Introducing Cryptanalysis

Dr Richard Clayton

richard.clayton@cl.cam.ac.uk

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Hills Road Sixth Form College
8 October 2009

It’s all Greek to me!

ΡΙΧΑΡΔ

ΡΙΧΗΑΡΔ

Secret communication

- Steganography: hiding the message
  - Under the hair (Histaiaeus c480BC)
  - Invisible ink
  - Pin pricks (Victorians)
  - Microdots (WWII spies)
  - LSB of colour data of digital photographs
- Cryptography : making the message unintelligible
  - Codes one symbol per idea
    - 101 = attack at dawn
    - 102 = enemy ships sighted
  - Transpositions
    - T I M S A E S L G T Y C A B E H S E S G I S I H L S R M L D
  - Ciphers transliterating letters (or sometimes syllables)
    - YHQL YHGL YLFL

Mono-alphabetic ciphers

- Caesar cipher
  - YHQL YHGL YLFL use third letter along
  - “ROT13” widely used in Usenet, email etc
- Generalise with a “key”

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

- Now only need to transmit the key, and we apparently have an unbreakable cipher (given some obvious improvements such as scrambling the right hand end of the alphabet…)
- Unfortunately not…

Breaking mono-alphabetic ciphers

- This advance was made in the Arab world, c 800 AD
- Letters have a constant mapping
- Letters are not equally likely to appear in real text
- Hence by counting letter frequencies you can pick out some of the letters, and then mop up the rest with inspection…

a	b	c	d	e	f	g	h	i	j	k	l	m
8.2	1.5	2.8	4.3	12.7	2.2	2.0	6.1	7.0	0.2	0.8	4.0	2.4
n	o	p	q	r	s	t	u	v	w	x	y	z
6.7	7.5	1.9	0.1	6.0	6.3	9.0	2.8	1.0	2.4	0.2	2.0	0.1

(hence ETAOIN SHRDLU)

- Note that the letter frequencies differ in other languages (worth bearing in mind if you need to decode a French missive)

Modern attacks on mono-alphabetics

- Digram (letter pair) frequency also highly variable hence can use this as an alternative or additional approach
- One way of programming the attack is using a “hill climbing” algorithm:
  1. Create a scoring function (e.g. sum the likelihood of digrams)
  2. Randomly select a starting arrangement and calculate score
  3. Swap two letters at random, if this improves the score keep it
  4. Continue swapping for a while until score does not increase
  5. Return to step 2 until bored
  6. Use the best score
- Step 3 finds local maxima, step 5 avoids sub-goals
- Method is entirely general and can be used for cryptanalysis of many different ciphers

Other information may help

- GUR YVBA VF BSGRA PNYYRQ GUR XVAT BS GUR WHATYR, OHG GUR ORFG CYNPR GB SVAQ BAR VF BA GUR CYNVAF
- QBA'G GUVAX GJVPR VG'F NYEVTUG
- "V NZ N TRAVHF", UR FNVQ
- NNEQINEX VF BAR BS GUR SRJ RATYVFU JBEQF FGNEGVAT JVGU GJB YRGGREF GUR FNZR!

Defences against cryptanalysis

- Avoid giving away structure of the message!

  GURYV BAVFB SGRAP NYYRQ GURXV ATBSG URWHA

  TYROH GGURO RFGCY NPRGB SVAQB ARVFB AGURC

- Misspell ze wordz to distort ze frekwensie
- Ad-d nu-l-l-s t-o th-e me-ss-age
  - If coding into two digits then get 74 nulls “for free”
- Use of codes for standard words (king, advance, tonight…)
- Or even encode syllables, add “delete” codes etc
  - c.f. “Great Cipher” of Louis XIV
- Homophonic ciphers
  - Use more symbols for more common letters
- Nevertheless… mono-alphabetics are not very strong

Vigenère cipher

- Multiple Caesar ciphers

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
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	a
c	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
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
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	d
f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	I	j	k	l

…etc

Encoding with the Vigenère cipher

- Keyword used to select rows to use

  B L A C K B L A C K B L A C K B L A C K B L A C K

  f l e e a t o n c e a l l i s d i s c o v e r e d

- Now look up column “f” in the row starting “B”
- Then look up column “l” in the row starting “L”
- Then look up column “e” in the row starting “A”

  g w e g k u z n e o b w l k c e t s e y w p r g n

- Viz:
  - apply Caesar cipher + 1 to every 6th letter starting at 1
  - Apply Caesar cipher +12 to every 6th letter starting at 2
  - apply Caesar cipher + 0 to every 6th letter starting at 3 &c
- This is a “poly-alphabetic” cipher and is much stronger

Babbage attack on Vigenère

- Observe that only 6 ways of encoding “the”
  - A common sequence, but there will be many others such as “ing”
  - Encoding depends where in the keyword it starts
- So look for repeated sequences, likely (no guarantee!) to be same sequence of letters encoded against the same key
- Then assess how far apart these sequences occur within the ciphertext: there will be a common factor to these distances which must be the keyword length
- Now we have <n> mono-alphabetic ciphers (possibly even just Caesar ciphers which will be especially easy) to analyse… and that’s a solved problem
  - only practical issue is that we have just one sixth the text
- Usually known as the Kasiski Test (!)

