MATH 314 Spring 2018 - Class Notes

2/5/2018

Scribe: Bart Allen

Summary: In today's lecture we were introduced to the Substitution Cipher and the Vigenre cipher. Learning how each one works and how each one has its own method for encrypting and decrypting. Also, we see the faults with monoalphabetic ciphers.

Both the Affine and Caesar Cipher are both examples of an**Monoalphabetic Cipher**
Affine Cipher = 312 possible keys
Caesar Cipher = 26 possible keys

Monoalphabetic Cipher: One letter of plaintext always encrypts to the same letter of ciphertext.

How can we come up with other ciphers by mixing up the letters in other ways?
We can use a **Substitution Cipher** for that!

Substitution Cipher: this cipher is most commonly used in the puzzle section of the newspaper. Its key: is a Permutation of letters a - z.

Example:

Plaintext:	a	b	c	d	.....	z
Ciphertext:	K	Q	Z	A	.....	L

The number of possible keys 26 * 25 . . . . . 2 * 1 = 26!
$26! \approx 4.03 * 10^{26}$

* Unlike Caesar and Affine Ciphers, a brute force attack against a general substitution cipher isn't feasible 26! is way too big.

Attack Substitution Cipher:

- Cipher only attack
- Frequency attack
- Just because we can't use brute force doesn't mean that cipher is secure.

Known plain text: Just start reading off the key
Chosen plain text: Pick a sentence with all letters for example, "The quick brown fox jumps over the lazy dog" or simply list out "a,b.....z"

Now we need a new cipher that isn't monoalphabetic. **The Vigenre cipher!**

Vigenre cipher: this cipher uses a vector for both encryption and decryption.

Example: We first set our key to any word, lets make our word `"vector"` and then we convert the letters in the words into numbers $< 21, 4, 2, 19, 14, 17 >$.

To encrypt a message: We write the message without spaces or punctuation.
- convert letters into numbers and below that we write the word `"vector"` over and over again
- Add these two lines together (mod 26) convert numbers to letters to get ciphertext

Example: Let's encrypt `"here is how it works"`

h	e	r	e	i	s	h	o	w	i	t	w	o	r	k	s	
7	4	17	4	8	18	7	14	22	8	19	22	14	17	10	18	`Plaintext`
+ 21	4	2	19	14	17	21	4	2	19	14	17	21	4	2	19	`Ciphertext`
2	8	19	23	22	9	2	18	24	1	7	13	9	21	12	11	(mod 26)

Here is the encrypted message: `CITXWJCSYBHNJVML`

To Decrypt: we do the same process as we did to encrypt except we use subtraction instead

Example: Let's take `CITXWJ`

C	I	T	X	W	J	
2	8	19	23	22	9	
− 21	4	2	19	14	17	
7	4	17	4	8	18	(mod 26)
h	e	r	e	i	s	

Since the same letter of the plaintext encrypts to different ciphertexts frequency analysis no longer works when trying to attack or decrypt the Vigenere Cipher.

So how do we attack the Vigenere Cipher?

Known plaintext attack: - We can just subtract the plaintext from the ciphertext which gives us the key repeated over and over again.

Choosen Plaintext: -Pick `"aaaa.....a"` *plaintext* which we know "a" to be zero and therefore the encrypted ciphertext is just the key.

Ciphertext only: This method of attack is harder to implement because it requires the

length of the key to be calculated or found. This then led the Vigenere Cipher to be unbreakable for over 200 years! That is until Charles Babbage came along and cracked the code on how to break it with these various techqiues of finding the key length.

Finding the key length:

1. Write out the ciphertext message, excluding punctuation and spaces.
2. Below that write the message again except, shift all the ciphertext once to the right.
3. Then write the same ciphertext message again shifted two times, your repeating this same process by shifting by one from the previous line.

   Example:

```
Ciphertext:  VVHQWVVRHMUSG.....
              VVHQWVVRHMUS......
               VVHQWVVRHMU......
```

4. Count how often in each line the same letter occurs in the shifted ciphertext in correspondance to the unshifted ciphertext in the first line. These are known as *coincidences*
5. Count the number of coincidences for each shift length.
   Note: More coincidences occur when the shift is a multiple of the key length because the shifts cause letters of the english text to spike up as well as match up with other leters of the same freuquency.

Once we figure out the key length from the method above, we can use frequency analysis in each position to determine each letter of the key.