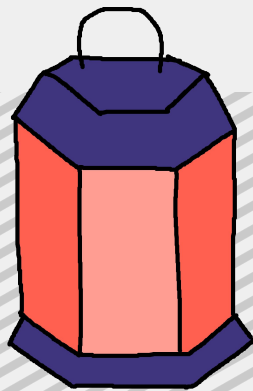




# A Practical Introduction to Cryptanalysis



Cameron Lonsdale  clonsdale  @myoutpost

# \$ whoami

- UNSW Computer Science Undergrad
- Security Society && SECedu events
- (Future) Crypto Nerd

```

      \x90 \x90 \x90
      \x90 \x90 \x90
      \x90 \x90 \x90
      \x90 \x90 \x90
      \x90 \x90 \x90
      \x31 \xc0 \x50 \x5b \x68
      \x20 \x3e \x0a \x68 \x65
      \x73 \x6f \x68 \x73 \x77
      \x73 \x68 \x3c
      \x20 \x75 \x63
      \xb2 \x10 \x54
      \x59 \x43 \xb6
      \x04 \xcd \x80 \x90 \x90
      \x90 \x90 \x90 \x90 \x90
      \x90 \x90 \x90 \x90
      \x90 \x90 \x90
      \x90 \x90 \x90
      \x90 \x90 \x90

```

university of new south wales

/sec/soc



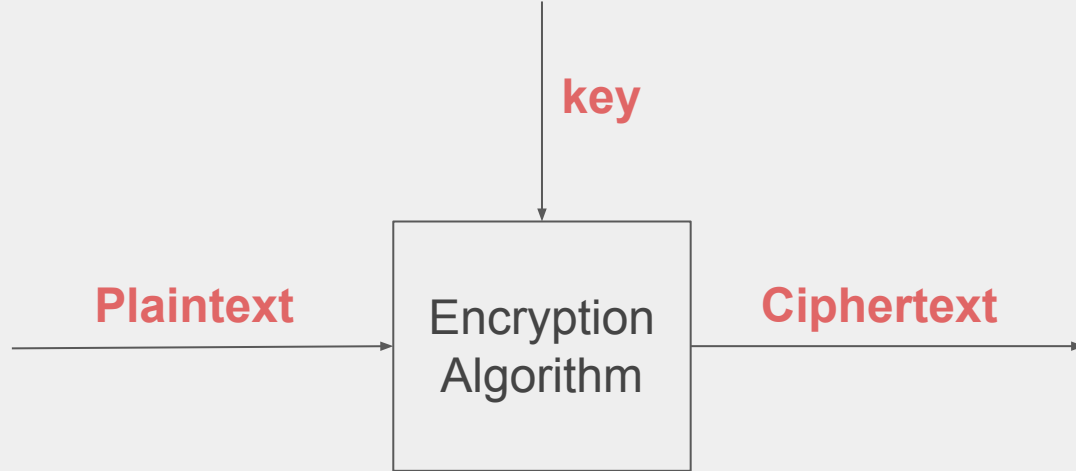
I sometimes blog things:  
[cameronlonsdale.wordpress.com](http://cameronlonsdale.wordpress.com)

And am on the tweeters:  
[@myoutpost](https://twitter.com/myoutpost)

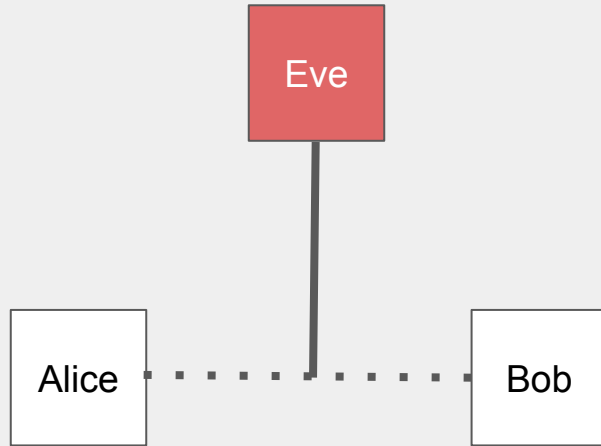
# What is Cryptography?

Design of systems for securing information.

