Cryptology Through History and Inquiry

Cryptology Through History and Inquiry

Charles F. Rocca Jr.
Western Connecticut State University

Introduction

Goals and Aims of the text:

- Present a *brief* overview of the history of cryptology
- Present the cryptology as it was not with significantly newer or older concepts
- Exception to the previous statement will be the use of some modern terminology where appropriate
- Present a variety of mathematical topics: Statistical analysis, Linear Algebra, Number Theory, Group Theory
- Present the material from an inquiry based perspective when possible
- Present the text as a battle back and forth between making and breaking codes

What should be covered in terms of content can be summarized by the list:

- 500 B.C.E. (and before): Atabash, Scytale, Demaratus, Histiaeus
- 50 B.C.E.: Caesar
- 700-1000: Arab Cryptology*
- 140”0-1850: Foiling Frequencies*: Alberti’s, Vigenére’s , Jefferson’s, and Playfair’s Ciphers
- 1846: Babbage / Kasiski Test*
- 1914-1918: World War I: Zimmermann’s Telegram, ADFGVX Cipher, etc.
- 1919: One Time Pad*
- 1929: Hill’s Cipher*
- 1934: Enigma* in the service of the German military
- 1978:” RSA*
- 1991: Zimmermann and PGP
- 2000: Quantum Cryptology

Contents

Chapter 1

Caesar’s Shifty Idea

In this chapter discuss ancient encryption systems:

- Caesar’s Shift
- Skytle
- Others?
- Slight improvements: Nomenclatures, Nulls, etc.

1.1 Simple Early Ciphers (and a little math)

Definition 1.1.1 Cipher. A **cipher** is a system by which a letter or block of letters in a message is replaced by another letter or block of letters in a systematic way. In this chapter we will focus on **monoalphabetic substitution ciphers** in which the substitution is always carried out in the same way with one-to-one correspondence between letters.

One of the earliest types of cipher (sometimes spelled *cypher* in older documents) that we have a record of was the Caesar shift cipher used by Julius Caesar around 100 BCE. Here it is as described by Suetonius who lived about 150 to 200 years after Julius Caesar.

> “There are extant likewise some letters from him to Cicero and others to his friends concerning his domestic affairs in which if there was occasion for secrecy he wrote in cyphers, that is he used the alphabet in such a manner that not a single ward could be made out. The way to decipher those epistles was to substitute the fourth for the first letter as d for a and so for the other letters respectively.” [12, p. 37]

Augustus Caesar a short time later also employed a shift cipher, though arguably a much less effective one:

> “When he had occasion to write in cypher he put b for a, c for b [and] so forth and instead of z, aa.” [12, p. 134]

Comprehension Check:

- In Julius Caesar's cipher what letter was substituted for a? What about in Augustus' cipher, what letter was substituted for a?
- In Julius Caesar's cipher what letters would replace b, c, and d? What letter should replace n?
- In Julius Caesar's cipher how would you need to handle the letters x, y, and z?
- For each **plaintext** letter write in the corresponding **CIPHERTEXT** letter using Julius Caesar's cipher system. (We will generally follow the convention that plaintext is lowercase and ciphertext will be uppercase.)

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

Table 1.1.2: Monoalphabetic Substitution Table

Now use the table you filled in to check you understanding.

Checkpoint 1.1.3. Use your table to convert this to CIPHERTEXT: "hello world"

Checkpoint 1.1.4. Use your table to convert this to plaintext: `WKLV LV D WHVW`

Checkpoint 1.1.5. Use your table to convert this to plaintext: `CRPELHV OLNH CHEUDV`

Both of the ciphers described above are called **shift ciphers** because of how they move letters along in the alphabet, but do not change their order.

Try to figure out the message in the passage below which has been **enciphered** using a different shift.

Checkpoint 1.1.6. Cipher Text:

```
OLYL PZ H SVUNLY TLZZHNL MVY FVB AV WYHJAPJL VU.
AOL ZOPMA MVY AOPZ TLZZHNL DHZ IF H MHJAVY VM
ZLCLU, ZV AOL H NVLZ AV AOL O.  AOPZ KVLZU'A
YLHSSF VMMLY TBJO TVYL ZLJBYPAF, LZWLJPHSSF ZPUJL
AOLYL HYL VUSF ADLUAF ZPE WVZZPISL ZOPMA JPWOLYZ
HUK VUL VM AOVZL KVLZU'A JOHUNL HUFAOPUN.  DOPSL
FVB DLYL KLJPWOLYPUN AOPZ DOHA WHAALYUZ KPK FVB
UVAPJL? DOHA DVYKZ ZAVVK VBA?  OVD JVBSK DL THRL
AOPZ H TVYL ZLJBYL (OHYKLY AV JYHJR) JPWOLY?
AOLZL HYL HSS XBLZAPVUZ AOHA DL DPSS PUCLZAPNHAL
PU TVYL KLAHPS AOYVBNOVBA AOL ALEA.
```

Hint.

- Which letters stand alone, i.e. are there one letter words? Can this tell you something about which letters they represent?

- What about two or three letter words? What might they be?
- Do some of the letters appear more often than others?
- Can you think of other properties of words and letters that could help us understand what a message says?

Were you able to **decrypt** the message? Throughout this text we will present material through increasingly more difficult ciphers and codes. Now, take some time to try and hone your skills on the next few passages that demonstrate ways in which ciphers may be made more secure. As you go through each be sure to consider the following questions:

- How does the new method compare to those which we have previously considered, what does it do to make the message more secure?
- Each time we change the method in which messages are enciphered what doesn't change?
- Are there any properties that can not change? That is, are there rules about how English is put together that we can make use of?
- If you have suspicions about what a message, or part of a message, might say how can you use this to help you?
- What sort of data might we keep track of or how can we organize our thoughts to help us solve these puzzles?

Checkpoint 1.1.7. Cipher Text:

```
IL JVWF UVD AV TLU VM NYVZZLY ISVVK, HUK ALHJO AOLT
OVD AV DHY. HUK FVB, NVVK FLVTHU, DOVZL SPTIZ DLYL
THKL PU LUNSHUK, ZOVD BZ OLYL AOL TLAASL VM FVBY
WHZABYL; SLA BZ ZDLHY AOHA FVB HYL DVYAO FVBY
IYLLKPUN; DOPJO P KVBIA UVA; MVY AOLYL PZ UVUL VM
FVB ZV TLHU HUK IHZL, AOHA OHAO UVA UVISL SBZAYL
PU FVBY LFLZ. P ZLL FVB ZAHUK SPRL NYLFOVBUKZ PU
AOL ZSPWZ, ZAYHPUPUN BWVU AOL ZAHYA. AOL NHTL'Z
HMVVA: MVSSVD FVBY ZWPYPA, HUK BWVU AOPZ JOHYNL JYF
'NVK MVY OHYYF, LUNSHUK, HUK ZHPUA NLVYNL! - OLUYF
C, DPSSPHT ZOHRLZWLHYL
```

Checkpoint 1.1.8. Cipher Text:

```
BJKJB BJMFU UDKJB BJGFS ITKGW TYMJW XKTWM JYTIF DYMFY
XMJIX MNXGQ TTIBN YMRJX MFQQG JRDGW TYMJW GJMJS JJWXT
ANQJY MNXIF DXMFQ QLJSY QJMNX HTSIN YNTSF SILJS YQJRJ
SNSJS LQFSI STBFG JIXMF QQYMN SPYMJ RXJQA JXFHH ZWXJI
YMJDB JWJST YMJWJ FSIMT QIYMJ NWRFS MTTIX HMJFU BMNQJ
XFSDX UJFPX YMFYK TZLMY BNYMZ XZUTS XFNSY HWNXU NSXIF
DMJSW DAGDB NQQNF RXMFP JXUJF WJ
```

Hint.

- This is a quote by the same author as the previous cipher.
- The most common letters in the cipher text are `J`, `M`, and `X`.

Each of the previous messages used a simple shift cipher of which, in English, there are only twenty five (twenty six if you allow a shift of 0). However if we look at all the possible ways that we can rearrange twenty six letters we really have a lot more options.

- Assuming 26 letters in the alphabet, how many letters can you use to replace `a`? (For simplicity assume you could choose to replace `a` with itself if you were so inclined.)
- Once you have chosen a letter to use in place of `a`, how many choices for `b`?
- What about `c`?
- Going in order how many choices do you have for all the remaining orders?
- What happens if you multiply all these numbers of options together in order to get the total number of choices?

The number you should have come up with is 403 septillion 291 sextillion 461 quintillion 126 quadrillion 605 trillion 635 billion 584 million, or

$$403,291,461,126,605,635,584,000,000.$$

John Falconer in the seventeenth century described this as follows:

> *Schottus* demonstrates, (though the calculation in his book be not exact) that a thousand million of men in as many years could not write down all those different transpositions of the alphabet, granting every one should complete forty pages a day, and every page contain forty several positions: For if one writer in one day write forty pages, everyone containing forty combinations, 40 multiplied by 40, gives 1,600, the number he completes in one day, which multiplied by 366, the number (and more) of days in a year; a writer in one year shall compass 585,600 distinct rows. Therefore in a thousand million of years he could write
>
> $$585,600,000,000,000,$$
>
> which being multiplied by 1,000,000,000, the number of writers supposed, the product will be
>
> $$585,600,000,000,000,000,000,000,$$
>
> which wants of the number of combinations no less than
>
> $$348,484,017,332,394,393,600,000.$$
>
> [4, pages 5-6]

Comprehension Check:

- How many lines of cipher alphabets does Falconer say there could be on one page?
- How many pages a day does one person potentially write?

- How many men does Falconer suggest could be writing and for how long?
- Are these people able to in that time write out all the possible cipher alphabets?

Checkpoint 1.1.9. Imagine that instead of men writing out the cipher alphabets it was a computer. Suppose that a computer can produce one new cipher alphabet every 0.1 seconds (so 10 alphabets a second), how many seconds would it take for the alphabet to write out all $403,291,461,126,605,635,584,000,000$ alphabets? How many hours is that? How many years? (Assume, like Falconer, that there are 366 days in a year.)

Today we would call the huge number we calculated above 26!, which is read "26 factorial," and certainly takes up a lot less room.

Definition 1.1.10 Factorial. Given a counting number n we define **n-factorial** as the number of ways in which we can arrange n objects, we calculate it by taking the product of all the consecutive positive integers from n down to 1, which we write:

$$n! = n(n-1)(n-2)\cdots 2\cdot 1 \tag{1.1.1}$$

Note that for a variety of reasons $0! = 1$.

Checkpoint 1.1.11. Use the definition above to calculate 4!, 7!, and 10!.

We can make a message harder to decipher simply by using a more inventive method of encipherment than just a shift. The last exercise in this section was enciphered with a slightly trickier cipher than the previous ciphers so I put the spaces and punctuation back in. Also, while it is a harder cipher it is not completely random, there is a pattern to how the key was generated; so in the table below the message try filling in the cipher alphabet below the plain alphabet as you work, and see if you can spot a pattern.

Checkpoint 1.1.12. Cipher Text:

```
"EACMPA XMT EADGL Z SFGLD, PAKGLR XMTPQAJC SFZS RGCCGBTJSGAQ
ZLR RAJZXQ OTGSA GKNMQQGEJA SM CMPAQAA ZPA ZFAZR. GC XMT
BMTJR QAA SFAK BJAZPJX, LZSTPZJJX XMT BMTJR RM Z DPAZS RAZJ
SM DAS PGR MC SFAK ETS XMT BZL'S. XMT BZL MLJX QAA MLA SFGLD
BJAZPJX ZLR SFZS GQ XMTP DMZJ. CMPK Z KALSZJ UGQGML MC SFZS
ZLR BJGLD SM GS SFPMTDF SFGBI ZLR SFGL." - IZSFJAAL LMPPGQ
```

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

Table 1.1.13: Monoalphabetic Substitution Table

Hint.

- What letters are on their own in the cipher? Which letters can be on their own in plain English?
- The cipher letter `A` is most common, what does that tell you? Likewise, `S` is second most common so what might that mean?

- The word `QAA` appears twice on its own and once as the end of another word, what three letter plaintext words might look like this?
- After you have spent some time looking at the message and trying to identify some of the ciphertext start to fill in the copy of substitution table and see if you notice any patterns.

1.2 Arabic Numerical Ciphers

Mathematics played no role in the ciphers we examined in the previous section. In fact math played no role in cryptology at all until **Arabic scholars**[1] performed basic data analysis on the Arabic language in the 9th century CE. They quickly realized that their observations could be used to break the basic ciphers we have been considering. However, that is a topic for the next chapter.

In this section we will introduce some other work done by Arabic scholars. [9, vol. 3] We are going to look at some methods of enciphering text by first changing the text to numbers, some of these methods today would be called **affine ciphers**.

They began by associating each letter of the alphabet with a number in a manner similar to the following:

1	2	3	4	5	6	7	8	9
a	b	c	d	e	f	g	h	i
10	20	30	40	50	60	70	80	90
j	k	l	m	n	o	p	q	r
100	200	300	400	500	600	700	800	900
s	t	u	v	w	x	y	z	

Table 1.2.1: A Numerical Alphabet

Once this is done there are then a variety of ways in which we may encipher our message. Carefully read through these passages from *ibn ad-Durayhim's* treatise written in the 12^{th} century CE; try to get an idea of how his encipherments worked before going on.

> ”***5. On the replacement of letters using the decimally-weighted numerical alphabet:***
>
> - By substituting decimal numerical alphabet for letters in four different ways: by writing the numbers in words as pronounced; or by finger-bending, using the fingers to communicate the message visually to a recipient; or by writing the numbers as numerals such as writing *(mhmd: forty, eight, forty, four)*; or by giving the cryptogram a semblance of a page of a financial register.
> - By recovering the cryptogram numeral into a number of letters - a method of encipherment which involves more sophistication. There are many combinations that can be used in this method; for example in *(mhmd:*

[1] When math historians, or historians in general, speak of Arabic scholars it is a reference to the language and not the ethnicity or location of the scholars.

> *jl, fb, jl, ca)* or *(kk, ga, kk, bb)* . One can even form delusive words such as *(mhmd: lead, cad, deal, baa)*, or substitute two words for a letter, e.g. *(ali:* $\overline{\textit{dig fad}}$*,* $\overline{\textit{cab ab}}$*)*, in which case a line is to be drawn over two words to denote that they represent one letter.
>
> - By multiplying the number representing the letter by two, and so write *(mhmd: q, jf, q, h)* and *(ali: ob, jh)*, etc; or multiply it by three, thus writing *(mhmd: sk, kd, sk, jb)* and *(ali: rc, kg)*. Numbers can also be multiplied by four or five." [a] [9, vol. 3, pp. 69-70]
>
> [a] The examples here are very loosely based on the Arabic examples in the translation. The "mhmd" is Mohamed since Arabic is written without vowels, and for "ali" the a and l together are treated as a single letter.

