmatrix}d\\e\\f\end{bmatrix} = \begin{bmatrix}5\\13\\20\end{bmatrix};\begin{bmatrix}g\\h\\i\end{bmatrix} = \begin{bmatrix}8\\25\\23\end{bmatrix}$$

Therefore, we have that

$$\text{M} = \begin{bmatrix}a & b & c\\ d & e & f\\ g & h & i\end{bmatrix} = \begin{bmatrix}3 & 9 & 17\\ 5 & 13 & 20\\ 8 & 25 & 23\end{bmatrix}$$

□

(b)

No, the given matrix cannot be used for encryption as it is not invertible in mod 26. $104\%26 = 0$, therefore the matrix is singular, and has no inverse.

Problem 1-6

(a)

Given : $M = \{5, 6, 7, 8, 9\}$, $K = \{0, 1, 2\}$; the probability $P[M = m]$ for $m \epsilon M$ is respectively given as $\{1/3, 1/3, 1/6, 1/12, 1/12\}$ and the keys are drawn from a uniform distribution.

Assumption: the only messages in the language are the given and shifting 9 using a key of 1 will give a ciphertext of 5, and so on (a mod 5 world).

Under the given conditions, Alice will not be able to find a perfectly secure encryption scheme, and this is because the space of keys is smaller than the space of messages. For perfect secrecy, the condition is that the message and its ciphertext in the scheme have to be statistically independent. That is,

$$P[m] = P[m|c] = \frac{P[(M = m) \cap (C = c)]}{P[C]}$$

If this is shown untrue for any pair of m, c the system is not perfectly secure.

Let's say that Adam sees a $c = 5$, and wants to know the probability that the message was $m = 8$.

$$P[C = 5] = P[M = 9] \cdot P[K = 1] + P[M = 8] \cdot P[K = 2]$$

$$P[M = 8] = \frac{1}{12}(given)$$

$$P[(M = 8) \cap (C = 5)] = P[M = 8] \cdot P[K = 2] = \frac{1}{36}$$

Therefore, we have

$$P[M = 8|C = 5] = \frac{1/36}{1/18} = \frac{1}{2} \neq \frac{1}{12} = P[M = 8]$$

Which proves that Alice's method will not allow perfect security.

(b)

In Alice's new language, the probability of all $m \ \epsilon \ M = 1/5$, is given. The previous system will NOT be secure even in such a scenario. However, if we can add at least 2 more keys to her system, we will be able to build a perfectly secure cryptosystem, as long as the probabilities of the messages in M totally sum to 1. Proof in the next section.

(c)

Let K be expanded to the set $\{0, 1, 2, 3, 4\}$ and they are still all equally likely to be used.

Now, for the above case where the message space also forms a uniform distribution, the probability of the message being m given any ciphertext c will be

$$P[M = m|C = c] = \frac{P[(M = m) \cap (C = c)]}{P[C]} = \frac{P[M = m] \cdot P[K = k_1]}{P[C]}$$

where k_1 is the appropriate key for c. This will be

$$P[M = m|C = c] = \frac{1/5 \cdot 1/5}{5 \cdot 1/5 \cdot 1/5} = \frac{1}{5} = P[M = m]$$

One can thus observe that if $|M| > |K|$ then the system cannot be perfectly secure. There should be a one-to-one mapping from the message space to the keyspace and from the keyspace to the ciphertext space: effectively a one-to-one mapping from message space to ciphertext space via keys. This exercise led me to Shannon's Theory of Perfect Secrecy, and helped understand its conclusion that one time pads are perfectly secure and any system that aims to be perfectly (information theoretically) secure must have the same properties.

Problem 1-7

Coeve sends a message with multiple repeats of the plaintext a, in the manner $aaaa..$; Geith knows what cryptosystem Coeve is using to communicate with him. Geith also knows the ciphertext corresponding to this message Coeve has sent, but does not know the key so cannot readily decrypt the it. However, since he knows what cryptosystem is being used, he may be able to deduce the key and decrypt messages. There are 2 cases here:

- Case 1. Geith knows that a is the repeating character
- Case 2. Geith does not know that a is the repeating character

(a)

- **For Case 1.** If the system is a shift cipher, the intercepted message will have a sequence of repeat values in the ciphertext, corresponding to the sequence of a's. Assuming a mod 26 alphabet, we can calculate the distance between the ciphertext (c) representing a, and a, from the equation $c = p + k \pmod{26}$ where p represents the plaintext. Once k has been figured out, Geith can completely decrypt the message sent.

- **For Case 2.** Geith will not be able to figure out what the message is, because there are 26 possible keys and the repeat sequence will not be of any help in deducing the key. He will need some additional information to deduce which is the actual key for the shift.