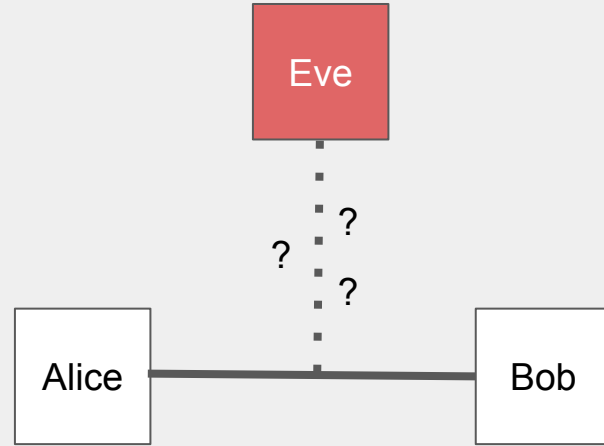
Different building blocks can give you different properties.



# Why do we care? - Communication



Without Cryptography

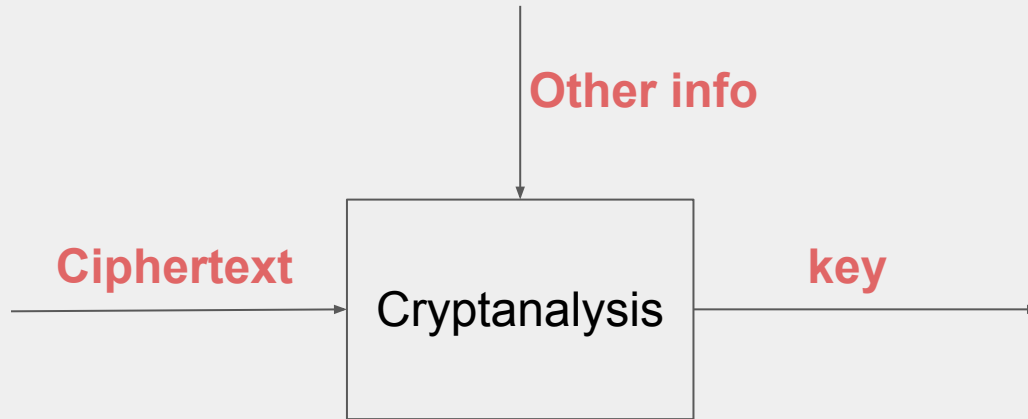


With Cryptography

# What is Cryptanalysis?

Recovering plaintext without knowing the key.

It's gotten a lot more difficult, NP difficult. Today we need to get creative and consider implementation bugs and crazy side channel attacks.



# Why do we care?



But also learning how things are weak help us build more secure systems.

# Different kinds of Cryptanalysis

We can do different attacks depending on how much information is available.

- **Ciphertext-only**
- **Known-plaintext**
- **Chosen-plaintext**
- Side Channel Attacks
- Brute Force
- Birthday attacks

Several general approaches

- **Frequency**
- **Differential**
- Integral
- Linear

Cryptanalysis can also be of varying levels of usefulness.

**Total break** vs Partial break.

# Ciphertext-only attacks

We only have access to a set of ciphertexts. Relies on the ability to be able to discover something about the key from the ciphertext alone.

- Having an appropriate amount of ciphertext is a must.
- Modern day ciphers have made this attack much harder.
- But Classical Ciphers are the best examples here.



# Caesar Cipher

Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Cipher: YZABCDEFGHIJKLMNOPQRSTUVWXYZ

# Encryption

**Cyclically Shift** each letter  $k$  places forward

$k = 3$

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

For  $k = 3$ , the plaintext **HELLO** is encrypted as **KHOOR**

Number of keys

$$|K| = 26$$

Or 25 if you don't count  $k = 0$

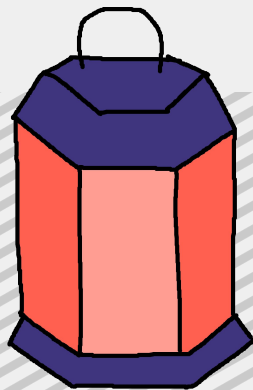
# Decryption

Brute force is best force

Only 25 possible keys to check, let's just check them all!

JGNNQ	ZWDDG	PMTTW
IFMMP	YVCCF	OLSSV
HELLO	XUBBE	NKRRU
GDKKN	WTAAD	MJQQT
FCJJM	VSZZC	LIPPS
EBIIL	URYYB	
DAHHK	TQXXA	
CZGGJ	SPWWZ	
BYFFI	ROVVY	
AXEEH	QNUUX	

# Demo



# Simple Substitution Cipher

THIS EXAMPLE IS TO SHOW YOU THE POWER

OF FREQUENCY ANALYSIS THE ENGLISH

LANGUAGE MAKES THIS POSSIBLE DUE TO THE

FREQUENCY OF THE LETTER E AND T.

