Aniruddha Jafa February 14, 2019

CS-302 Problem Set 1

Collaborators: *Paul Kurian*

Problem 1

'evire' decrypts to both 'arena' (shift of 4) and 'river' (shift of 13). If Anthony tries all 26 possibilities, he will observe this, and be confused about where to meet Caesar. One way for them to resolve this ambiguity would be to decide beforehand that in case of ambiguities, the smaller shift will be preferred.

Problem 2: Affine Cipher

Encryption is c $\equiv_{26}$ 9m + 2, where c is the cipher text.
Thus decryption is m $\equiv_{26}$ $9^{-1}(c-2)$

We know that $9^{-1} \equiv_{26} 3$

Thus decryption is m $\equiv_{26}$ $3(c-2)$

Our given ciphertexts are U, C and R (20, 2 and 17 mod 26 respectively).

Plugging in 20, 2, and 17 respectively into m $\equiv_{26}$ $3(c-2)$, we get:

Decryption(20) $\equiv_{26}$ 3(20 - 2) $\equiv_{26}$ 2 = C
Decryption(2) $\equiv_{26}$ 3(2 - 2) $\equiv_{26}$ 0 = A
Decryption(17) $\equiv_{26}$ 3(17 - 2) $\equiv_{26}$ 19 = T

Thus UCR decrypts to CAT.

Problem 3: Hill Cipher

Let the matrix used in the Hill Cipher be denoted by $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

The information given in the problem tells us the following:
$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \end{bmatrix}$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 25 \\ 25 \end{bmatrix} = \begin{bmatrix} 6 \\ 19 \end{bmatrix}$

On writing out the equations obtained from the first set of matrix multiplications, we get,

a + 0.b $\equiv_{26}$ 7, i.e. a $\equiv_{26}$ 7
c + 0.d $\equiv_{26}$ 2, i.e. c $\equiv_{26}$ 2

Using the values of a and c in the second set of matrix multiplication, we get

25(7+ b) $\equiv_{26}$ 6
25(2 + d) $\equiv_{26}$ 19

Since gcd(25,26) = 1, 25 has an inverse mod 26, and in fact this inverse is 25 itself. We can thus multiply both sides of the above 2 equations by 25 to solve for b and d.

Using the above reasoning and solving above for c and b, we get b $\equiv_{26}$ 13, and d $\equiv_{26}$ 5.

Hence $M = \begin{bmatrix} 7 & 13 \\ 2 & 5 \end{bmatrix}$

Problem 4

Let C be the set of ciphertexts, K be the set of keys, M be the set of messages.

(a) For Alice, $M = \{5, 6, 7, 8, 9\}$, $K = \{0, 1, 2\}$. Regardless of encryption scheme, perfect secrecy is not possible if $|M| > |K|$.

Why?

To show: If $|K| < |M|$, then perfect secrecy is not possible i.e. $\exists$ some $c \in C$ such that

$P(m|c) \neq P(m)$

We are given $|K| < |M|$ —- (1)

We know for any encryption scheme, $|C| \geq |M|$, since if $|C| < |M|$decryption would not be possible (multiple messages would be mapped to the same ciphertext by the encryption function, and there would be no way of decrypting). ———- (2)

From (1) and (2) we have $|K| < |M| \leq |C|$ ——— (3)

Assume there exists a message m in M which is being sent with a positive probability i.e. $P(m) > 0$

For message m, with $|K|$ keys, m can be mapped to at most $|K|$ different ciphertexts are possible (in the best case that each key leads to a unique ciphertext).

But from (3) since $|K| < |C|$, necessarily $\exists$ some c$\in$ $|C|$ such that m does not map to it (since the number of cipher texts m can map to is strictly less than the number of ciphertexts).

Thus, for c, $P(m|c) = 0$. But we had picked m so that $P(m) > 0$. So we have shown $\exists$ some $c \in C$ such that $P(m|c) \neq P(m)$

This finishes the proof.