Comprehension Check: (Be sure to reference Table C.0.2 as you try to answer these questions.)

- In the first paragraph how did *mhmd* become *forty, eight, forty, four*?
- In the second paragraph above, how did the author derive *jl* from *m* or $\overline{\textit{cab ab}}$ from *i* ?
- When converting letters to *delusive words*, *m* was enciphered as *lead* and *deal*, how did this work? Can you find another word to encipher it as?
- What letter(s) would be easier to translate into a variety of words? Which letters would be harder to change into words?
- In the third paragraph why is the *m* enciphered as a single *q* in the first example, but requires two letters, *sk* in the second?

Checkpoint 1.2.2. Demonstrate your understanding of the systems above by enciphering *"ibex"* using each of the methods described above.

1. By writing the letters as numerals.
2. By writing letters in combinations of two or more letters.
3. By tripling the values of the letters.

Hint. For all of these the first step is to use Table C.0.2 to translate the letters to numbers. Then you will need to in some way manipulate those numbers and/or translate them back to words and letters using the table. How specifically this happens depends on the cipher method, also for some methods there will be more than one answer.

Checkpoint 1.2.3. Encipher the word *"tan"* by doubling the values of the letters. In how many ways do you think you can encipher each letter if you are allowed to use up to four letters to represent each letter?

Hint. As an example b doubled is 4 which we could represent as a D or as AC or as CA or as BB (though the last one is kind of silly). Also, remember that you will need to use Table C.0.2.

Checkpoint 1.2.4. The following cipher text was enciphered using one of the above ciphers.

```
YU II J S JH ZT B J V V SK X S H J SQ T V B S H YU
IG B SU GAJ T JF B SM QO FD S JH OM JD UT KS EC B
ZX OO SLN V QS XVUS RJU SK EC J DB HJ H J YU JF B
QKU UY JH O MK FJ B XZ YVU J MO SU KS Q SQ RPK MQ
XV HJ O SK NP M D B F M NP Q B SQ M JF B VZT WXU B Q
```

Hint.

- The message is a quote from a famous Arabic poet and mathematician.
- There is at least one letter substitution which by necessity is always the same.
- Each pair or triple of letters represents only a single plaintext letter.
- The order of a pair or triple of letters doesn't change which letter it enciphers.
- As before you will need Table C.0.2.

Checkpoint 1.2.5. Enciphering by forming *delusive words* is potentially useful but also can also be very hard. See if you can convert the word "math" to delusive words.

Hint. Recall that before we could encipher "m" as either "LEAD" or "DEAL" this way. And, as with all of the ciphers here, you need to use Table C.0.2.

Reflection: Looking back at the exercises you have completed try to comment on the following questions:

- Looking at the number of different ways you could encipher some letters, like *t*, what advantage might these new ciphers have over the simpler ones we looked at previously?
- Why might having a large number of different ways to encipher a single letter be a disadvantage? (Note: A cipher in which a single character or block of characters may be enciphered in multiple ways in the same message is called **homophonic**.)
- Which of the methods described above do you think has the potential to be the most useful and why?
- Do you think these methods are more or less secure than the methods we looked at in the previous section? Or, are they about the same?
- In what way do you think you might be able to tweak one of the above methods in order to make it more effective?

How Shifty are You?

For each exercise where you need to encipher or decipher a message do enough work to demonstrate that you clearly understand how to do the job; say at encipher or decipher least 10 characters or write down the plain text alphabet and corresponding cipher text alphabet.**1**. Encipher the following using a shift cipher with a key of 9:

> "We've all heard that a million monkeys banging on a million typewriters will eventually reproduce the entire works of Shakespeare. Now, thanks to the internet, we know this is not true." - Professor Robert Silensky

2. Encipher “zoologist” by converting it to numerals with the *Numerical Alphabet Table* (Table C.0.2), multiplying by 5, and finally converting those numerals back to letters.

Did you run into any particular problem enciphering this message? Is there only one unique way to do it?

3. Encipher “dog” by converting it to numerals with the *Numerical Alphabet Table* (Table C.0.2), and then split those numeral up so as to form *delusive words*.

Did you run into any particular problem enciphering this message? Are there some letters that couldn’t be changed? Is there only one unique way to do it?

4. Another type of monoalphabetic substitution cipher is called **atabash** and first appeared in the bible. To encipher a message in atabash you replace *a* with *Z*, *b* with *Y*, *c* with *X*, and so on. Fill in the table below with the appropriate cipher alphabet for atabash. Now, try to use your table to encipher the following message with atabash.

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

Table 1.2.6: Monoalphabetic Substitution Table

> “Mistakes are a part of being human. Appreciate your mistakes for what they are: precious life lessons that can only be learned the hard way. Unless it’s a fatal mistake, which, at least, others can learn from.” - Al Franken, “Oh, the Things I Know”, 2002

5. Decipher the message that was enciphered with a shift of 10.

```
XYGDR OICRY GIYER YGNOD OBQOX DCDKU OYEDL VYYNC DKSXC KZBOD

DIFSY VOXDS WKQOD ROBOS DRSXU SPIYE FOQYD KDCRS BDGSD RKLVY

YNCDK SXKVV YFOBS DWKIL OVKEX NBISC XDIYE BLSQQ OCDZB YLVOW

WKILO IYECR YEVNQ ODBSN YPDRO LYNIL OPYBO IYENY DROGK CRTOB

BICOS XPOVN
```

6. Decipher this message that was enciphered using a shift of 25.

```
NMDRG NTKCF TZQCZ FZHMR SOQDZ BGHMF SNXNT MFODN OKDRT BBDRR

HMSGD BTRSN LZQXE NQLZR SGDLZ HMZHL HMKHE DSGDL NRSHL ONQSZ

MSLNS HUDEN QVNQJ HMRBG NNKZM CHMKH EDHRO KDZRT QDHMV NQJOK

DZRTQ DHMHS RQDRT KSZMC SGDJM NVKDC FDNES GDUZK TDNES GDQDR

TKSSN SGDBN LLTMH SXZKA DQSDH MRSDH M
```

7. Decipher this message which was enciphered using atabash (see Exercise 1.4).

```
RMZXL NKOVG VOBIZ GRLMZ OHLXR VGBGS VYVHG LUFHD LFOWY VGVZX

SVIHZ MWGSV IVHGL UFHDL FOWSZ EVGLH VGGOV ULIHL NVGSR MTOVH

HYVXZ FHVKZ HHRMT XRERO RAZGR LMZOL MTUIL NLMVT VMVIZ GRLMG

LGSVM VCGLF TSGGL YVGSV SRTSV HGSLM LIZMW GSVSR TSVHG IVHKL

MHRYR ORGBZ MBLMV XLFOW SZEVO VVRZX LXXZ
```

8. Another type of monoalphabetic substitution cipher is called a **keyword cipher**. The key for this cipher consists of two pieces, a *key word* and a *key letter*. You write the key word in the cipher alphabet starting underneath the key letter and then write the remaining letters of the alphabet in order after that. So, if the key word is *ZEBRA* with key letter *f* we start as follows:

plain	a	b	c	d	e	f	g	h	i
CIPHER	____	____	____	____	____	Z	E	B	R
plain	j	k	l	m	n	o	p	q	r
CIPHER	A	____	____	____	____	____	____	____	____
plain	s	t	u	v	w	x	y	z	
CIPHER	____	____	____	____	____	____	____	____	____

Table 1.2.7: Keyword Cipher: Word=*ZEBRA*, Letter=*f*

Fill in the rest of the alphabet and then use it to decipher this message.

```
MBYEH HXWBK RLMRU GLBHN DXVYP UKYHZ FUMBY FUMRW RUGLU GXUDD

MBHLY PBHFU CYYFI MSIKH IBYWR YLMBY XUGEY KUDKY UXSYQ RLMLM

BUMFU MBYFU MRWRU GLBUO YFUXY UWHOY GUGMP RMBMB YXYOR DMHXU

KCYGM BYLIR KRMUG XWHGZ RGYFU GRGMB YVHGX LHZBY DDLMU NENLM

RGY
```

9. Setup a the blank monoalphabetic sunstitution table in the same way as Table 1.2.7 with a plaintext and ciphertext alphabet for a keyword cipher with key letter *d* and keyword *ARNOLD*. Then use it to encipher this quote:

> "Strength does not come from winning. Your struggles develop your strengths. When you go through hardships and decide not to surrender, that is strength." - Arnold Schwarzenegger

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

Table 1.2.8: Monoalphabetic Substitution Table

Chapter 2

Attacking the Alphabet

In this chapter look at the contributions of Arabic speaking scholars and others mostly to analysis:

- Arabic analysis
- Analysis in the seventeenth century, Falconer

2.1 Arabic Analysis of the Alphabet

Definition 2.1.1 Cryptanalysis. Cryptanalysis is the process by which we try to determine the meaning of a message without the aid of a key. We will sometimes describe this as **decrypting** a message as opposed to **deciphering** a message which is what we do when we know the key.

The first place math was employed in the science of cryptology was in the analysis of languages. The following is from *al-Kindi's* treatise on cryptanalysis written around the year 873 CE. Try to read through it carefully and then consider the questions that follow it.

> ”***Algorithms of Cryptanalysis***
>
> So we say, the enciphered letters are either in numerical proportions, that is poetry -because poetic meter, ipso facto, sets measures to the number of letters in each line-, or they are not. Non- poetry can be cryptanalyzed using either quantitative or qualitative expedients.
>
> The ***quantitative*** expedients include determining the most frequently occurring letters in the language in which cryptograms are to be cryptanalyzed. If vowels functioned as the material from which any language is made, and non-vowels functioned as the shape of any language, and since many shapes can be made from the same material, then the number of vowels in any language would be greater than non-vowels. For instance, gold is the material of many shapes of finery and vessels; it may cover crowns, bangles, cups, etc.. The gold in these realizations is more than the shapes made of it. Similarly, the vowels which are the material of any kind of text are more than the non- vowels in any language. I mean by vowels the letters: (a), (y or i or e) and (o or u). Therefore the vowels in any language, inevitably, exceed in

> number the non-vowels in a text of that language. It happens that in certain languages some vowels are greater in number than some other vowels, while non-vowels may be frequent or scarce according to their usage in each language, such as the letter (s), of which frequency of occurrence is high in Latin.
>
> Among the expedients we use in cryptanalyzing a cryptogram if the language is already known, is to acquire a fairly long plaintext in that language, and count the number of each of its letters. We mark the most frequent letter ”first”, the second most frequent ”second”, and the following one ”third”, and so forth until we have covered all its letters. Then we go back to the message we want to cryptanalyze, and classify the different symbols, searching for the most frequent symbol of the cryptogram and we regard it as being the same letter we have marked ”first” -in the plaintext-; then we go to the second frequent letter and consider it as being the same letter we have termed ”second”, and the following one ”third”, and so on until we exhaust all the symbols used in this cryptogram sought for cryptanalysis.
>
> It could happen sometimes that short cryptograms are encountered, too short to contain all the symbols of the alphabet, and where the order of letter frequency cannot be applied. Indeed the order of letter frequency can normally be applied in long texts, where the scarcity of letters in one part of the text is compensated for by their abundance in another part.
>
> Consequently, if the cryptogram was short, then the correlation between the order of letter frequency in it and in that of the language would no longer be reliable, and thereupon you should use another, ***qualitative*** expedient in cryptanalyzing the letters. It is to detect in the language in which cryptograms are enciphered the associable letters and the dissociable ones. When you discern two of them using the letter order of frequency, you see whether they are associable in that language. If so, you seek each of them elsewhere in the cryptogram, comparing it with the preceding and following dissociable letters by educing from the order of frequency of letters, so as to see whether they are combinable or non-combinable. If you find that all these letters are combinable with that letter, you look for letters combinable with the second letter. If found really combinable, so they are the expected letters suggested by the combination and non-combination of letters, and also by their order of frequency. Those expected letters are correlated with words that make sense. The same procedure is repeated elsewhere in the ciphertext until the whole message is cryptanalyzed.”
> [9, vol. 1, pp. 121-123]

Comprehension Check:

- What do you think the author means when he says “vowels function as the material of a language”?

- In what way then do the “non-vowels function as the shape”?
- He also says that there are more vowels than non-vowels, how many vowels are in this sentence you are reading right now? Were there more vowels? If not then how might his statement still be true?
- How does the gold in his analogy function like the vowels?
- Finally, how do the author’s comments compare to your experiences in section Section 1.1?

What al-Kindi is describing above is what we now call **frequency analysis** which is the first step in cryptanalysis.

Definition 2.1.2 Frequency Analysis. Basic ***Frequency Analysis*** is the process of counting the characters in a text in order to determine how many of each character there are relative to the entire length of the text. This is typically the first step in the cryptanalysis, the process of breaking an unknown cipher or code.

Checkpoint 2.1.3. Try following al-Kindi’s directions in paragraph three above. Use the *n-gram counter* below to count the number of times each letter appears in the following paragraph and with what frequency, then plot the frequency of each character on the chart below.