Index of coincidence

- William F. Friedman (1891-1969)
  - Worked for US Army in WWI, then the “American Black Chamber” and thereafter the Army’s Signals Intelligence Service and then in the 1950s became chief cryptologist for the NSA
- Place two texts side by side and count when letters coincide
- Essentially this is what we did with Vigenère, using different parts of the same ciphertext
- Can be used to guess what language a text is in… (self-similarity is less in English than in French or German)

Playfair (Wheatstone 1854)

- Uses keyword (CWHEATSTONE) followed by rest of alphabet

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

- Split message into pairs of letters (with X if double), encode into the opposite pair from the table, going across first (or if same row/column then use letters from right/below)…

FL	**EX**	**EA**	**TO**	**NC**	**EA**	**LX**	**LI**	**SD**	**IS**	**CO**	**VE**	**RE**	**DX**
DM	**HY**	**AC**	**SN**	**TE**	**AC**	**PU**	**QD**	**TF**	**FN**	**HT**	**YW**	**QA**	**GU**

Cryptanalysis of Playfair

- Note that output will never contain doubled letters (so if there are some, it's not Playfair!)
- Digrams can only be encoded two ways, so do a frequency analysis of cipher text and thereby identify common pairs
- Reversed digrams in same word are common (REceivER, DEpartED) so try looking for these
- Modern approach is to use hill-climbing algorithm (as earlier) and permute the square each time looking for improvements in score as to whether decoding looks "more like English"
- In WWII Germans used a "Double Playfair" system (random squares and one letter from each of two squares). Broken at Bletchley Park because messages started with spelled out numbers (which gave away much of the squares)

One time pads (provably unbreakable)

- If key is same length as the message (and has no structure), then impossible to break without knowing one or the other
  - Consider how broke poly-alphabetic by using frequency of $1/n^{th}$ of the message – in the limit there's no frequency
- Note that modern ciphering systems tend to use numeric (binary representations of the text):

```
Text:    4D 6F 64 65 72 6E 20 6D 65 73 73 61 67 65
Key:     D0 11 18 B3 F6 9A 36 77 49 D2 78 0A 9F 0E
Total:   1D 80 7C 18 68 08 56 E4 AE 45 EB 6B 06 73
```

- Combination usually done with PLUS (mod 256) or XOR

Two-time pads are very breakable

- Plaintext 1 + OnetimePad = Ciphertext 1
- Plaintext 2 + OnetimePad = Ciphertext 2
- Hence Ciphertext1 – Ciphertext2 = Plaintext1 – Plaintext2
- So get zeros where texts match up
- And very usefully, if any text is expected to be present then can assume it is present at particular positions, and then see what the other plaintext must then be
  - Bletchley called overlaying “depth” and would “drag” words such as “eins” through combined ciphertext
  - They were much helped by text such as “oberleutnantfuehrer”

Mechanical encoding methods

- Distributing one-time pads difficult and expensive
  - Nevertheless, widely used by diplomats and spies
  - Venona project dealt with re-use of Soviet key material!
- Hoped to approximate one-time pad with mechanical schemes
  - Sekret Decoder Rings (effectively Vigenère)
  - Enigma (multiple wheels being stepped along)
  - Rotor machines such as the US SIGABA
- Enigma cryptanalysis relies on many weaknesses
  - Enigma operators encoded a message setting twice at message start and this gave patterns that the Poles exploited
  - Message settings often non-random (“cillies”)
  - Repeated plaintext (weather reports) gave “cribs”
  - Bletchley sometimes just used captured codebooks

Modern cryptography (e.g. RC4)

```
for i from 0 to 255
        S[i] := I
endfor
j := 0
for i from 0 to 255
        j := (j + S[i] + key[i mod keylength]) mod 256
        swap(S[i],S[j])
endfor
i := 0
j := 0
while GeneratingOutput:
        i := (i + 1) mod 256
        j := (j + S[i]) mod 256
        swap(S[i],S[j])
        output S[(S[i] + S[j]) mod 256] ^ input[i]
endwhile
```

Gratuitous advice!

- Expect early rounds to be simple classical ciphers
  - Don’t forget simple stego!
- Expect later rounds to be variants of more complex schemes
- Check what the letter frequencies are
- Bletchley recruited people good at crosswords and anagrams, look at the shape of the words you’re discovering
- Pay attention to what the decoded text says, it may be useful in a later round (viz: it may be providing cribs or keys)
- Don’t assume everything is in English!
- Lots of resources (and a few programs) on the Internet (but the puzzle setters know this as well!)

Introducing Cryptanalysis

http://www.lightbluetouchpaper.org

UNIVERSITY OF CAMBRIDGE
Computer Laboratory