(b) By the proof given in part (a), as long as $|K| < |M|$, perfect secrecy is not possible; the exact probabilities assigned to different messages is irrelevant to perfect secrecy. So the fact that Alice makes her messages equiprobable is useless unless her key space is at least as large as the message space.

Yes, it is possible to to modify the system Alice is using to make it perfectly secure.

NEW SYSTEM: Consider Alice's message space M = $\{5, 6, 7, 8, 9\}$. From the description of part (b) ("In this new language, all her messages are the same, but their probabilities are equal"), we know Alice sends each message with equal probability i.e. P(m) = 1/5 $\forall$ m. Pick keyspace K' = $\{0, 1, 2, 3, 4\}$. For sake of simplicity, as done is the slides, assume each key is used with an equal probability. Encryption is c = m + k (mod 5). Clearly, C = $\{0, 1, 2, 3, 4\}$

(c) Is the above (simple) system perfectly secure?

We need to show $P(m) = P(m|c)$ for all m and c.

We know P(m) = 1/5. Given any c, could any possible m have led to it ?

We can easily see that every message in $\{5, 6, 7, 8, 9\}$ is mapped to each c in $\{0, 1, 2, 3, 4\}$ exactly once under the encryption scheme. Thus P(c and m) = 1/25 for all m and all c.

Hence $P(c|m) = P(candm)/P(m) = \frac{1/25}{1/5} = 1/5 = P(m)$

This system is perfectly secure.

:: What these exercises reveal about perfect security:
A necessary condition is $|K| \geq |M|$
Exact probabilities of messages don't matter much.

Problem 5

(a) For a shift cipher, all Zark will see is a string of repeated characters. Since each 'a' will be encrypted to the same letter (based on the shift chosen), Zark can detect that the message consists of a repeated character. But he has no means to guessing the shift (key), and cannot determine what the original character is since multiple combinations of keys and characters could have led to the ciphertext he intercetps.

(b) For an Affince cipher, encryption('a') = p.m + q (mod 26) ≡ p.0 + q (mod 26) ≡ q, since q is an integer from 0...25.

Hence, Zark can identify that the same character is being sent - hee sees a bunch of q's. But he has no way of knowing if character 'a' is being sent, since other combinations p'.m' + q' could also be congruent to q mod 26. e.g. if q = 7, put p' = 3, m' = b = 1, q = 4. p'.m' + q' $\equiv_{26}$ 3.1 + 4 $\equiv_{26}$ 7 = q.

(c) Consider a Hill cipher, say with key $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
To encrypt repeated 'a' characters, since 'a' is represented an 0 mod 26, the following matrix product is computed:

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

So Zark intercepts a string of 0's. Clearly one way this can happen is when the message is of "a" characters are being encrypted, in which case the key cannot be known (due to multiplciation by 0). So Zark makes note of this

Can any other encryption of non-consecutive ”a” characters and keys lead to Zack observing a string of 0’s ?

Consider an arbitrary key Q = $\begin{bmatrix} p & q \\ r & s \end{bmatrix}$
Take the following encryption:
$$\begin{bmatrix} p & q \\ r & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
where at least one of x or y is non-zero (i.e. doesn’t represent an ’a’).WLOG say x is definitely non-zero. We get

$$\begin{bmatrix} px + qy \\ rx + qy \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

If at least one of x or y is non-zero, and since all of p, q, r, s, x and y are non-negative, the only way to obtain $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is if the matrix Q is non-singular i.e. at minimum you either need 2 zeroes in a column of Q, or two zeroes in a row of Q (n which case the determinant would be 0). But this cannot be the case is Q is being used as a key, since matrices used for keys must be non-singular.

Hence, Zark knows the message is a string of ’a’ characters, but he cannot determine the key used.

Problem 6

(a) helpmeimtrappedinsideacaesarcipherandcantgetout

(b) youmustbespeedoflightbecausetimestopswhenilookatyouhappyvalentinesday

(c) ihopeyouinterceptthissecrettransmissionwithoutanyerrorthistraonsmissionhastravelledamillionlightyears

(d) olddebayanisagoodoneexceptformelonmelonmelonmelonmelon

Problem 7

See README file for problem 7.