> “In the year 1878 I took my degree of Doctor of Medicine of the University of London, and proceeded to Netley to go through the course prescribed for surgeons in the army. Having completed my studies there, I was duly attached to the Fifth Northumberland Fusiliers as Assistant Surgeon. The regiment was stationed in India at the time, and before I could join it, the second Afghan war had broken out. On landing at Bombay, I learned that my corps had advanced through the passes, and was already deep in the enemy’s country. I followed, however, with many other officers who were in the same situation as myself, and succeeded in reaching Candahar in safety, where I found my regiment, and at once entered upon my new duties.” - *A Study in Scarlet*, Sir Arthur Conan Doyle [3]

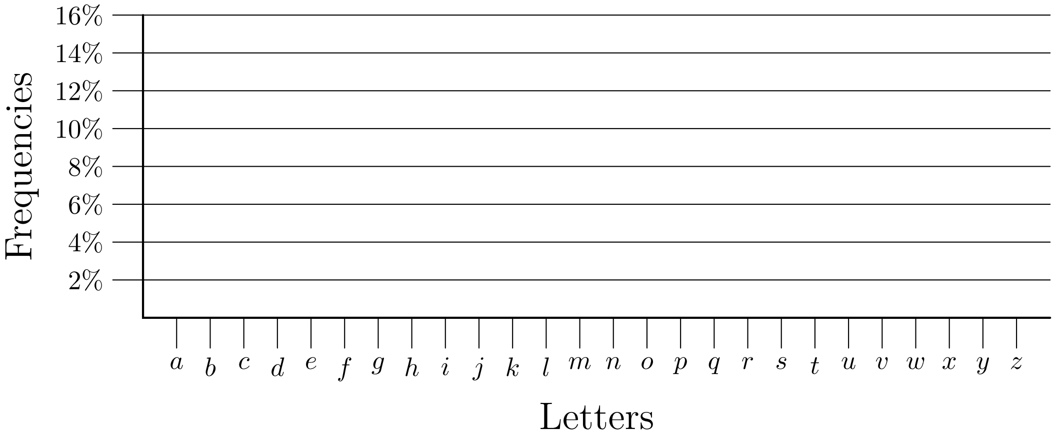


Figure 2.1.4: Axes for Mapping Letter Frequencies

Checkpoint 2.1.5. Below is the same paragraph as in Checkpoint 2.1.3 only now enciphered with a shift of three. As above, use the *n-gram counter* below to count the number of times each letter appears and with what frequency, then plot the frequency of each character on the chart below.

```
LQWKH BHDUL WRRNP BGHJU HHRIG RFWRU RIPHG LFLQH RIWKH XQLYH
UVLWB RIORQ GRQDQ GSURF HHGHG WRQHW OHBWR JRWKU RXJKW KHFRX
UVHSU HVFUL EHGIR UVXUJ HRQVL QWKHD UPBKD YLQJF RPSOH WHGPB
VWXGL HVWKH UHLZD VGXOB DWWDF KHGWR WKHIL IWKQR UWKXP EHUOD
QGIXV LOLHU VDVDV VLVWD QWVXU JHRQW KHUHJ LPHQW ZDVVW DWLRQ
HGLQL QGLDD WWKHW LPHDQ GEHIR UHLFR XOGMR LQLWW KHVHF RQGDI
JKDQZ DUKDG EURNH QRXWR QODQG LQJDW ERPED BLOHD UQHGW KDWPB
FRUSV KDGDG YDQFH GWKUR XJKWK HSDVV HVDQG ZDVDO UHDGB GHHSL
QWKHH QHPBV FRXQW UBLIR OORZH GKRZH YHUZL WKPDQ BRWKH URIIL
FHUVZ KRZHU HLQWK HVDPH VLWXD WLRQD VPBVH OIDQG VXFFH HGHGL
QUHDF KLQJF DQGDK DULQV DIHWB ZKHUH LIRXQ GPBUH JLPHQ WDQGD
WRQFH HQWHU HGXSR QPBQH ZGXWL HV
```

A Study in Scarlet, Sir Arthur Conan Doyle [3]

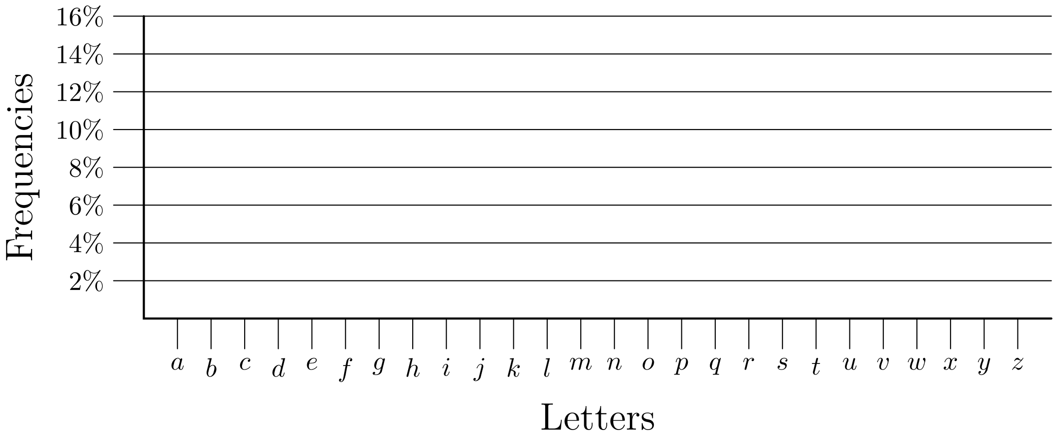


Figure 2.1.6: Axes for Mapping Letter Frequencies

How did the plot change? In what ways did the plot not change?

Checkpoint 2.1.7. Find a large sample of normal English text (at least 500 characters) and repeat what you did in Checkpoint 2.1.3; that is use the *n-gram counter* below to count the number of times each letter appears and with what frequency. Then plot the frequency of each character on the chart below.

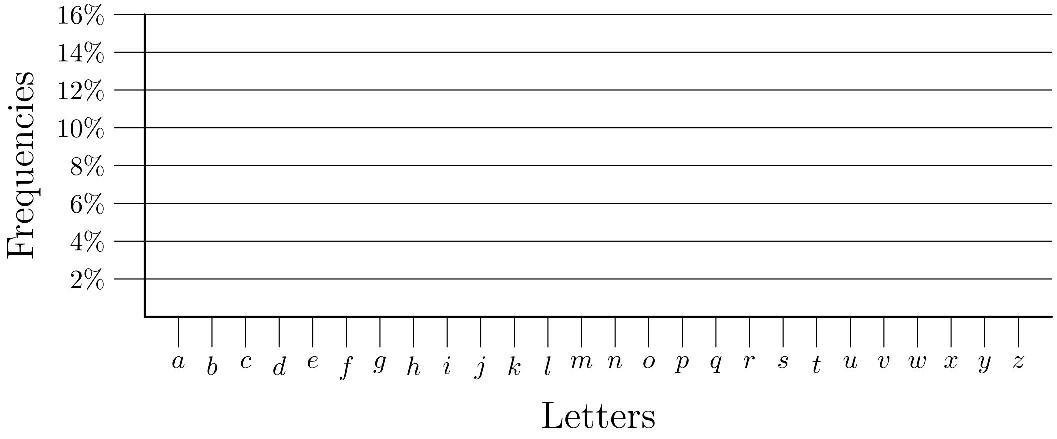


Figure 2.1.8: Axes for Mapping Letter Frequencies

Checkpoint 2.1.9. Take the English text you used in Checkpoint 2.1.7 and encipher it with a shift cipher. Analyze the ciphertext as you did in Checkpoint 2.1.5. How did the plot change? In what ways did the plot not change?

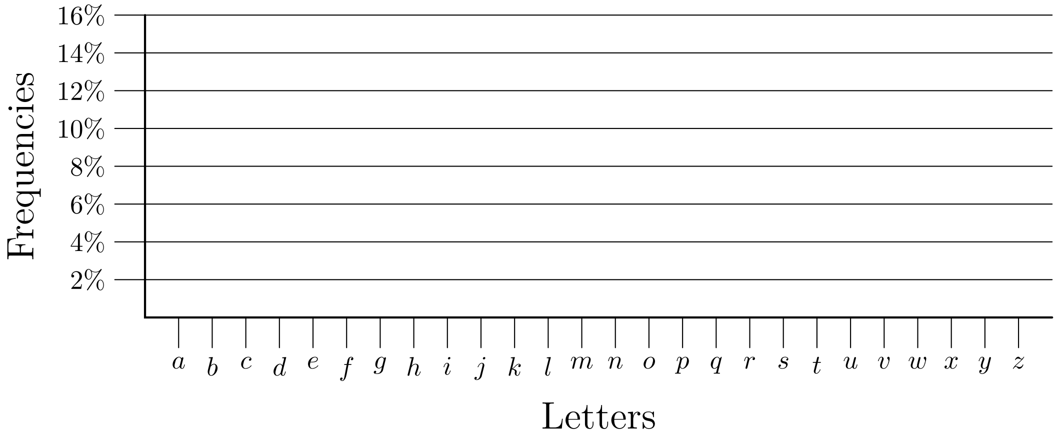


Figure 2.1.10: Axes for Mapping Letter Frequencies

Checkpoint 2.1.11. Now use the n-gram counter to find the letter frequencies for the letters in this cipher text.

```
XLMWM EFPSG OSJVI PEXMZ IPCRS VQEPI RKPMW LXIBX LSTIJ YPPCA
LIRCS YEREP CDIXL MWCSY AMPPW IIXLE XIZIR ALIRX LMWMW IRGMT
LIVIH XLIPI XXIVJ VIUYI RGMIW WXECX LIWEQ IEWPS RKEWX LIGMT
LIVMW QSRSE PTLEF IXMGM RTEVX MGYPE VJSVE WLMJX GMTLI VXLIP
IXXIV JVIUY IRGMI WIZIR QEMRX EMRXL IWEQI VIPEX MZITS WMXMS
RXSIE GLSXL IVNYW XWPMH EPSRK PMOIX LIPIX XIVWX LMWQE OIWWY
GLEGM TLIVZ IVCIE WCXSW TSXER HGVEG O
```

Plot the frequencies you found using Figure C.0.3. If you compare the shape of the chart you just made to the chart for normal English which you made previously in Checkpoint 2.1.3 do you notice any similarities? Can you use this to try and decrypt this message?

N-Gram Counter:

To use the n-gram counter copy and paste the text you wish to analyze into the input box, and select 1 for N since we are analyzing single letters.

[INTERACTIVE]

Figure 2.1.12: N-Gram Analysis Tool

Substitution Cipher Tool:

To use the substitution cipher tool to encipher a message leave the plain text alone and enter the corresponding ciphertext in the box labeled cipher. For a simple shift cipher you can put the alphabet into the cipher box in the regular order and then use the shift drop down menu to select your desired shift.

[INTERACTIVE]

Figure 2.1.13: Substitution Cipher Tool

Checkpoint 2.1.14. Repeat what you did before in Checkpoint 2.1.7 with text from a variety of sources. Be sure to try both long and short pieces of text. Do you agree with the al-Kindi's statements about shorter pieces of text? Finally, make a table of your results for future reference.

2.2 (*)Alberti's Approach

Text based on the work of Leon Battista Alberti will go here.

2.3 Falconer's Approach

In 1685 John Falconer, who we met briefly in Section 1.1, wrote the second ever published text in English on the subject of cryptology; it has the rather lengthy title *Cryptomenysis patefacta; or, the art of secret information disclosed without a key. containing plain and demonstrative rule, for decyphering.* Beyond this work little is known about him for certain. A descendant of his writing in the nineteenth century claimed that John was the personal cipher keeper for King James II and followed the king into exile in France where he died. In slim support of this the book is dedicated to King James II and addressed to the king's secretary of state. However, no readily available documented evidence exists to independently corroborate his descendant's claims, and in fact there are some tax records for John and his wife, Mary Dalmahoy, in Edinburgh Scotland, from 1695, over six years after James fled to France. [10]

In his text John Falconer presents a wide variety of ciphers and, for most of them suggestions, for how we might attack them. In this section we will focus on his directions for how to break, or attempt to break, what we now call monoalphabetic substitution ciphers.

2.3.1 Step 1:

First, Distinguish the Vowels from the Consonants.

1. And first, the vowels generally discover themselves by their frequency; for because they are but few in number, and no word made up without some of them, they must frequently be used in any writing.
2. Where you find any character or letter standing by it self, it must be a vowel.
3. If you find any character doubled in the beginning of a word, in any language it is a vowel, as *Aaron, Eel, Jilt, Oogala, Vulture, etc.*, except for some English proper names, as Llandaff or Lloyd.
4. In monosyllables of two letters you may distinguish it from the consonant joined with it by its frequency.
5. In a word of three letters beginning and ending in the same letter the vowel is probably included.
6. When you find a character doubled in the middle of a word of four letters, 'tis probably the vowel *e* or *o*.
7. In Polysyllables, where a character is double in the middle of the word, it is for the most part a consonant; and if so, the precedent letter is always a vowel, and very often the following.[4, pp. 8-9]

Comprehension Check:

- Do some of these sound familiar from what you read in Section 2.1?
- Which of these are new? Do any of those seem strange to you?
- In what ways does Falconer say that one, two, and three letter words can help us?
- According to Falconer when might we expect a double letter to be a vowel and when might it be a consonant?
- Try to find out why Falconer says *"Jilt"* and *"Vulture"* begin with a double vowel.

For the following *Checkpoints* find a large sample of normal English text (you can use the one you found for the chekpoints in Section 2.1, but something bigger may be better). You will also want to use the n-gram counting and word counting tools below.

Checkpoint 2.3.1. In your text, what letters are most common? How many vowels are there? How many consonants are there?

Checkpoint 2.3.2. What are the one, two, and three letter words in the text you found to analyze? Do they follow the pattern that Falconer described? Are any of them more common than others?

Checkpoint 2.3.3. Look at your text for words with double letters, do they behave the way that Falconer says they should?

N-Gram Counter:

[INTERACTIVE]

Figure 2.3.4: N-Gram Analysis Tool

Word Counter:

[INTERACTIVE]

Figure 2.3.5: Word Counter Tool

2.3.2 Step 2:

Secondly, Distinguish the Vowels from Themselves..

1. Compare their frequency, and *e*, as we observed before, is generally the most used in the English tongue, next *o*, then *a* and *i*; but *u* and *y* are not so frequently used as some of the consonants.
2. It is remarkable that amongst the vowels, *e* and *o* are often doubled, the rest seldom or never.
3. *e* is very often a terminal letter, and *y* terminates words, but they are distinguishable, because there is no proportion to their frequency: *o* is not often in the end of words, except in monosyllables.
4. *e* is the only vowel that can be doubled in the end of an English word, except *o* in *too*, etc.
5. You may consider which of the vowels, in any language, can stand alone, as *a*, *i*, and sometimes *o* in English, *a*, *e*, *o*, in Latin or *i* the imperative of *eo*.[4, pp. 9-10]

Comprehension Check:

- Do you agree with his comments about *e, o* and the other vowels? Why or why not?
- Can you think of examples of or exceptions to his third and fourth comments above?
- Can you think of or find an example of *o* standing alone in English?

Checkpoint 2.3.6. Test out Falconer's observations about vowels using the same sample of text that you analyzed in exercise Checkpoint 2.3.1. Do his statements hold true or has the language changed since 1685?

2.3.3 Step 3:

Distinguish the Consonants Amongst Themselves..

1. As before observe their frequency. Those of most use in English are *d, h, n, r, s, t*, and next to those may be

reckoned *c, f, g, l, m, w*, in third rank may be placed *b, k, p*, and lastly *q, x, z*...

2. You may consider which consonants may be doubled in the middle or end of words.
3. What are terminal letters, etc.
4. The number and nature of consonants and vowels that fall together, or do usually fall together.[4, p. 10]

Checkpoint 2.3.7. What consonnts does Falconer say are most common? In the text you have been analyzing which are most common?

Checkpoint 2.3.8. Consider statement two above and look at the same text you used previously, which consonants are commonly doubled in the middle and at the end of words?

Checkpoint 2.3.9. In your sample of text which consonants do you commonly see at the end of words? Are there some that, when they do show up, are almost always at the end of a word? Are there any that are almost never at the end of a word?

Checkpoint 2.3.10. Falconer's first three pieces of advice are similar to what we have looked at before; his fourth comment is something a little new. Using the same text which you analyzed before set the *n-gram counter* to count two letter combinations in your text ($N = 2$). Once you have the count take note of which combinations of letters appear most often and for each vowel which two or three consonants it is most often paired with.

