Cracking the Vigenère Cipher in Haskell

A *Lightning Talk* for **BayHac 2015**

David Banas capn.freako@gmail.com

June 14, 2015

The Caesar Cipher

- *Mono*-alphabetic rotation cipher
  (Shift of 3 shown, but can be anything in [1, 25]:

a	b	c	d	e	f	g	h	i	j…
d	e	f	g	h	i	j	k	l	m…

- Easily cracked, using *frequency analysis*.
  - Find the most frequently occurring letter in the ciphertext.
  - Assume that letter represents ‘e’; now, you know the shift.

The Vigenère Cipher

- *Poly*-alphabetic rotation cipher:

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

- Foils frequency based cryptanalysis.

Vigenère Cipher Example

- The ciphertext:

j yt up esaf tm bg byjk bt iaaiaa! k rchlnz jvvg scliph wvu blj aibil ape g wkth ahcu ul fjd ahkt kvrg odaep tfhn pnal c ychr. g lqpk motxapk vp flatjne anm roe uajrs zos hcwe wrgqapld bnb dktcszskog ahgn uptj ymb cu jlniuh, upuij ma hchd ceepnu tm sopkc. uhgz kt y wqodcyfwm eyowq mm rfonse, ajpvg wgah drchtkwirf, bcalpvbnal, bnb cqnmsuivz qwitjt, ape g cqvnr matejm xfrw hprrbncue ao ce h rbrr oh ir. k'm h njtrse keysowt mm vie iaaqienigt klevvp, wjjcf iu ajsoyfd ao dolcepf gu c rchlnz lpcg lgukgeil epndlrgocc rqpm hnf iq sgsvck fjnllr. hmd ebn de her owsscsvgt y cqspmyavf qwoptopzhkq jpkg tfht? i diui g cqvlb ahgopk vie juv il scmapf jt dowmd yesvipl, jn vrffr ao her sqnemue uo oitf kl vp bv jbsilln ppvfgtsgvncmlw. cmaq, oz ipdu apl vpo keobnbpni. khydf ymtgs pltkseklnv. p jppc yqv ysl iatl c wmudgsfss yfeilnf ar bczhyj!

Vigenère Cipher Example (cont'd.)

- Result of attempting frequency based cryptanalysis:

  j yt up esaf tm bg byjk bt iaaiaa! k rchlnz jvvg scliph wvu blj aibil ape g wkth ahcu ul fjd ahkt kvrg odaep tfhn pnal c ychr. g lqpk motxapk vp flatjne anm roe uajrs zos hcwe wrgqapld bnb dktcszskog ahgn uptj ymb cu jlniuh, upuij ma hchd ceepnu tm sopkc. uhgz kt y wqodcyfwm eyowq mm rfonse, ajpvg wgah drchtkwirf, bcalpvbnal, bnb cqnmsuivz qwitjt, ape g cqvnr matejm xfrw hprrbncue ao ce h rbrr oh ir. k'm h njtrse keysowt mm vie iaaqienigt klevvp, wjjcf iu ajsoyfd ao dolcepf gu c rchlnz lpcg lgukgeil epndlrgocc rqpm hnf iq sgsvck fjnllr. hmd ebn de her owsscsvgt y cqspmyavf qwoptopzhkq jpkg tfht? i diui g cqvlb ahgopk vie juv il scmapf jt dowmd yesvipl, jn vrffr ao her sqnemue uo oitf kl vp bv jbsilln ppvfgtsgvncmlw. cmaq, oz ipdu apl vpo keobnbpni. khydf ymtgs pltkseklnv. p jppc yqv ysl iatl c wmudgsfss yfeilnf ar bczhyj!

- Held for 3 centuries!