# Encryption

**Permute** the alphabet for a key, then map letters to encrypt.

Mapped alphabet to a scrambled version

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

The plaintext **HELLO** is encrypted as **XUOOB**

## Number of Keys

$$|K| = 26! \approx 4 \times 10^{26}$$

About the number of atoms in your head!

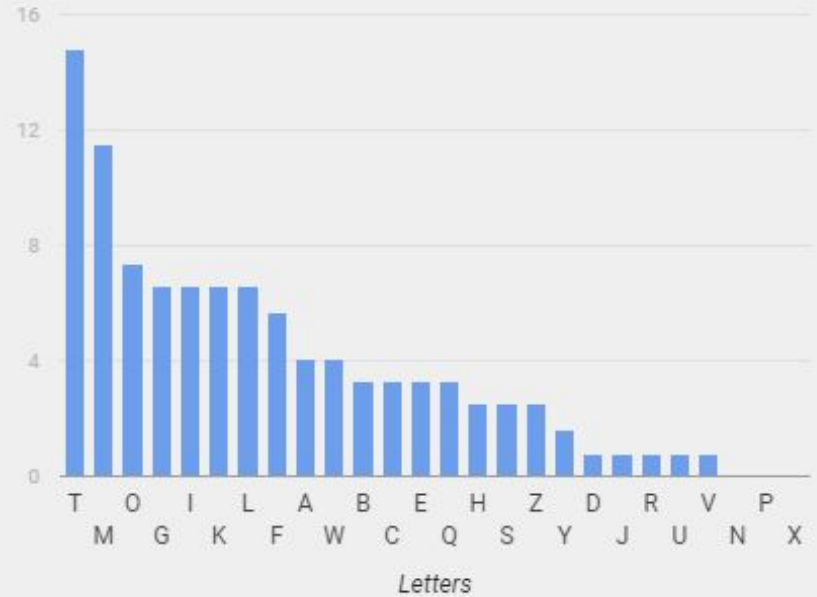


# Decryption - The magic of frequency

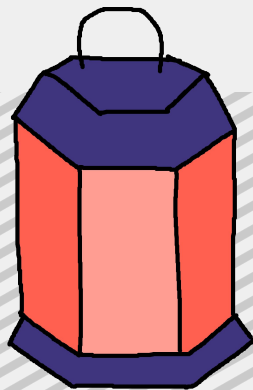
English Frequency



Ciphertext Frequency



# Demo



# Decryption - More letters the better

N-grams, like letters but more of them!

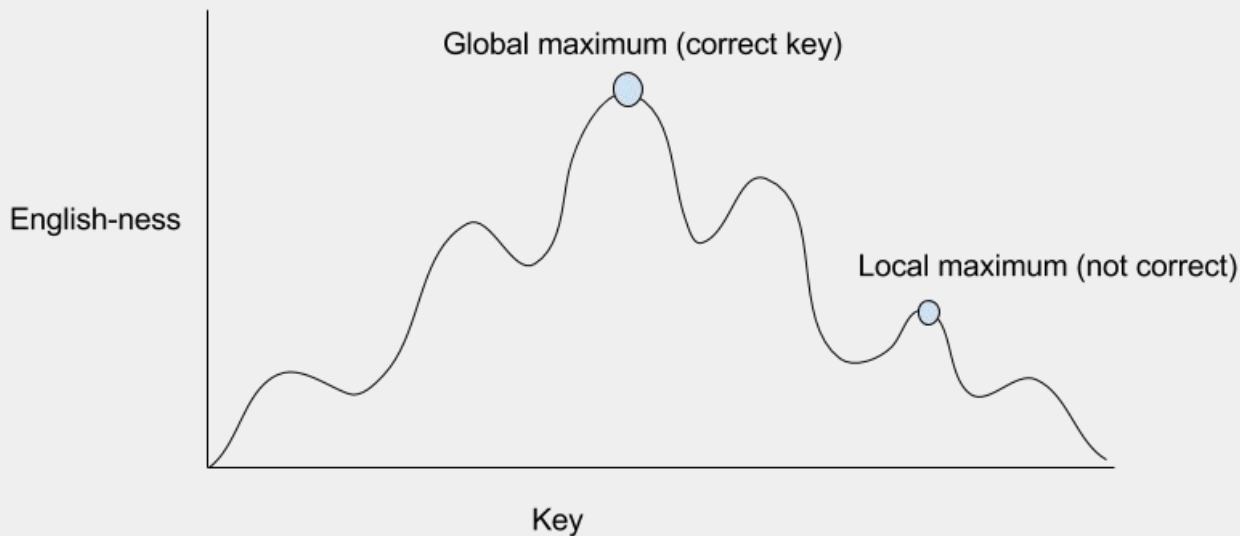
Bigrams - TH is common, QU normally appear together...