2.3.4 Step 4:

Additional Observations.

1. A word of three letters, beginning and ending with the same, may be supposed *did*
2. A word consisting of four characters, with the same letter in the beginning and end, is probably *that* or *hath*
3. A word consisting of five letters, when the second and last are the same, is commonly *which*, though it may be otherways, as in *known*, *serve*, etc. And you may judge of the truth of such suppositions by the frequency of the letters in the word supposed.

Next you may compare words one with another, as *on* and *no*, each being the other reversed; so *of* and *for*, the last being the first reversed with the addition of a letter; for and from will discover each other, etc.

You may also likewise observe some of the usual propositions and terminations of words, such as *com, con, ing, ed*, etc. Note that *t* and *h* are often joined in the beginning and end of English words, and sometimes in the middle.[4, pp. 11-12]

Comprehension Check:

- Can you think of more examples of three or four letter words beginning and ending with the same letter?

- What about five little words in which the second and last letters are the same?

- How many words like *on* and *no* or *of* and *for* can you think of?

- Compared with other suggestions we have looked at, how practical do these seem? Do they seem a little too subtle?

Checkpoint 2.3.11. Looking one last time at the text which you have been using identify common prefixes and suffixes (what Falconer called *propositions and terminations*) within the text. Are you finding any of the examples which he gave?

Bringing it all Together

For exercises Exercise 2.1 to Exercise 2.4 you will review the ideas we have explored in this chapter. When you are done you should have generated for yourself a guide which you can use when you are attempting to analyze a piece of cipher text. To begin find a new piece of relatively typical English text which is at least 1000 characters long. Then, you can use the *n-gram counter*, *word counter*, and *string counter* at the bottom of the page to help you examine the text for the following information:

1. letter frequencies

2. **bi-gram** (two letter combination) and **tri-gram** (three letter combination). frequencies

3. Common words

4. prefixes and suffixes

Now, finally, put together all the information you have just gathered into a neat one page analysis guide. You should type this up neatly, in a reasonably sized font. If you want you can print out a copy of Figure C.0.4 and use it to help organize the information. It should include:

- A list of frequencies for all the individual letters in the alphabet.

- The frequencies for at least the top 25 bi-grams and tri-grams.

- A list of the top 25 most common words.

- A list of your top 25 most common prefixes and suffixes.

- And, any other types of information you think might be useful which you have looked at in this chapter.

5. Use the skills we have been practicing in class together with the cheat sheet you just put together in order to crack this cipher.

```
LU ZOO GSZG RH TLLW, HFYORNRGB RH HFKIVNV. HFXXVVWRMT RH GSV
XLNRMT GLTVGSVI LU ZOO GSZG RH YVZFGRUFO. UFIGSVIZMXV RH GSV
ZTIVVNVMG LU ZOO GSZG RH QFHG. KVIHVEVIZMXV RH GSV ULFMWZGRLM
LU ZOO ZXGRLMH. - OZL GAF
```

Hint. Be sure to look at repeated common words. Also, as you uncover cipher letters try writing them down in Table C.0.1 and looking for a pattern.

6. Use the skills we have been practicing in class together with the cheat sheet you just put together in order to crack this cipher.

```
SNE ORGGRE VF VG GB QNER ZVTUGL GUVATF, GB JVA TYBEVBHF GEVHZCUF,
RIRA GUBHTU PURPXRERQ OL SNVYHER ... GUNA GB ENAX JVGU GUBFR CBBE
FCVEVGF JUB ARVGURE RAWBL ABE FHSSRE ZHPU, ORPNHFR GURL YVIR VA N
TENL GJVYVTUG GUNG XABJF ABG IVPGBEL ABE QRSRNG. - GURBQBER
EBBFRIRYG
```

Hint. Be patient, the letter frequencies are ... abnormal.

7. Use the skills we have been practicing in class together with the cheat sheet you just put together in order to crack this cipher.

```
WKSTD QJUTJ OTKLE JGGJI DQUQU SFFYS CUDRL PSTKL PTKSI SKDIR

PSIEL DIQJF VDICS MPJHF LGJOT KDQQJ PTTKL CPSIR TKDIC DQTJH

LSHFL TJPLS QJIHS EBWSP RTKST DQSVL PYUQL OUFSE EJGMF DQKGL

ITSIR SVLPY LSQYJ ILHUT MLJMF LRJIJ TMPSE TDELD TGUEK DITKL

LVLPY RSYSO OSDPQ JOFDO LDTDQ GJPLU QLOUF TJPLS QJIOJ PWSPR

SIRQJ TKLJT KLPEJ GLQTJ HLILC FLETL RTKLP LSPLO DOTYW KJESI

PLSQJ IQYIT KLTDE SFFYO JPJIL WKJES IPLSQ JISIS FYTDE SFFYG

JQTML JMFLD OYJUR LQEPD HLSTP SDIJO LVLIT QTJTK LGWDF FTLFF

YJUWK STTKL PLQUF TWJUF RHLTK LYESI MUTTK JQLLV LITQT JCLTK

LPDIT KLDPG DIRQS IRSPC ULOPJ GTKLG TKSTQ JGLTK DICWD FFEJG

LTJMS QQTKL PLSPL OLWML JMFLK JWLVL PWKJD OYJUT JFRTK LGSPL

QUFTW JUFRH LSHFL TJLVJ FVLOP JGTKL DPJWI DIILP EJIQE DJUQI

LQQWK STTKL QTLMQ WLPLW KDEKF LRUMT JTKST PLQUF TTKDQ MJWLP

DQWKS TDGLS IWKLI DTSFB JOPLS QJIDI CHSEB WSPRJ PSISF YTDES

FFYSQ TURYD IQESP FLTQK LPFJE BKJFG LQ
```

Hint. This is a quote from a famous fictional nineteenth century detective and makes use of a keyword cipher.

8. Consider the following comment from John Falconer:

> I have not meddled here with any language but English; ... However, by a little practice of decyphering in one language, you may decipher an epistle in any, even tho the plain speech itself be a mystery to you, if you first observe the frequency of the letters, the terminal letters, what letters can be doubled in the beginning, middle, and end of words; and such general rules.[4, p. 12]

Find a large piece of typical text written in a language other than English; it needs to be native to that language not a translation from another language. Perform an analysis of this text and then create a cheat sheet just like you did for English (exercises Exercise 2.1 to Exercise 2.4). Finally, encipher a message written in this new language which your classmates can try to decipher using your cheat sheet.

9. A lot of the general advice that Falconer gave us was good, but his specifics didn't match what we expect to see when we read a more modern text. Find a typical piece of text or literature from seventeenth century England (or better yet Scotland) and analyze this text to create a cheat sheet like the one you did for modern English (exercises Exercise 2.1 to Exercise 2.4). How do your findings match what Falconer told us to expect?

N-Gram Counter:

[INTERACTIVE]

Figure 2.3.12: N-Gram Analysis Tool

Word Counter:

[INTERACTIVE]

Figure 2.3.13: Word Counter Tool

String Counter Tool: You can use this tool to look for specific combinations of letters in you text rather than gathering gerneral information. This is helpful for looking for prefixes, suffixes, and double letters.

[INTERACTIVE]

Figure 2.3.14: String Counter Tool

Chapter 3

Mixing Things Up

In this chapter look at real improvements to cryptology:

- Polyalphabetics (Alberti)
- Autokey Cipher (Vigenère)
- Keyed Columnar Ciphers and Codes (Falconer)
- Thomas Jefferson's contributions???

3.1 Alberti's Great Idea

Objectives

- Alberti's Biography
- Alberti's Polyalphabetic
- Modern/Vigenère's Polyalphabetic

3.1.1 Alberti's Biography

For now see his biography on the MacTutor History of Mathematics Site: Leon Battista Alberti

3.1.2 Alberti's Polyalphabetic Cipher

Definition 3.1.1 Polyalphabetic Substitution Cipher. A **polyalphabetic substitution cipher** is a cipher in which a single plaintext letter maybe replaced by several different ciphertext letters, groups of letters, or symbols and every letter, group of letters, or symbol in the ciphertext may represent more than one plain text letter.

One of Leon Battista Alberti's greatest contributions to cryptology was not his cryptanalysis skills but an entirely new kind of cipher. Many efforts had been and would be made to shore up monoalphabetic substitution ciphers. Some of these efforts worked well enough but they were never sufficiently different to make them immune to the sorts of attacks we have studied before.

Alberti made something truly different he invented a **polyalphabetic substitution cipher**. This was a cipher that did not just replace the plain text alphabet with a single separate alphabet but instead used multiple enciphering alphabets to create greater confusion and security.

The key to Alberti's new system was a device he called a **"formula"** . This was a cipher disk consisting of a fixed outer disk and a mobile inner disk. In his basic description of his new cipher he would use the outer ring of upper case letters for his plain text and the inner mobile ring of lower case letters for his cipher text.

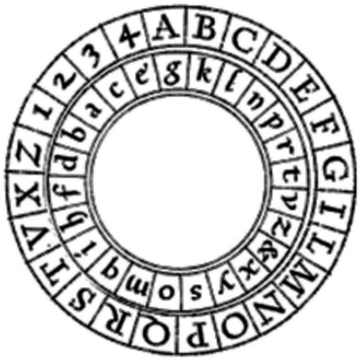

1 2 3 4 A B C D E F G I L M N O P Q R S T V X Z
a c e g k l n p r t v z & x y s o m q i h f d b

Figure 3.1.2: Alberti's Cipher Disk or **"Formula"**

> "Say for example we mutually establish *k* as the index of the mobile circle. Writing, the formulae are positioned at will, say such *k* lies under the uppercase *B* [Figure 3.1.2] and the next letter corresponds to the letter that comes next. In writing to you, I will first put the uppercase *B* under which lies the index *k* in the formula; this is to signal you far away, wanting to read what I have written, that you should set up the twin formula in your keeping, positioning the mobile circle so that the *B* sits over the index *k*. Then all of the rest of the lowercase letters present in the coded text will take their meaning and sound from those of the fixed circle above them."
>
> "After I have written three or four words I will mutate the position of the index in our formula, rotating the disk let's say, so that the index *k* falls below the upper case *R*. Then in this missive I write an uppercase *R* to indicate that *k* no longer refers to *B*, but to *R*, and the letters that will follow will assume new meanings." - Alberti [14, p. 181]

Comprehension Check:

- If you compare the inner and outer rings of the formula (cipher disk) what difference do you notice? Why do you think Alberti might have made these choices? (For reference I made an updated version in the appendix where I made similar choices, Figure C.0.6.)

- What step in the process above stops this from simply being a shift cipher?

- How does the recipient of a message using Alberti's method know how to set their disk and when they should change it?

- Is there an obvious weakness in the way that Alberti suggests using his system?

Checkpoint 3.1.3. If you have not done so already make a copy of Alberti's formula [Figure 3.1.2] so that you can use it for enciphering and deciphering. Now encipher "See my first message" using an index of c initially underneath the G, but shift it underneath V after the first two words.

Hint.

- Remember to line up the index c with the capital letters used of the key, here G and V.
- For Alberti, uppercase letters (the outer ring) are the plaintext and lowercase letters are the ciphertext.
- Because of what was and was not included on the disk you will need to make some choices.

Checkpoint 3.1.4. Suppose that `B lgiq R yam V ydme` was enciphered using an index of k as directed by Alberti, what does it say?

Hint.

- First identify the three key letters used in this cipher.
- Remember to line up the index k with the capital letters used of the key, and to move the index when the key letter changes.
- For Alberti, uppercase letters (the outer ring) are the plaintext and lowercase letters are the ciphertext.

Checkpoint 3.1.5. Team up with a partner and agree on a lower case index. Now each of your should try enciphering a short message, only five or six words, be sure to adjust your formula after every couple of words. Then exchange messages with your partner and decipher each others messages.

3.1.3 Vigenère's Cipher

In spite of the fact that Alberti invented the polyalphabetic cipher his is not the name associated with it, instead that honor goes to the Frenchmen ***Blaise de Vigenère***. Vigenère made a careful study of the cryptographers who had come before him and in 1586 he published the culmination of his work in his *Traicté des chiffress, ou secrètes manières d'escrire*[13]. In the Traicté he described a wide variety of older ciphers and his improvements to them. Among these was his take on Alberti's polyalphabetic cipher.

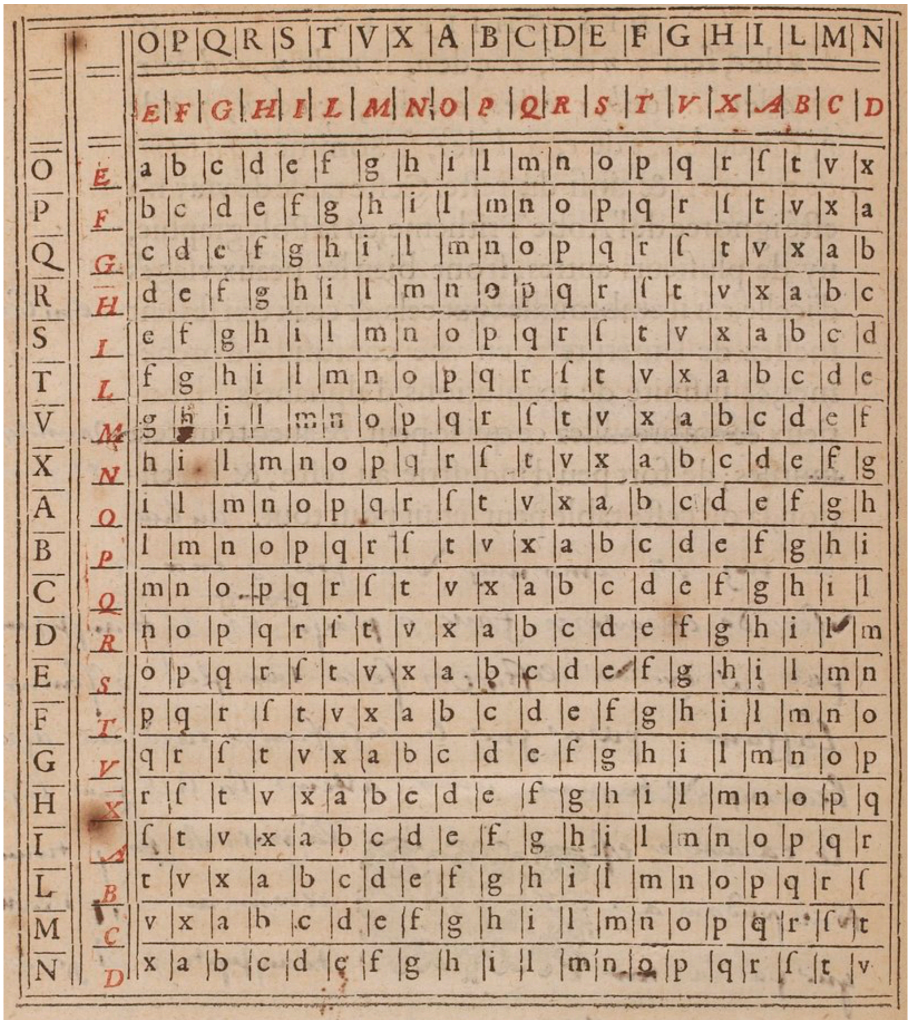
One of the key advances that Vigenère made was to replace the disk with a table [Figure 3.1.6, [13, p. 50b (102)][1]]. This made it easier to vary between alphabets without having to move inner and outer disks to different positions.