Charles Babbage's Approach

- Find the spacing between several pairs of identical "words" in the ciphertext:

j yt up esaf tm bg byjk bt iaaiaa! k **rchlnz** jvvg scliph wvu blj aibil ape g wkth ahcu ul fjd ahkt kvrg odaep tfhn pnal c ychr. g lqpk motxapk vp flatjne anm roe uajrs zos hcwe wrgqapld bnb dktcszskog ahgn uptj ymb cu jlniuh, upuij ma hchd ceepnu tm sopkc. uhgz kt y wqodcyfwm eyowq mm rfonse, ajpvg wgah drchtkwirf, bcalpvbnal, bnb cqnmsuivz qwitjt, ape g cqvnr matejm xfrw hprrbncue ao ce h rbrr oh ir. k'm h njtrse keysowt mm vie iaaqienigt klevvp, wjjcf iu ajsoyfd ao dolcepf gu c **rchlnz** lpcg lgukgeil epndlrgocc rqpm hnf iq sgsvck fjnllr. hmd ebn de her owsscsvgt y cqspmyavf qwoptopzhkq jpkg tfht? i diui g cqvlb ahgopk vie juv il scmapf jt dowmd yesvipl, jn vrffr ao her sqnemue uo oitf kl vp bv jbsilln ppvfgtsgvncmlw. cmaq, oz ipdu apl vpo keobnbpni. khydf ymtgs pltkseklnv. p jppc yqv ysl iatl c wmudgsfss yfeilnf ar bczhyj!

Charles Babbage's Approach (cont'd.)

- Find several sets of these pairs, at different spacings, and factor the spacings:

	Factors									
Spacings	2	3	4	**5**	6	7	8	9	10	11
35				**x**		x				
95				**x**						
120	x	x	x	**x**	x		x		x	
130	x			**x**					x	

- Find the single common factor; that is the keyword length.

Charles Babbage's Approach (cont'd.)

- De-interleave the ciphertext, by taking every n^{th} letter, where *n* is the keyword length, and doing this n times, starting at positions 0 thru (n – 1).
- You now have n subsets of the original ciphertext, which have all been enciphered, using a simple Caesar cipher.
- Crack them, individually, using frequency analysis, and re-interleave the results to form the final answer.

Charles Babbage's Approach (cont'd.)

j yt up esaf tm bg byjk bt iaaiaa!

jemja! {0, 5, 10, …}

Ysbka {1, 6, 11, …}

Teagbi {2, 7, 12, …}

Ufbta {3, 8, 13, …}

Ptyia {4, 9, 14, …}

vigeneres_crack.hs

```
pairSpacings :: Int -> String -> [Int]
pairSpacings l xs =
  concat
    [[n | n <- [l..(length xs' - l)],
          take l xs' == take l (drop n xs')]
          | xs' <- [drop m xs |
                    m <- [0..(length xs - 2 * l)]]]
```

vigeneres_crack.hs (cont'd.)

```
commonFactors :: [Int] -> [Int]
commonFactors xs =
  foldl intersect (head xss) (tail xss)
    where xss = [factors x | x <- xs]
```

vigeneres_crack.hs (cont'd.)

```
vcrack :: String -> String
vcrack xs = interleave
  [crack xs' | xs' <- [takeEvery n (drop m xs)
                          | m <- [0..(n - 1)]]]
    where n = maximum com_facts
          com_facts = commonFactors pair_spacings
          pair_spacings = pairSpacings 5 xs
```

Result

i am so glad to be back at bayhac! i really love seeing you all again and i wish that we did this more often than once a year. i look forward to hearing all the talks you have prepared and discussing them with you at length, until my head begins to smoke. this is a wonderful group of people, alive with creativity, acceptance, and community spirit, and i count myself very fortunate to be a part of it. i'm a little jealous of the baypiggies meetup, which is allowed to convene in a really nice linkedin conference room and is served dinner. how can we get ourselves a corporate sponsorship like that? i wish i could afford the cut in salary it would require, in order to get someone to hire me to do haskell professionally. alas, my kids are too demanding. maybe after retirement. i hope you all have a wonderful weekend at bayhac!