(b)

- **For Case 1.** If the system is an affine cipher, there are two subkeys to be deduced. If the sent message has only a sequence of a's, Geith will not be able to deduce the subcomponents of the key as he can only form one equation. Additionally, multiplication of the α params with 0, which represents the letter a, will also prove unwieldy to deduce the key from any equation formed. However, if the message looks like so: *...aaaa...aab..*, the junction between a and b will also be detectable in the corresponding ciphertext. From that, Geith can identify 2 plaintext-ciphertext pairs to form 2 equations of the form $c = \alpha p + \beta \pmod{26}$ and solve for α and β.

- **For Case 2.** Geith will not be able to deduce the key or understand the message because he does not have a pair of known plaintext characters.

(c)

- **For Case 1.** If the cryptosystem is a Hill cipher, Geith will be unable to use a in any way to figure out the exact key matrix because the number that represents a is 0. However, like in the second case in part (b), he can use the repeat sequence of a to identify junctions. Since Geith knows that the cipher transforms digraphs, he can split both the plaintext and ciphertext in twos, and identify 2 non-a digraph plain-ciphertext pairs, and form a solution (find the 4 elements of the key matrix). Using this, he will be able to decrypt the message Coeve sent to him. (Here, if the plaintext has at least 2 digraphs that have no a's in them, it can be cracked.)

- **For Case 2.** Without a known pair of plain-text-ciphertext digram pairs, Geith will not be able to break this cipher.

Problem 1-8

Submitted in appropriate files. Code used is in `vigenere.py`.

Problem 1-9

Briefly, the cryptosystem proposed here is a Feistel network similar to DES, whose output will be fed into an expansion permutation function that gives an expanded output, which is then passed into a simple affine cipher.

Let the input size be $2m$ bits where m is a multiple of 8, and the round keys generated from a master key be of size n such that $m < n < 2m$. We divide the input into halves, L, R. An initial permutation is applied to the input. Let there be r rounds to this Feistel network and at each round, to increase confusion and diffusion, we apply the Feistel function twice on R_{r_i}, using keys generated from 2 separate master keys M_1 and M_2. Both subkeys are added to the key schedule in some order, say 1 then 2, and that order is maintained at each round. After application of the Feistal function twice, the half block is XORed with the other half of the block (: L_{r_i}) after which it becomes the left half of the input to the next round. After r such rounds, the output is still of size $2m$. As in DES, this output is subjected to a final permutation, which 'undoes' the initial permutation. After this, this output is split into $4-bit$ blocks and there will be a total $m/2$ such blocks, each of which is expanded into $5-bit$ blocks using some secure and tested coding scheme, and the scheme is assumed to be known only to the users of this cryptosystem. The output will have $5m/2$ bits in total. This is then subjected to an affine substitution, where to effectively make it a one time pad, the function is decided at every encryption.

To decrypt, one just has to use the keys in the reverse order: subkeys for the affine cipher, then the lookup table for the expansion coding, then feed this into the Feistel network from the top, but this time, with a reversed key schedule.

I cannot guarantee that this system will provide much security as I have not proved it, however, I have deliberately tried to increase the properties of diffusion and confusion as proposed by Shannon.

Problem 1-10

(a)

The key given is `PLAYFAIRSUCKS`, which can be put into a 5×5 table as follows:

Split the ciphertext into digrams as follows: `TN DY WI XH OD MB SK KG` Now we can use the table and take one digram a time and look at the rectangle formed by the digram in the key table, then substitute it according to the rules mentioned in class. The substitution will

P	L	A	Y	F
I/J	R	S	U	C
K	B	D	E	G
H	M	N	O	Q
T	V	W	X	Z

be done as follows:

```
TN DY WI XH OD MB SK KG
WH EA TS TO NE BR ID GE
```

Thus, the plaintext has the message `WHEATSTONE BRIDGE`.

(b)

Without a key, one can attempt to break the Playfair cipher using frequency analysis of digrams. This is analogous to what one can do when trying to break a letter substitution cipher: find the frequency distribution of the underlying alphabet (of both the plaintext and ciphertext, assumed to be the same); given the ciphertext discovered is amply large (a is a representative sample), one can perform a frequency analysis of the letters used in it and compare it to the frequency distribution of the letters in the underlying alphabet. Matching up the frequencies, one can deduce what each letter is substituted by in the ciphertext, and easily decrypt any message sent encrypted using this cipher. Similarly, one can make use of a digram frequency distribution table for the underlying language, and conduct a frequency attack on the Playfair cipher. This method can ideally be used for any n-gram substitution cipher.