> "... set the capitals which run across the top [of the table] for the message to be conveyed & those that run perpendicularly down the left for the keys. I have put two rows of capitals here, one black and the other red, to show that alphabets of the text as, the keys, may be transposed and changed in as many ways as you want to keep knowledge

[1] In the page designation "p. 50b (102)" the 102 is the page in the pdf document while the 50b is the page in the manuscript.

of them from all others except ones correspondents. Therefore, to encipher the saying we used previously *"au nom de l'eternel,"* with the key *"le jour obscur"* proceed in this way; from *a* in the alphabet across the top in red, come to the row with *l* and the box with *b*: u from e will be q: n from i, n: o from o, s: m from u, a: d from r, m: e from o, i: l from b, c: e from s, o: t from c, n: e from u, q: r from r, c: n from l, o: e from e, a: l from i, l. Altogether we get *bqnsamiconcoal* & so on for the remainder." [13, pp. 49b, 50, 50b (100-102)], [8, p. 110][a]

[a] Most of this translation is due to Mendelsohn [8], I have only filled in with my own at the end which he had not translated.



		O	P	Q	R	S	T	V	X	A	B	C	D	E	F	G	H	I	L	M	N
		E	F	G	H	I	L	M	N	O	P	Q	R	S	T	V	X	A	B	C	D
O	E	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x
P	F	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a
Q	G	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b
R	H	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c
S	I	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d
T	L	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e
V	M	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f
X	N	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g
A	O	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h
B	P	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i
C	Q	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l
D	R	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m
E	S	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n
F	T	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o
G	V	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p
H	X	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q
I	A	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r
L	B	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ
M	C	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t
N	D	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v

Figure 3.1.6: Vigenère's Tableau

Comprehension Check:

- How did the *a* in the saying *"au nom de l'eternel"* get replaced with a *b* in the ciphertext *bqnsamiconcoal*?

- How did the *u* in the saying *“au nom de l’eternel”* get replaced with a *q* in the ciphertext *bqnsamiconcoal*?
- How did the first *e* in the saying *“au nom de l’eternel”* get replaced with a *i* in the ciphertext *bqnsamiconcoal*?
- How did the second *e* in the saying *“au nom de l’eternel”* get replaced with a *o* in the ciphertext *bqnsamiconcoal*?
- All four *e*’s in the plaintext were replaced different ciphertexts, why and how did this happen?

Checkpoint 3.1.7. To further check your understanding try enciphering the message *“vive la france”* using the key *aider* and the black (outer) rows in Figure 3.1.6, since there are more letters in the message than the key you will need to restart at the beginning of the key when you get to the end.

Hint. Try writing the key above the message so that you can keep track of how they correspond, like this:

```
Key:        a i d e r a i d e r a i
Plaintext:  v i v e l a f r a n c e
Ciphertext: Q O
```

Also, recall that you look up the letters in the message along the top and the key along the side.

Checkpoint 3.1.8. To further check your understanding try enciphering the message *“escargot”* using the key *ail*; in Figure 3.1.6 use the red (inner) labels for the message and the black (outer) column for the key, since there are more letters in the message than the key you will need to restart at the beginning of the key when you get to the end.

Hint. Try writing the key above the message so that you can keep track of how they correspond, like this:

```
Key:        a i l a i l a i
Plaintext:  e s c a r g o t
Ciphertext:
```

Also, recall that you look up the letters in the message along the top and the key along the side.

Finally, enciphering (and deciphering) goes faster if you do all the characters enciphered with a particular key letter at one time; so first do everything enciphered with *a*, then *i*, and then *l*.

Checkpoint 3.1.9. Try deciphering the message NODDTEMDQNHO using the key *vilo* and the black (outer) labels.

Hint.

- Recall that the plaintext letters run along the top and the key letters are along the left side.
- You need to look up the ciphertext letters in the interior of the table.
- Line up the ciphertext and the key in the same way you lined up the plaintext and key previously.

```
Key:        v i l o v i l o v i l o
Ciphertext: N O D D T E M D Q N H O
Plaintext:  t o u
```

Checkpoint 3.1.10. Try deciphering the message `DOLSASXS` using the key *manger* and the red (inner) labels.

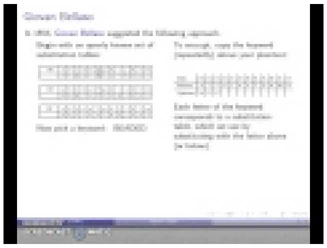
Hint.

- Recall that the plaintext letters run along the top and the key letters are along the left side.

- You need to look up the ciphertext letters in the interior of the table.

- Line up the ciphertext and the key in the same way you lined up the plaintext and key previously.

Today when we use the Vigenère Cipher we use a table Figure C.0.8 which is simpler, and makes the resulting cipher a little easier to crack. The video below explains using the more modern table and you will play with the modern version in the exercises.



YouTube: `www.youtube.com/watch?v=5ISnCm4_V-Y`

Figure 3.1.11: Modern look at the Vigenere Cipher

3.2 Variation on a Theme

Objectives

- Agripa's Biography

- Agripa/Vigeners's Pigpen Cipher

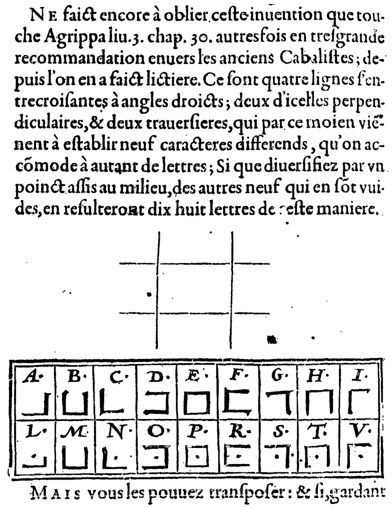
- Modern Pigpen Cipher

3.2.1 Biography of Heinrich Cornelius Agripa

Insert Bio Here When Available ...

3.2.2 Pigpen Cipher

In his treatise on cryptology [13, p. 275b (534)] Vigenère describes a cipher that he found in Agripa's *De Occulta Philosophia* [1, Vol. 3, p. CCLXXV (279)] in chapter 30 which discusses traditional ciphers used by cabals.

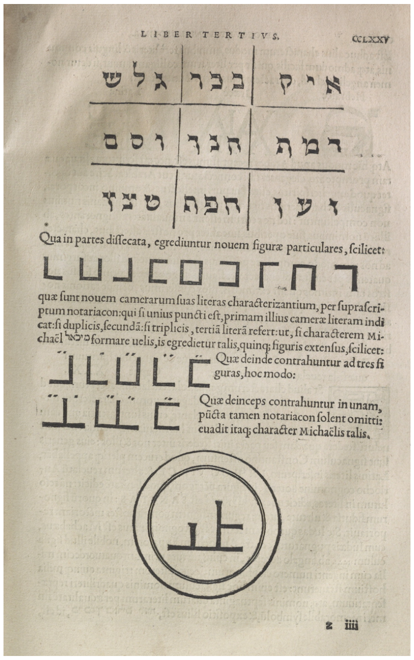


Ne faict encore à oblier ceſte inuention que touche Agrippa liu.3. chap. 30. autresfois en treſgrande recommandation enuers les anciens Cabaliſtes; depuis l'on en a faict lictiere. Ce ſont quatre lignes s'entrecroiſantes à angles droicts; deux d'icelles perpendiculaires, & deux trauerſieres, qui par ce moien viēnent à eſtablir neuf caracteres differends, qu'on accōmode à autant de lettres; Si que diuerſifiez par vn poinct aſſis au milieu, des autres neuf qui en ſōt vuides, en reſulteront dix huit lettres de ceſte maniere.

A·	B·	C·	D·	E·	F·	G·	H·	I·
L·	M·	N·	O·	P·	R·	S·	T·	V·

Mais vous les pouuez tranſpoſer: & ſi, gardant

Figure 3.2.1: Vigenere's Variant on the Pigpen Cipher



LIBER TERTIVS. CCLXXV

ג ל ש	ב כ ר	א י ק
ו ס ם	ה נ ך	ד מ ת
ט צ ץ	ח פ ף	ז ע ן

Qua in partes diſſecata, egrediuntur nouem figuræ particulares, ſcilicet:

quæ ſunt nouem camerarum ſuas literas characterizantium, per ſupraſcriptum notariacon: qui ſi unius puncti eſt, primam illius cameræ literam indicat: ſi duplicis, ſecundā: ſi triplicis, tertiā literā refert: ut, ſi characterem Michaël מיכאל formare uelis, is egredietur talis, quinq; figuris extenſus, ſcilicet:

Quæ deinde contrahuntur ad tres figuras, hoc modo:

Quæ deinceps contrahuntur in unam, pūcta tamen notariacon ſolent omitti: euadit itaq; character Michaëlis talis.

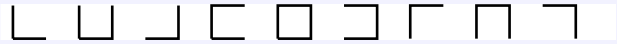
z iiii

Figure 3.2.2: Agripa's Variant on the Pigpen Cipher

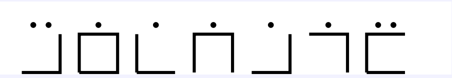
One may frame nine chambers by the intersection of four parallel lines intersectiong themselves at right angles as expressed in the figure:

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

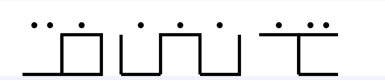
Which being dissected into parts generates nine the particular figures:



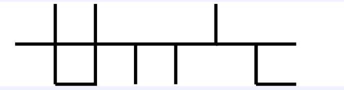
from the nine chambers. Characterize each letter in a chamber by the notation of one point to show the first letter in the chamber; two for the second letter; three for the third letter; so that the characters of Michael may be written in seven characters:



Which may be written one after another drawn as three figures:



Which written one after another drawn as one figure, omitting the usual marks, gives the characters of Michael as:



[1, Vol. 3, CCLXXV (p.279)] [2, Vol.3, Chapter XXX] [a]

[a] The translation is based, for the most part, on [2] with reference to the original text. The example was changed some in order to make the work more accessible.'

Comprehension Check:

- If you hadn't seen the first step in how the name *"Michael"* was enciphered above, then how might you misinterpret the second enciphering?
- Why in the final step do you think some of the symbols must be raised or lowered?
- In the final step above Agripa eliminates all the marks, why would this cause confusion? That is, if you follow his directions which letters will be confused with one another?
- In the translated grid the letters are placed into the grid right to left, this is also the case in Agripa's work Figure 3.2.2, try to find out why.

Checkpoint 3.2.3. Use Agripa's cipher as translated above to encipher the quote: "Pigs are smarter than dogs, and both are smarter than Congress" - Elayne Boosler

Checkpoint 3.2.4. Decipher the words enciphered here using Agripa's cipher as presented above.

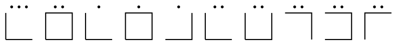


Figure 3.2.5: Cipher Text:

The pigpen cipher as we use it today has evolved and standardized to use the key in Figure 3.2.6 which includes all twenty six letters of modern English. It has the advantages that it is quick to learn and it is easy to recreate the cipher key. Also, for each shape you never add more than a single dot so that there is less opportunity for confusion.

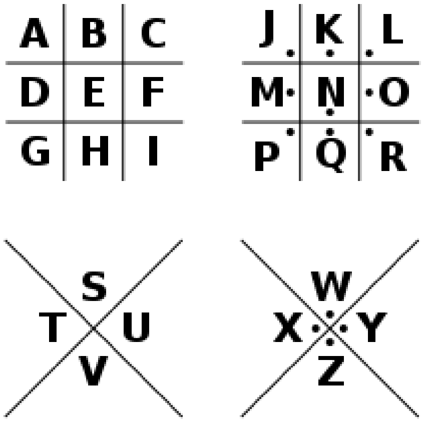


Figure 3.2.6: Modern key for the Pigpen Cipher

Checkpoint 3.2.7. Use the key in Figure 3.2.6 to decipher the message below.

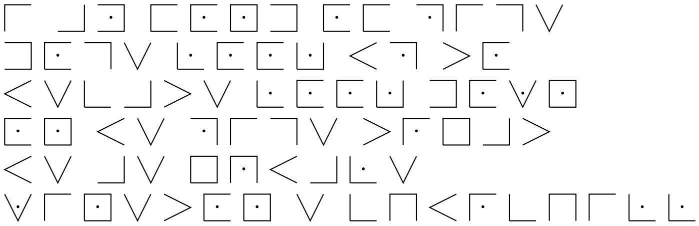


Figure 3.2.8: Cipher Text

Checkpoint 3.2.9. Try to encipher the first few words of this next message using the modern variation on the pigpen cipher (Figure 3.2.6).

“Never wrestle with pigs. You both get dirty and the pig likes it.” George Bernard Shaw

Reflection:

- Do you think that the pigpen cipher offers as much security as the Vigenère cipher? Why or why not?
- In how many ways can you encipher each character using the pigpen cipher? What type of cipher does that make this?
- If you were confronted with a similar cipher which used symbols in place of letters how could you first rewrite your message so that you could then apply the cryptanalysis techniques we discussed previously?

3.3 An Automatic Hit

Objectives

- Vigenère’s Biography
- Vigenère’s Autokey Cipher

3.3.1 Biography of Blaise de Vigenère

Insert biography when available

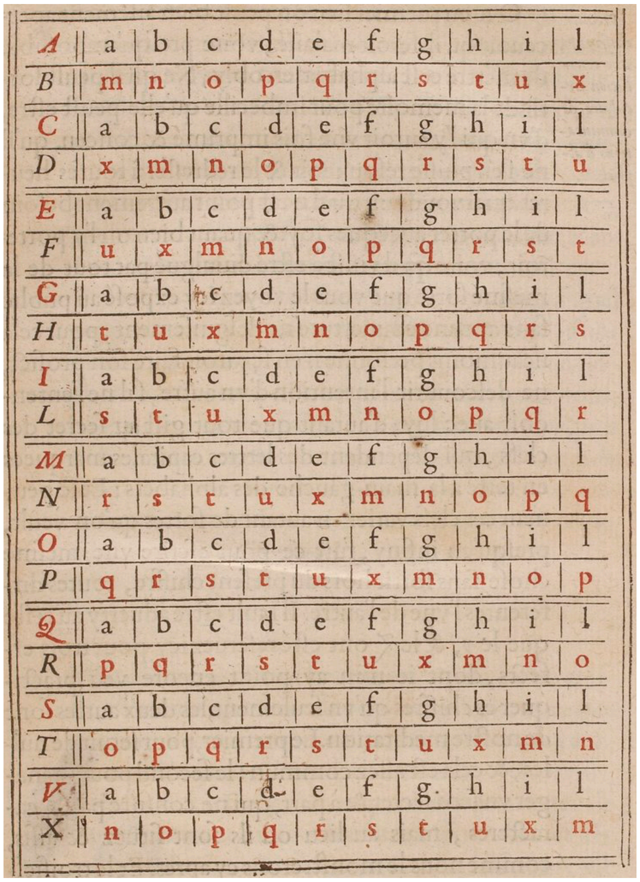
3.3.2 The Autokey Cipher

Description of Alberti's autokey from Kahn ...

