MATH 314 Spring 2018 - Class Notes

9/5/2018

Scribe: Evana Karim

Summary: Substiution ciphers such as Caesar ciphers and Affine ciphers are simple and easy to break. Today we talked about Vigenere ciphers, a more complex cipher.

Notes:

Monoalphabetic Ciphers: Any cipher that encrypts one letter at a time where one letter of plaintext always corresponds to the same letter of ciphertext, like caesar and affine ciphers.

- Caesar cipher: One of the simplest ciphers, only 26 keys; easy to brute force
- Affine cipher: Like a Caesar cipher but with multiplication, 312 keys; easy for a computer to brute force

Substitution Cipher: General monoalphabetic cipher with any possible permutation of letters.

So in this case, the key for a substitution cipher is a table listing `plaintext` letters and their corresponding `ciphertext`.

Ex:

a	b	c	d	...	z
V	Q	E	R	...	L

*each letter shows up only once in this table

How many possible keys are there for a substitution cipher?

$26! \approx 4.03 * 10^{26}$ because if you start with 26 letters, for each letter as you go on, you lose a code letter since no letters repeat.

This is too big for even a computer to brute force.
But just having too many keys isn't enough.

How would we attack a substitution cipher if brute force doesn't work?

- `Ciphertext` only attack: Frequency analysis can still break messages easily
- Known `plaintext` attack: Start reading off elements of the table any holes can be guessed later

- Chosen `plaintext` attack: Encrypt the sentence "the quick brown fox jumps over the lazy dog" and then you can read off the entire table

Any monoalphabetic cipher is a substitution cipher so we need to move on to new types of ciphers

Vigenere Cipher

- different letters get encrypted in different ways for the first time
- key is any word; take the key and write it as a vector of numbers
- to encrypt a `plaintext`, we convert a `plaintext` to numbers as well; write without spaces
- below that we write our key vector over and over again until we have one number beneath each letter of `plaintext`

Ex:
Encrypt "this is how it works", key = "vector"

	t	h	i	s	i	s	h	o	w	i	t	w	o	r	k	s
	19	7	8	18	8	18	7	14	22	8	19	22	14	17	10	18
	v	e	c	t	o	r	v	e	c	t	o	r	v	e	c	t
+	21	4	2	19	14	17	21	4	2	19	14	17	21	4	2	19
	14	11	10	11	22	9	2	18	24	1	7	13	9	21	12	11
	O	L	K	L	W	J	C	S	Y	B	H	N	J	V	M	L

*note how the first "i" became a "k" while the second became a "w" For this reason, the Vigenere cipher was considered unbreakable for about 200 years

What does this table mean:
The first row is the `plaintext` with no spaces
The second row is the `plaintext` converted into numbers
The third row is the key (vector) repeated over and over again until the end of the `plaintext`
The fourth row is the key converted into numbers
Add up the numbers and you get the `ciphertext` in numbers
Convert the numbers and you get the full encrypted ciphertext

How can we break a Vigenere cipher?

1. figure out length of the key (the hardest part)
2. use frequency analysis on letters in each position

How can we figure out the length of the key?

1. Write out `ciphertext` in one big long line.

2. Below it write the `ciphertext` shifted to the left one letter.

3. Below that write it again, shift it twice to the left and so on

4. Count the number of coincidences between the shifted `plaintext` and original `plaintext` (By coincidence, we mean the same letter in the same position)

Ex:

		V	V	H	Q	W	**V**	V	R	H	...
	V	**V**	H	Q	W	V	**V**	R	H	M	...
V	V	H	Q	W	V	V	R	H	M	U	...

The bolded "V"'s are coincidences. There are a total of 2 coincidences in these shifts. We find more coincidences when the shift is the length or a multiple of the key length. This happens because letters occur in different frequencies in English.
Probability says that we should get more coincidences when letters are shifts the same amount.