Trigrams - THE, AND, ING are common

Example time:

Ei fyprx rqkx qb djpus fqtqtql. Q mpt's rpbx pksxt. Q drepbs txgxz rpbx. Q ydgx d lzxds  
zxrdsqptbyqa fqsy syx Exhqndt axparx. Q fdb d lzxds bsumxts. Q fdb lppm ds xgxzisyqtl.  
Q'rr mzqtc fdsxz. Bpexsqexb spedsp ouqnx, fyqny Q rqcx. Bpexsqexb pzdtlx ouqnx, fyqny Q  
rqcx. Q'rr mzqtc mqkkxxts syqtlb. Jus syx Npcx pz Axabq jppbsb ipu ua d rqssrx. Q  
fpurmt's eqtm d rqssrx jpf. Qt Odadt, syxi jpf. Q rpgx qs. Ptri syqtl Q rpgx djpus Odadt.  
Fx fqrr edcx Dexzqnd bszptl dldqt. Fx fqrr edcx Dexzqnd azpum dldqt. Fx fqrr edcx Dexzqnd  
bdkx dldqt. Dtm fx fqrr edcx Dexzqnd lzxds dldqt.

# Automating the Substitution Cipher



Fiddling around with the key moves us around on the graph.

We can determine englishness by using frequency AND letter dependencies for more accuracy.

# Determining English-ness

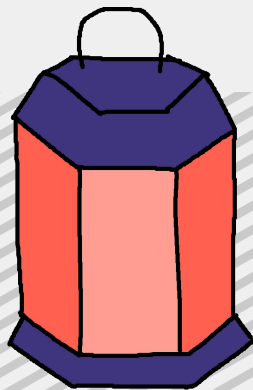
No, we don't need machine learning.

Lets just use frequency of n-grams!

Text that has more THE, AND, ING, will have a higher score than text that mostly has JZX, QPJ, ZMQ.

```
>>> fitness = NgramScore(english.trigrams)
>>> fitness("THE")
-1.7413540758516952
>>> fitness("JZX")
-9.630847514866344
```

# Demo



# Vigenère Cipher



# Encryption

$r$  different Caesar Ciphers applied periodically

Key	C	O	D	E	C	O	D	E	C	O	D	E	C	O	D
Plaintext	T	H	I	S	I	S	A	N	E	X	A	M	P	L	E
Ciphertext	V	V	L	W	K	G	D	R	G	L	D	Q	R	Z	H

$$A = 0, B = 1, C = 2$$

$$T + 2 = V$$



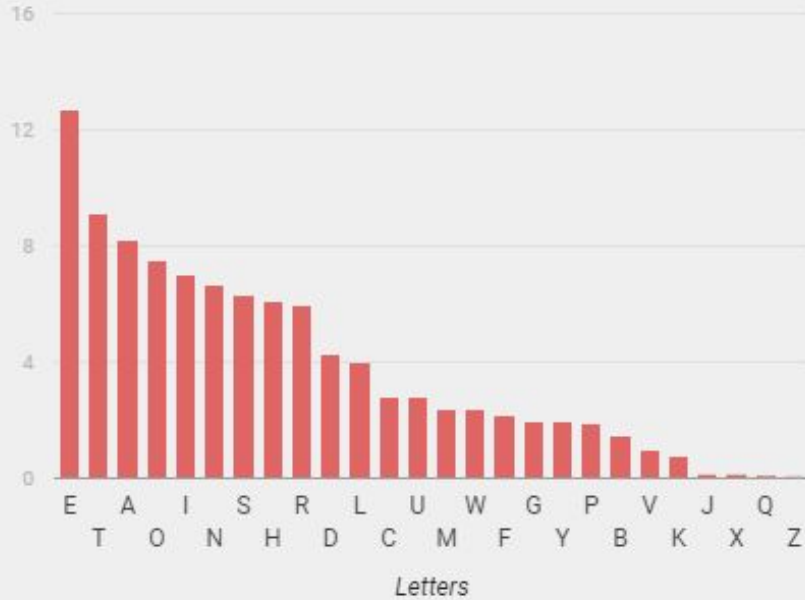
## Some Extra information

$$|K| = 26^r$$

Will get too big to brute force

# Decryption - That don't look right

English Frequency



Ciphertext Frequency



# Decryption - Frequency can still save us

Remember back to the frequency when encrypting with Substitution / Caesar, it did not change!