Below is a description of Vigenère's more secure variation on an autokey cipher. [1] Read carefully through the passage above, you will need to use Figure D.3.2 in order to follow along.

> "Each letter may be enciphered by the preceding letter, thus: with the . . . text. . . . *"Au nom de l'eternel"* and the key D, we say, a [the first letter of the text] from D [the key letter] gives x; u [the second letter of the text] from A [the first letter of the text, which now becomes the key] gives i; n from U, a; o from N, h; m from O, g; d from M, u; e from D, p; l from E, t; e from L; m, t from E, l; e from T, s; r from E, h; n from R, i; e from N, x; l from E, t. By which method, taking D for the key, we would arrive at *DXIAHGUPTMLSHIXT*. The other method, which is more secret, is to encipher each letter of the clear text not by the letter [of that text] which precedes it, but by the letter by which the preceding letter is enciphered. Thus, a from D, which is the key, gives x; u from X [the first letter of the cipher text] h; n from H, e; o from E, e; m from E, c; d from C, o; e from O, u; l from U, m; ... etc." [13, pp. 49-49b] [8, p. 128]

[1] This translation is mostly from Charles Mendelsohn in [8, p. 128], I have supplemented with material from the original description by Vigenère is in [13, pp.49-49b].



A	a	b	c	d	e	f	g	h	i	l
B	m	n	o	p	q	r	s	t	u	x
C	a	b	c	d	e	f	g	h	i	l
D	x	m	n	o	p	q	r	s	t	u
E	a	b	c	d	e	f	g	h	i	l
F	u	x	m	n	o	p	q	r	s	t
G	a	b	c	d	e	f	g	h	i	l
H	t	u	x	m	n	o	p	q	r	s
I	a	b	c	d	e	f	g	h	i	l
L	s	t	u	x	m	n	o	p	q	r
M	a	b	c	d	e	f	g	h	i	l
N	r	s	t	u	x	m	n	o	p	q
O	a	b	c	d	e	f	g	h	i	l
P	q	r	s	t	u	x	m	n	o	p
Q	a	b	c	d	e	f	g	h	i	l
R	p	q	r	s	t	u	x	m	n	o
S	a	b	c	d	e	f	g	h	i	l
T	o	p	q	r	s	t	u	x	m	n
V	a	b	c	d	e	f	g	h	i	l
X	n	o	p	q	r	s	t	u	x	m

Figure 3.3.1: Table used for Vigenère's Autokey Cipher

Comprehension Check: It will be easier to follow what is happening in this passage if we write out the message as follows:

Text for the first cipher method:

```
Plaintext:    a u n o m d e l e t e r n e l
Key Letter:   D A U N O M D E L E T E R N E
Ciphertext: D X I A H G U P T M L S H I X T
```

- Vigenère uses a D as the key to encipher the first letter, in the first method described where does he get the subsequent letters for the key?
- The letter D enciphers a as X, but the a is not in row D of Figure D.3.2 so how do we know that the a should be X ?
- At one point Vigenère uses a key letter of U but there is no row U on his table, looking at how the U enciphers the n, what row is he using in place of U?
- The a was enciphered as X, but the first letter of the ciphertext is D, also the ciphertext is one character longer than the plaintext, what is the explanation for both of these? Where did the leading D come from?

Text for the second cipher method:

```
Plaintext:    a u n o m d e l e t e r n e l
Key Letter:   D X H E E C O U ...
Ciphertext: D X H E E C O U M ...
```

- In the second method described he again starts with a key of D but then where does he get the the next key letter X? What about the key letter H? Try to complete the cipher using the second method.

- Vigenère believes his second method is more secure, looking at the *key* for the second method why do you think he believes this?

- If you compare the *key* for his second method to the cipher text he generates why is Vigenère wrong about his second method being more secure?

Checkpoint 3.3.2. Check your understanding by enciphering the message *"automatic confusion"* using Vigenère's first and second autokey methods, use Q for the initial key letter.

Checkpoint 3.3.3. The message `CHFABH CXNXM GFMRR MDDMR` was enciphered using Vigenère's first autokey method, check your understanding of the method by deciphering it.

Checkpoint 3.3.4. The message `QMPTM GPOHG SOOHL QXUGC` was enciphered using Vigenère's second autokey method, check your understanding of the method by deciphering it.

Reflection: If you go back and look carefully at the preceding couple of exercises you will notice that deciphering a message enciphered using the first autokey method is equivalent to second method and vice versa. In what way does this weaken the cipher?

3.4 Stiring Things Up

Objectives

- Falconer's Biography

- Falconer's Discussion of Permutations

- Falconer's Columnar Transposition

3.4.1 John Falconer's Biography

Insert biography here when ready

3.4.2 Key Permutations

> ***Sect. 2: Of Secret Writing by Altering the Places of the Letters, Where the Powers Remain the Same***
> *Part 1: Of the Combinations of Three or More Letters*
> The first remarkable improvement I find of this kind of cryptography by altering the places of letters, is by the regular combinations of three, four, or more letters: I had it of a gentleman, who (I am fully satisfied) would put it to no bad use; but since it may fall of bad hands, I have his leave to provide against its harm. And that we may proceed regularly

therein it is necessary, first to inquire, "how many several ways any given number of letters may be combined?" (i.e.) How many differnt positions they can regularly admit of. And for that end I have hereunto subjoined the following table:

Letters		Several Ways
1		1
2		2
3		6
4		24
5		120
6		720
7		5040
8		40320
9		362880
10		3628800
11		39916800
12		479001600
&c.		

...

Demonstration.

1. It is manifest, that one letter or thing has but one position, and two letters have twice the position of 1, viz. once before and once after it. e.g. *AB*, *BA*.

- John Falconer [4, pp. 37-40 (65-68)]

3.4.3 Transposition

A New Method How to Write Secretly by the Art of Combinations

1. To write by the method proposed, a certain number of letters are combined to lock and unlock the epistle. The differences of writing down the positions [of the letters] ... may be varied to a vast number; ...

2. The order of rows is agreed upon in parting.

3. The number of letters combined, which is the key, may be expressed in the epistle by some mathematical figure, as △ for three letters, □ for 4, etc. or by some private mark.

4. They [the individuals communicating] frame a rectangular table of as many columns as there are letters combined.

> 5. The letters so combined are placed in their natural order along the top of the table.
>
> 6. Having determined of how many lines the table shall consist, the order of combinations agreed upon is set down in a row in the first column towards the left hand; as you may see in the following table.
>
> 7. The table being thus prepared for writing, they observe the order of the combinations, and write according to its direction.
>
> 8. When they have placed one letter of every column of all the lines, they begin a new, and so go on until the writing be finished.
>
> 9. And lastly, they take the letters out of the table according to their partitions, as so many barbarous words, upon paper apart and send it to the confidant. - John Falconer [4, pp. 68-72 (40-43)]

Let's see if you can follow Falconer's directions. Below I set up Table 3.4.1 according to his description in steps (5) and (6) and used it to encipher the pangram *"the quick brown fox jumps over the lazy sleeping dog."*

		A	B	C
1	CBA			
2	CAB			
3	ACB			
4	BCA			
5	BAC			

Table 3.4.1: Falconer's Transposition Table Initial Setup

		A	B	C
1	CBA	E	H	T
2	CAB	U	I	Q
3	ACB	C	B	K
4	BCA	W	R	O
5	BAC	F	N	O

Table 3.4.2: Falconer's Transposition Table First Pass

		A	B	C
1	CBA	EUS	HJY	TXZ
2	CAB	UPE	ISE	QML
3	ACB	COP	BEN	KVI
4	BCA	WHO	RRG	OTD
5	BAC	FL	NEG	OA

Table 3.4.3: Falconer's Transposition Table Filled Table

Final Message:

> “△ EUS HJY TXZ UPE ISE QML COP BEN KVI WHO RRG OTD FL NEG OA”

Refection Questions:

- Looking at Table 3.4.2 why is *the* from the beginning of the sentence written backwards? (Be sure to look carefully at the letters to the left of the row when answering.)
- In the second row of the same table why is *qui* from *quick* written in the order it is written? (Again, be sure to look carefully at the letters to the left of the row when answering.)
- In the last row we put down *n* from *brown* and the *fo* from *fox*, why are they in the order they are in and looking at the next table (Table 3.4.3) where do we put the *x* from *fox* and why?
- How do we finish writing the rest of the message into the boxes in the table?
- Refection Questions: Looking at the final message why is there a little triangle at the start of the message and why were the blocks of letters written in the order they were written?
- In what ways is this different from other ciphers we have looked at? (Hint: in this cipher what does cipher text *E* represent, or cipher text *F*?)

Checkpoint 3.4.4. Decipher the following message which used the same key as the one represented in Table 3.4.1:

> △ WU OO NR MD A IN L T L N U R D E A

Checkpoint 3.4.5. Encipher the following message using the same key as the one represented in Table 3.4.1:

> “As long as the world is turning and spinning, we’re gonna be dizzy and we’re gonna make mistakes.” - Mel Brooks

Checkpoint 3.4.6. Encipher the following message using the three letters as before, but with the row keys `ACB - CAB - BAC - ABC`, Table C.0.11 is a blank table you can copy if you need to:

> “A person who never made a mistake never tried anything new.” - Albert Einstein

3.5 (*)Presidential Secrets

Objectives

- Jefferson’s Biography
- Jefferson’s Wheel Cipher

3.5.1 Thomas Jefferson’s Biography

A brief biography of Thomas Jefferson.

3.5.2 Thomas Jefferson’s Wheel Cipher

A brief description of Thomas Jefferson’s wheel cipher.

Checkpoint 3.5.1. Insert Jefferson Exercise Here!!!

How Mixed Up Do You Feel?

1. Encipher this message using a *Pigpen Cipher*. Figure C.0.10

> “You can put wings on a pig, but you don’t make it an eagle.” - William J. Clinton

2. Decipher this message written using a *Pigpen Cipher*. Figure C.0.10

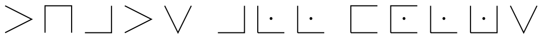


Figure 3.5.2: Cipher Text

3. Encipher the following message with Alberti’s polyalphabetic cipher, Figure C.0.5, use and index letter of *s* and key letters *C*, *G*, and *X* (switch key letter after every other word).

> “Men can do all things if they will.” - Leon Battista Alberti

4. Encipher the following message using the modern version of the Vigenere Square Figure C.0.8 and a keyword of *FRANCE*.

> “Hope is a good breakfast, but it is a bad supper.” - Francis Bacon

5. Use the modern Vigenere Square Figure C.0.8 to decipher the following message which was enciphered using a key word of *STATE*.

```
LAEKI SKEGS KXCKI LLTHW MVCXW KBTBW LAEKI KNLMS
XIRXT SKAMM GGHTV VPOKO SGDEI SKNBR YYRHQ XTIEY
JXCHP AGPHA WEL
```

6. Encipher this quote with an *Autokey Cipher* with a key letter of *F*, use Vigenere’s original table for this one. Figure C.0.9

> “Obstacles are those frightful things you see when you take your eyes off your goal.” - Henry Ford

7. Encipher this quote with an *Autokey Cipher* with a key letter of *J*, use the modern Vigenere Square for this one. Figure C.0.8

> “Success is not final, failure is not fatal: it is the courage to continue that counts.” - Winston Churchill

8. Use the modern Vigenere Square Figure C.0.8 to decipher the following message which was enciphered using an Autokey Cipher and key letter of *X*.

```
ARBHN LAALA LVVEN LRAPV TZALC ACNGW RAMTM ZXFSR VVT
```

9. Encipher the following message using Falconers Columnar Transposition with a key of *ACB,BCA,CAB,ABC*. Table C.0.11

> “Were I not a king, I would be a university man.” - King James I

10. Encipher the following message using Falconers Columnar Transposition with a key of *CBA,ABC,BCA,CAB,ACB,BAC*. Table C.0.11

> "I have loved justice and hated iniquity: therefore I die in exile." - Pope Gregory VII

11. Decipher the following message using Falconers Columnar Transposition with a key of *ABC,BCA,BCA,ACB*. Table C.0.11

```
ISUIIP TTRNLE IADOLA OTTSM SREUIR NSSRAE NOYVA TOI
ES IHNLH TLBEK EOTWS HDUSE
```

Substitution Cipher Cell:

```
import textwrap
@interact
def _(p =
    input_box('abcdefghijklmnopqrstuvwxyz',label='Plain',
    type=str,width=50,height=1),
            c =
                input_box('ZYXWVUTSRQPONMLKJIHGFEDCBA',label='Cipher',
                type=str,width=50,height=1),
            shift=[0..25],
            mode=selector(['encipher','decipher'],
                buttons=True),
            spaces = selector(['yes','no'], buttons=True),
            m=input_box('sage', label="Message", height=5,
                width=50, type=str)):
    P = str(p.encode('ascii','replace')).upper()
    C = str(c.encode('ascii','replace')).upper()
    C = C[shift:]+C[:shift]
    Message = str(m.encode('ascii','replace')).upper()
    print "\nPlain␣alphabet:␣\t", str(P).lower()
    print "Cipher␣alphabet:\t", str(C)
    if len(C)!=len(P):
        print "Key␣lengths␣do␣not␣match."
    else:
        if mode == 'encipher':
            inText = P
            outText = C
        else:
            inText = C.lower()
            outText = P.lower()
            Message = Message.lower()
        output = ""
        for char in Message:
            try:
                position = inText.index(char)
                output += outText[position]
            except:
                if spaces=='yes': output += char
                else: pass
        print "\nHere␣is␣your␣output:\n"
        if spaces == 'yes':
            print textwrap.fill(output, 50)
        else:
            for i in xrange(0,len(output),5):
                print output[i:i+5],
                if (i+5)%50 == 0: print "\n"
```

Vigenere Cipher Cell:

```
import re
import textwrap
@interact
def _(m=input_box('sage', label="Enter␣your␣message",
    height=3, width=50, type=str),
        key=input_box('sage', label="Enter␣your␣key",
            height=1, width=20, type=str),
        mode = selector(['encipher','decipher'],
            buttons=True),
        spaces = selector(['yes','no'], buttons=True)):
    plain_alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    clean_message = str(m.encode('ascii','replace')).upper()
    if spaces == 'no':
        clean_message = re.sub('[^A-Z]','',clean_message)
    cipher_key =
        re.sub('[^A-Z]','',str(key.encode('ascii','replace')).upper())
    key_list = [plain_alpha.index(ch) for ch in cipher_key]
    if mode == 'decipher':
        key_list = [-1*k for k in key_list]
    cipher_text = ""
    key_counter = 0
    for ch in clean_message:
        try:
            tmp_pos = plain_alpha.index(ch)
            cipher_pos = (tmp_pos+key_list[key_counter])%26
            cipher_text += plain_alpha[cipher_pos]
            key_counter = (key_counter+1)%len(key_list)
        except:
            if spaces == 'yes':
                cipher_text += ch
    print "\nHere␣is␣your␣output:\n"
    if spaces == 'yes':
        print textwrap.fill(cipher_text, 42)
    else:
        for i in xrange(0,len(cipher_text),6):
            print cipher_text[i:i+6],
            if (i+6)%42 == 0: print "\n"
```

Falconer's Tabular Cipher:

```
import textwrap
import re
@interact
def falconer(message=input_box("The␣quick␣brown␣fox␣jumps␣
    over␣the␣lazy␣sleeping␣dog.",
                                  label="Message:",
                                      type=str, width=50,
                                      height=3),
             keys=input_grid(1,6,default=["CBA", "CAB",
                 "ACB", "BCA","BAC",""],
                    label="Keys:", to_value=list, type=str),
            chars=[3..5]):
    text =
        re.sub('[^A-Z]','',str(message.encode('ascii','replace')).upper())
    columns = "ABCDE"
    key = keys[0]
    while "" in key: key.remove("")
    message_table = [["" for x in range(chars)] for y in
        range(len(key))]
    for i in xrange(0,len(text),chars):
        row = (i/chars)%len(key)
        for j in range(chars):
            try:
                col = columns.index(key[row][j])
            except:
                col = chars-1 #pass
            try:
                message_table[row][col] += str(text[i+j])
            except:
                pass
    out_message = ""
    print "Chracters␣in␣text:␣",len(text)
    print "Cipher␣Table:"
    for k in range(len(key)):
        print
            "\t",str(key[k][0:chars]),":\t","\t".join(message_table[k])
        for i in range(chars):
            out_message += str(message_table[k][i])+"␣"
    print "Completed␣Message:"
    #for i in xrange(0,len(out_message),50):
    #    print
        "\t",out_message[i:min(i+50,len(out_message))].strip()
    print textwrap.fill(out_message, 50)
```

Chapter 4

Triumphs of Logic and Statistics

In this chapter discuss the emerging roll of mathematics in analysis:

- Falconer's Attacks
- Babbage's attack
- Friedman and Statistics
- Sukhotin's Algorithm

4.1 A Simple Solution

Objectives

- Factoring Integers
- Falconer's Solution to the Columnar Transposition

4.1.1 The Power of Prime Factors

The strength of Falconer's keyed columnar transposition cipher is the almost astronomical number of combinations of keys that can be used. As we have seen previously with just three characters we have six row keys which, depending on how many of them we use and how we arrange them, can give us one thousand nine hundred fifty six subtly different ciphers. If we increase the number of characters to four or five then that number goes up to

$$1,686,553,615,927,922,354,187,744$$

and

$$\begin{gathered}18,183,954,211,052,603,322,452,474,095,140,831,\ldots\\738,614,446,765,249,926,715,439,301,461,074,500,\ldots\\732,116,312,180,273,095,148,765,061,059,468,326,\ldots\\378,636,312,510,693,233,993,926,141,650,787,502,\ldots\end{gathered}$$

$$931, 879, 726, 557, 669, 253, 713, 958, 681, 131, 266, 045, \ldots$$
$$931, 864, 486, 980, 283, 708, 000$$

respectively. This would seem to offer security in the extreme, but as we will see, Falconer himself gives us two relatively simple ways to crack this cipher. We begin with the more mathematical approach.

> ... take the number of partitions of the seeming words in the epistle, and find out their several divisors, which may be performed by the following rules.
>
> *How to find out the equal divisors of any number.*
>
> 1. Divide the number given by some **prime number** (i.e.) such a number that cannot be divided, but by it self, or unity; and the *quotient by some or other prime number*, and the *last quotient again by a prime number*; and so go on until the *last quotient of all be one*; and thus you shall find a certain number of *prime divisors*.
>
> 2. Make a rectangular table that shall consist of as many columns as you have prime divisors, which you must place one after another at the tops of the columns; ...
>
> By multiplying the first prime divisor towards the left hand of the table by the second, and writing the product under the second.
>
> Next, by the third prime divisor, multiplying all the figures in the table towards the left hand, setting the several products in the third column: and so forth, throughout all the prime divisors; ...
>
> *Example to find out all the divisors of 450*

450	225	75	25	5	1
2	3	3	5	5	

> The first line contains the first divided, and the respective quotients; the lowest line is the several prime divisors.
>
> Now 450, the number given, being divided by 2, a prime divisor, the quotient is 225, which being divided by 3, you have 75 for a new quotient; and again divided by 3, you have 25 for another quotient. This last divided by 5, gives 5, which being a prime number, you have 1, or unity in the last quotient of all: so that your prime divisors are, 2, 3, 3, 5, 5, all which set down in the tops of the columns [of a new table], and multiplying them according to the [second] rule given, the operation will stand thus.

2	3	3	5	5
	6	9	10	25
		18	15	50
			30	75
			45	150
			90	225
				450

> All the divisors of 450, are 2, 3, 5, 6, 9, 10, 15, 18, 25, 30, 45, 50, 75, 90, 150, 225; and one of them (supposing the epistle to have consisted of 450 seeming words) should have been the number of letters combined

> for the key: ...
> - John Falconer [4, pp. 44-47 (72-75)]

- What are *prime numbers* and what are the *prime divisors* of a number?
- How did Falconer decide to start dividing by 2?
- Have you ever learned a rule that may have told you that 225 and 75 were divisible by 3?
- Looking at the second table in the above quote, where do we get the numbers in the top row?
- In the second column of the second table there is a 6 under the 3, how did Falconer come up with that?
- In the third column how did he get the 9 and the 18?
- What about the other columns, where did those numbers come from?
- In the last paragraph Falconer tells us that if 450 were the number of blocks of text in the enciphered message then one of the factors of 450 is the number of key letters in the message. Looking back at how Falconer's transposition cipher works why is this the case?

Checkpoint 4.1.1. Test your understanding of what Falconer has described by factoring 980 into prime factors and then writing out all the different divisors

Checkpoint 4.1.2. Look at the text enciphered below using Falconer's transposition cipher. Using the techniques described above, how many key letters may have been used to encipher this message?

```
NILOEOTHIETTSGTN HCITRTSRNIAEOIOC TATEUOOOETURHVFS
LNEURRHYMSIHHIHQ EELHEFETTMRDUNRO NEAMTDPLHYETLOAS
TTEPEEYTPEOESTTA THMOIIITESHTEUPUE ETDSEBASETFEONOU
OEKEBDAFASNHDSRT NSECLSEREWHHEICM EMTWKRIEBRNTAYTK
TUEEFUWEREIAPTHA IYWAEUHBIFTWDRUD WHSLSNTOPLFNTEMN
GRVEDCGTEPTDATIM RMSIIYLDEEECRBOE AEEBEINOTPNLEOAR
SEOTLWLAIHTOTEOW RRMHLHINTAEULNFO DHLTAAAROREANLOH
ETAUHHOEFEMEIETT HOTYTPNTRTSTGFBI AURSUPTHOHRHTWEN
GEIATHLHNIIDVSRS FSRSNHFTNOUTIDHR EIUASTSREHGFLETI
DDSEISTOTTIIELAE DEAREEAEENFOELEH NBETSIHENAEDHIDE
```

Hint. Start by counting the number of blocks of letters.

4.1.2 But Really There's an Easier Way (sort of)

After walking us through a careful exposition of how to find all the factors of a number and encouraging us to use this in order to find out the number of letters used for our key, Falconer lets us know that there is effectively an easier way to attack his own cipher.

> "or rather for dispatch, take out the seeming words, and write them down in [columns] beginning at the first, and then proceed to the second,

> third, fourth, fifth, etc., until you have gone through them" - John Falconer [4, p.47 (75)]

Let's follow Falconer's advice using the message we used when we introduced his cipher in Section 3.4:

> "△ EUS HJY TXZ UPE ISE QML COP BEN KVI WHO RRG
> OTD FL NEG OA"

Which transforms to the following when we rewrite it following Falconer's directions:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	E	H	T	U	I	Q	C	B	K	W	R	O	F	N	O
2	U	J	X	P	S	M	O	E	V	H	R	T	L	E	A
3	S	Y	Z	E	E	L	P	N	I	O	G	D	·	G	·

Table 4.1.3: Transpose Table for a Falconer Cipher

> "1. Search in the several lines for some of the particles [(words or n-grams)] of that language you shall suppose the epistle to have been writ in. If in English, make suppositions, e.g. for such little words as *the, that, for, of, to, and, etc.* and the like, without some of which no man can well express business of any moment."
> "2. Having supposed in any of the lines; for some one of those mentioned, or the like particles, you may prove the truth of your supposition, by taking out the opposite letters of all the lines: And if they do not make words, or syllables, or produce such letters as can probably follow one another in that order, your first supposition is false, and you must suppose anew."
> "3. Having by fresh suppositions found some useful word: And the letters of the other lines (in the same order) agreeing, the words or syllables arising from them, will direct you to some new [column] that goes before or after in the true order: And thus you may proceed till you have found out the whole writing, which by this time will be no great difficulty." - John Falconer [4, pp.48-49 (76-77)]

Taking Falconer's advice from step one look at each line of Table 4.1.3 we created above as you consider the following.

- Are there three letters in line one which we would expect to go together assuming this is written in English?
- There is a Q in row one column six, what needs to come after that? What column is it in?
- Looking now at line two what three letters do we again see that should go together?
- In line three are there three letters you could put together to get a common English ending? Are there other letters you could arrange with them in order to get an entire word?

- Finally, columns thirteen and fourteen only have two characters in them each, where do you think they belong if we were to rearrange the lines?

Use the observations you just made in order to rearrange the columns in the table, you should make a copy of this table to help you.

Col's															
1	____	____	____	____	____	____	____	____	____	____	____	____	____	____	____
2	____	____	____	____	____	____	____	____	____	____	____	____	____	____	____
3	____	____	____	____	____	____	____	____	____	____	____	____	____	____	____

Table 4.1.4: Blank Transpose Table for a Falconer Cipher

Test your understanding by working on the following exercises of increasing size, be patient they are finicky and time consuming. As a hint, all of the quotes contain the word *WINTER*.

Checkpoint 4.1.5. Try to use this new method to decipher this quote.

```
IRPL WPDL FSNE INIH NIEE
RBY  OFY  TNRY CEB  EGC
MAS  SBH  ERS  CEE  AHS
```

Hint.

- Transpose each block, you will end up with a table of four lines and fifteen columns.
- Can you find common or even not so common words or combinations in any lines?
- When you rearrange the columns do you see more words?
- Stay organized, if you don't keep things lined up this will never work

Checkpoint 4.1.6. Try to use this new method to decipher this quote.

```
UIA ARM LDU GVN EWC RIE TSA HEF INV HRT
STI TEC NMU EFO SRR UOH AE  TTG TH  HHO
```

Hint.

- Transpose each block, you will end up with a table of three lines and twenty columns.
- Can you find common or even not so common words or combinations in any lines?
- When you rearrange the columns do you see more words?
- Stay organized, if you don't keep things lined up this will never work

Checkpoint 4.1.7. Try to use this new method to decipher this quote.

```
WYTWIA TAODLH IDHLES ASAHGD NETIAH ENHTNA OHDNTC
SWNEHE OTEIDR HSNUNS FHWSWL TEISIE MHORIK OUDMTD
SNBMEI ESLERC CECTHS AIWINE HSOHE  RNSNTN
```

Hint.

- Transpose each block, you will end up with a table of six lines and twenty columns.
- Can you find common or even not so common words or combinations in any lines?
- When you rearrange the columns do you see more words?
- Stay organized, if you don't keep things lined up this will never work

A few closing observations:

- Look carefully at how the blocks of characters (columns in the tables) are grouped after they are deciphered, i.e. did you ever get column one next to column thirteen or fourteen?
- Why do you think this happened? How might it be related to the number of key letters?
- Given your answers to the previous questions, how can we make use of Falconer's strategies for finding divisors and the number of key letters?

4.2 A Seventeenth Century Idea

Objectives

- Falconer's Attack

4.2.1 Falconer's Attack on Polyalphabetics

> "There is an invention of secrecy much insisted on (though none of the swiftest) by the author of the *Secret and Swift Messenger*, and others, beyond any yet mentioned, for intricacy, wherein each particular line, word, or letter, is written by a new alphabet: but the cited author himself acknowledges it too tedious for a current correspondence;" - John Falconer [4, p.17 (45)]

So begins John Falconer's discussion of using the polyalphabetic cipher. As with other cryptographers of his time he agrees that the cipher, while secure, is to slow an cumbersome and so not worth the trouble of using. However, for the sake of giving a complete account of the current state of the science of cryptography in his time, Falconer goes on to discuss in detail how to encipher and decipher messages with this type of cipher. What is interesting and more significant is that he makes a solid attempt at cracking this cipher, possibly the first real attack since its invention[6, p.155].

For his first example of how to attack a polyalphabetic cipher Falconer assumes each line is enciphered by a separate cipher alphabet. [4, pp.20-23 (48-51)]

I. Example in the Lines

```
Y pb vdgrts id ztte ixt HdafytgI am forced to keep the Soldiers
idcb wofr rihm obr rihm rxfh   upon hard duty and hard diet:
dfaawi fd ze espi gtww cpfzwe eSupply us, or they will revolt to
cqn Nwuxg bynnmrtg. qibc.      the enemy speedily. Hast.
```

“1. When there is only one alphabet used for a line, the writing might be discovered an in plain cipher, if you make a new operation for each line. But there may be other ways to decypher any such writing: for,”
“2. If you find out but one letter in a line, (and that may certainly be done by a few suppositions) it will of itself give an alphabet for the whole line, as you may perceive by the *counter-table*, which follows: ... you need only to search for *i* in the upper line of it, and try in what line *Y* is opposite to it; and those two lines give you an alphabet.”
“Having found one alphabet for the first line, you have likewise by this means the first letter of the key. e.g. In the fifteenth line of the table, *Y* standing against *i*, and *P* beginning that line (as you may perceive) *P* must be the first letter of the key;”
“... you may proceed to find the alphabet of the second, third, or any other line, as you did the first;”

1 *A*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*
2 *B*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*
3 *C*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*
4 *D*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*
5 *E*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*
6 *F*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*
7 *G*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*
8 *H*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*
9 *I*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
10 *K*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*
11 *L*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*
12 *M*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*
13 *N*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*
14 *O*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*
15 *P*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*
16 *Q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*
17 *R*	*s*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*
18 *S*	*t*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*
19 *T*	*u*	*w*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*
20 *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*
21 *W*	*x*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*
22 *X*	*y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*
23 *Y*	*z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*
24 *Z*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*w*	*x*	*y*	*z*

Figure 4.2.1: Falconer's Counter-Table

- Looking at Falconer's cipher text, why would you assume that the *Y* represented *i*? What other letter might it represent?