If the key length was 4..

HNQD

LVYO

POKF

ACCE

KYAT

....

Frequency of each column should look like the frequency of english.

# Decryption - What a coincidence!

So I'm meant to ~feel~ whether or not the frequency is similar to English?

I didn't come here to feel.

**Index of Coincidence** - A summary of frequency

The probability of two letters randomly selected being the same.

$$I.C. = \frac{\sum_{i=A}^{i=Z} f_i(f_i - 1)}{N(N - 1)}$$

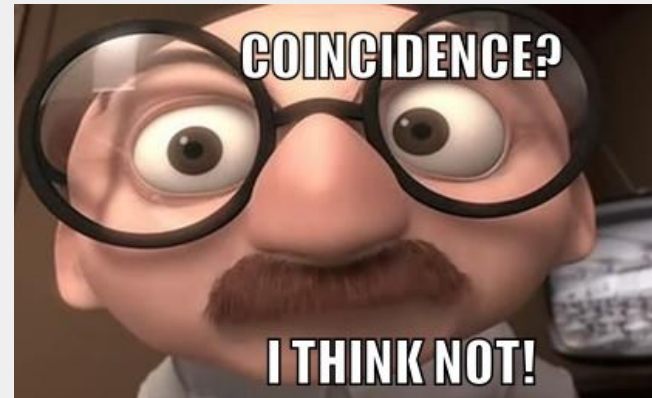
$f_i$  is the count of the letter  $i$ .

$N$  is total number of letters in the ciphertext

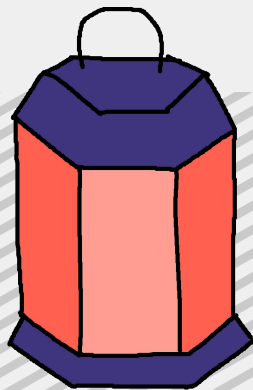


# Decryption - I.C of English

Text	I.C
English	0.066
Substitution Cipher	0.066
Vigenère Cipher	0.042



# Demo



# Custom Ciphers & Extra



# Try to get as much information as you can

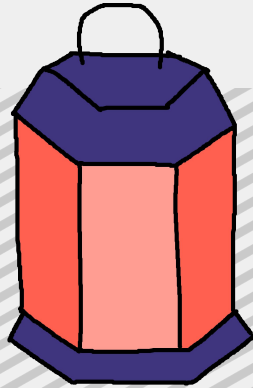
Try to identify the cipher and the source language from the ciphertext.

If you can know plaintext or chosen plaintext attack, look for **patterns** and **differences**.

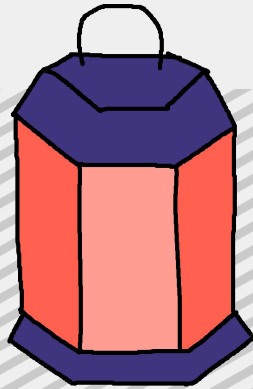
**Differential Cryptanalysis** - Studying the differences in input can affect output. Help to discover non-random behaviour.



# Chosen-plaintext Demo

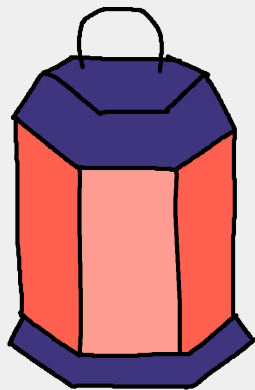


# Known-plaintext Demo



# Lantern - It's only basic - for now

Lantern is pretty basic, and a little buggy, but does provide some useful features.



Issues & PR's  
Welcome!

[github.com/CameronLonsdale/lantern](https://github.com/CameronLonsdale/lantern)

# Torch - for when you want to burn things

Command-line Cryptanalysis

Issues & PR's

Welcome!



[github.com/CameronLonsdale/torch](https://github.com/CameronLonsdale/torch)

## Where can I learn more?

- <http://practicalcryptography.com/>
- <http://overthewire.org/wargames/krypton/>
- <https://www.crypto101.io/>
- <https://cryptopals.com/>
- Serious Cryptography - JP Aumasson (Early Access)
- Handbook of Applied Cryptography and similar

# Thank You!