- How did He decide that line 15, which starts with *P*, was the one that was used to encipher the first line of the cipher text?

- Verify that the fifteenth line is the correct one to use for the first line by deciphering the rest of the line.

- Looking at the next three lines of the cipher text try, without looking at the accompanying plain text, to decide which row of the *counter-table* was used to encipher them. What can you look at to try and help you decide?

- Finally, what is the keyword of phrase that was used to encipher this message?

In the next section of his text Falconer tackles the problem of deciphering a message in which we switch alphabet with each new word. It is here that he makes a significant observation that can help us find the length of the keyword.

> "1. Having found an alphabet for the first, second, or indeed any word near the beginning of the epistle, go through all the immediate following words, until you find another that is decyphered by the same alphabet."
> "2. From the last found word count the like number, and you have a new word decypherable by the found-alphabet: and thus you may go on until you have once gone through the whole writing, marking the whole series with some particular mark: And then,"
> "3. Begin the epistle again at some word immediately before or after that which was first found, and count forwards as before, until you come to the end of the epistle. [Repeat this process until each word is marked as part of a series.]" - John Falconer [4, pp.24-25 (52-53)]

Let's try to see if we can make sense of Falconer's advice and decipher the message below.

- Start by trying to find some repeated pieces of text. Falconer seems to imply that these are enciphered using the same alphabet from his counter-table; thinking about how we use the Vigenère cipher normally why could we end up getting theses repetitions?

- How far apart are the repeated strings? What factors do they have in common?

- Follow Falconer's advice and break the message into groups of strings which were enciphered (hopefully) with the same shift.

- Now use what we learned about frequency analysis to figure out the words in each group. Keep in mind that each shift corresponds to a row in Falconer's counter-table and that the row markers should spell out a keyword.

- What does our message say? What was the keyword?

- How did finding the length of the keyword help reduce this problem to one we had already solved?

```
MX EIB WKG LOCD SK GWZRF,
BM ZEW CQN YQTUW YP YNQIX,
WG OSL XLH IPN QH FSCNYW,
NY INF MAX EKH XO HQQNLUKPGUU,
SD AEX GUR XHGUA RI KNTRNO,
LW FKC YMI RCBPU GY MQFUHGYOMXB,
RC YCU DRO XIEXSR BS DBZAM,
MX EIB WKG COKCYX SK QNEXARFF,
BM ZEW CQN URTLPI YP MSTI,
WG OSL XLH ERWCNA QH NOCZKSB,
AI UNQ XNXKQMABFY FHIRUH DB,
YG TKN RSYMNRL ORSBER NL,
ZH ENAN CNN QYSXQ HNWIGY GB AXNXF,
ZH ENAN CNN QYSXQ HNNIGY GUR GMAXK ZDB -
RW UKQTW, DRO TIWNSH INF LG IDU TRSN WKG ZBOCOXD TIWNSH, GUNG LGEX
RI RCB PQLULGUW KEDRYBSDSOC NRXNXYIH BA BML EHWQK ANLNRDNM, HQT QYYN
SW SBE XNBD, MQ CQN UXRGTNCWLXG NOQBOO SK PBZCNEWFBA GFDQ
```

Exactly one hundred years after Vigenère published his work Falconer lead us in the right direction to crack the polyalphabetic substitution cipher. First, he pointed out that whether we encipher each new line, word, or letter by a different alphabet if we can group elements of the message which were enciphered the same way together then we can treat each group as a monoalphabetic substitution cipher. Additionally, if we can identify the length of the key then we can use that to figure out how to group the strings or characters together. However, Falconer was still very much tied down to the use of word spacings, without them his descriptions and attacks would not work. It would be another century and a half before Charles Babbage and Friedrich Kasiski would recognize how to expand these ideas into a general plan of attack.

4.3 Nineteenth Century Revelations

Objectives

- Babbage/Kasiski's Attack

4.3.1 Kasiski's Attack

4.3.1.1 Initial Observations

What follows are excerpts from Friedrich Kasiski's 1863 text on cryptography *Die Geheimschriften und die Dechiffrir-Kunst*, which was translated into English and privately published by R. W. Pettengill in 1954 as *Secret Writings and the Art of Deciphering*[7]. For brevity we will only examine a few of Kasiski's key points and we will use a long English rather than a short German quote in order to make his points. While these adjustments have been made to aid in understanding whenever possible we will remain faithful to the original.

For this section we will use the following quote for our analysis. It is enciphered using a Vigenère cipher with the keyword BERLIN.

Example 4.3.1 Sample Vigenère Enciphering. Plain Text:

"Tell the general, Remarque, that the army has mobilized to attack the Russians on the eastern front, and the French, the English, and the Americans on the western front which is currently all quiet." [1]

Cipher Text:

[1] Kasiski's original quote was "Die Armee wird mobil gemacht" also enciphered with BERLIN.

```
      123456  123456  123456  123456

 Key: BERLIN  BERLIN  BERLIN  BERLIN

  (1) UICWBU  FKVYME  BPIPUN  SULPBU

  (2) BXKSMN  SQPSIF  NSSTTV  AIUEWN

  (3) UXRNSG  IIIFAF  JEEDWA  ULVPIF

  (4) UIIYNE  PRKLVQ  ULVQZR  OGYEPR

  (5) FRXWQF  IEEOBU  FEDPZV  DEEDWA

  (6) ULVHMF  UIIYNE  PRKHPV  DLZDKH

  (7) SVVYBY  ZECWYH  JIK
```

> VI. The Actual Art of Decipherment [Cryptanalysis].
> A. General Remarks.
> 72. The actual art of deciphering consists in deciphering any unknown cipher text without previous knowledge of the meaning of the ciphers and without the key.
> ...
> It is impossible to discover the key from a few words of cipher text ... . Consequently we must have available at least four to six rows of cipher text if we are to be able to recover the key.
> ...
> 75. The influence of the key finds expression in the fact that at intervals of as many ciphers as the key contains letters repetition of single ciphers or of several ciphers can occur, which can only be due to the fact that, when the same letters of the key were united with a repetition of the plain text to form ciphers. If we have a relatively long cipher text which is enciphered with the key "Berlin," and if we divide it off into rows of six ciphers and number the ciphers of each row from one to six, we shall find that among the ciphers numbered 1 of each row the cipher U occurs most frequently, among the ciphers numbered 2 I is most frequent, the most frequent among the ciphers numbered 3 is V, among the ciphers numbered 4 Y, among the ciphers numbered 5 B, and among the ciphers numbered 6 F. This is because the ciphers U, I, V, Y, B and F have [likely] arisen by combination of the letters b, e, r, l, i, n of the key with the letter e of the text; since according to [previous observations] the letter e occurs most frequently in the text, naturally the above ciphers must occur most frequently under the numbers mentioned.
> Next most frequent in this cipher text are the ciphers S in column one, E in column two, K in column three, P in column four, W in column five and V in column six because these ciphers S, E, K, P, W and V result from a combination of the letters b, e, r, l, i, n of the key with the letter t of the text and t is the next most common letter letter after e. [7, pp.29-30 (34-35)]

Reading Questions:

- In section (75) Kasiski writes about letters of the *key* uniting with repetitions of *plain text*, how does this compare to Falconer's observations in Section 4.2?

- We know that the Vigenère cipher can encipher many different plain text letters as the same cipher text letter, so why is it reasonable for Kasiski to believe that all the U's in column 1 should represent e or that all the P's in column 5 might represent t? (Answer in terms of how we know the Vigenère cipher works.)

- Break the plain text message in Example 4.3.1 into blocks of six then use the key BERLIN and the modern Vigenère table Figure C.0.8 to encipher it, you should get the same cipher text as that which is in the figure.

- Is Kasiski right about his guesses, that is do the U and I in the first and second columns represent e, or do the E and K in the second and third columns represent t? How do your answers to this question relate to Kasiski's comments in section (72)?

Further observations by Kasiski:

> 76. ... where repetitions of two or more ciphers are found in the cipher text one can conclude that these are due to the fact that, in enciphering, repetitions in the plain text were repeatedly combined with the same letters or the key.
> If we encipher with the key "Berlin" as above, we shall frequently find in the cipher text in columns 1 and 2 the ciphers UL, in 2 and 3 XY, in 3 and 4 KS, in 4 and 5 EP, and in 5 and 6 BU because these ciphers UL, XY, KS, EP and BU have arisen by combining the letters BE, ER, RL, LI and IN in the key with the [repetitions of] th in the text. [7, pp.30-31 (35-36)]

Reading Questions: (For these it will be helpful for you to look at the work you did in the above questions.)

- Look carefully at the cipher text in Example 4.3.1, how often do you find the bi-grams UL, XY, KS, EP and BU? Do they always line up with the cipher key so that they give th when you decipher them?

- For this observation Kasiski enciphered th using each of the pairs of letters BE, ER, RL, LI and IN from BERLIN, but he didn't consider enciphering it with NB (i.e. wrapping around from column 6 back to 1). Encipher the bi-gram th using the rows NB in the Vigenère table Figure C.0.8, what cipher bi-gram do you get? How often to you find that bi-gram in the cipher text in Example 4.3.1? (Remember that you need to look from columns 6 to 1.)

- Can you find other repeated bi-grams? Can you find longer repeated n-grams?

4.3.1.2 Determining the Key

First we find the key length:

> 78. ... In a simple cipher text repetitions are more frequent and are merely accidental, that is to say, rarely occur at regular intervals, whereas in the composite cipher text [i.e. Vigenère cipher] these repetitions occur less often but usually at intervals such that the number of letters between them is divisible by a number which gives the number of letters in the key.
>
> ...
>
> 80. ... We now try to determine the number or letters in the key. For this purpose we look for all repetitions of two or more ciphers in the enciphered text, then count the interval from one another of like repetitions; write these out beneath the enciphered text with numbers indicating the intervals, and endeavor to break this number up into its factors.
>
> For instance, we find 16 letters between two repeated ciphers `UL`, we add two ciphers to the number 16 and get 18 as the interval for the repetition `UL`.
>
> We write `UL` $= 18 = 2 \cdot 9 = 3 \cdot 6$.
>
> According to section (76) we can infer from this that the key contains either two, nine, three, or six letters.
>
> Since the number of letters in the key is still uncertain and ... the repetition of the ciphers `UL` might be purely accidental, we write out one by one all the repetitions of several ciphers, noting the number representing their intervals, and break these numbers down into their factors.
>
> Those factors which are found most frequent indicate the number or letters in the key. [7, pp.31-32 (36-37)]

Reading Questions:

- Why did Kasiski add two to the number of characters between the two repeated copies of `UL`?

- The three letter cipher `ULV` is also appears more than once, after counting the number of characters from one to the next how many letters should you add to that count before factoring it? What about for the repeated cipher `UIIY`?

- Go through the entire cipher text in Example 4.3.1, look for all the repeated ciphers which are at least two characters long, then for each one write down the distance between them and the factors of those distances. What number(s) pop up most often? Why does this make sense?

Finally we find the key: In what follows Kasiski makes use of what he calls *the key table* which is a rearranged Vigenère table with plain text letters on the left, cipher letters along the top and the possible key letters on the interior. So it tells us that if we have a cipher letter `J` which we believe represents a plain text letter *i* then looking in the table we can see the corresponding key letter was a `B`.

		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*
12.22%	*e*	*W*	*X*	*Y*	*Z*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*
9.28%	*t*	*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*
8.06%	*a*	*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*
7.62%	*o*	*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*
7.10%	*i*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*
6.82%	*n*	*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*
6.45%	*s*	*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*
5.91%	*r*	*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*
5.76%	*h*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*
4.19%	*l*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*
3.93%	*d*	*X*	*Y*	*Z*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*

Figure 4.3.2: Kasiski Key Table

- Check that you understand the *key table* by trying to find the key for this message based on the given plain text and cipher text.

```
plain:  winter is at a close
CIPHER: OXEBRX AH RB N IDDJM
```

- Why do you think that Kasiski ordered the plain text characters the way he did?
- Why do you think that he didn't include all the possible plain text letters?

Finding the key:

> 85. ... If we have determined according to section 80, the number of letters in the key, we divide the cipher text into rows containing as many ciphers as the key contains letters and number the several columns. In theory those ciphers which occur most frequently in a given column should signify the letter e in the text;
> ...
> 86. After we have divided the cipher text into rows according to the number of letters in the key and have numbered the several columns, we take out all ciphers from column 1 and note the number o£ occurrences, then we do the same for columns, 2, 3, etc. [See Example 4.3.1]
> If we have found in this way the cipher `U`, let us say, occurs seven times, cipher `S` three times and cipher `F` three times in column 1, we copy from the key table letters beneath the ciphers `U, S` and `F` on a separate sheet of paper as follows:
>
> ```
> 7 U - Q B U G M H C D N J
> 3 S - O Z S E K F A B L H
> 3 F - B M F R X S N O Y U
> ```
>
> Among these letters must be found the first letter of the key. The letter `B` occurs as second letter under cipher `U`, as eighth under cipher `S`, and as first under cipher `F`. Hence `B` is combined seven times with *t* to form

> U, three times with r to form S, and three times with e to form F. From this we must conclude that B is the first letter of the key. The letter M also has a more remote probability of being the first letter of the key since it occurs as fifth under U, and as second under F ... . In the same way we just found the first letter of the key, we now look for the other letters. [7, pp.34-35 (39-40)]

Reading Questions:

- In the columns labeled 2 in the example we have been working with three most common ciphers are I, E, and L appearing 6, 5, and 4 times respectively. From this Kasiski would have us write down the following from the columns of the key table:

  ```
  6 I - E P I U A V Q R B X F
  5 E - A L E Q W R M N X T B
  4 L - H S L X D Y T U E A I
  ```

  Which two possible key letters appear in all three rows?

- In the columns labeled 3 in the example we have been working with three most common ciphers are V, K, and I appearing 5, 4, and 4 times respectively. Follow Kasiski's lead and write down the corresponding columns from the key table:

  ```
  6 V -
  5 K -
  4 I -
  ```

  Which possible key letters appear in all three rows?

- In the columns labeled 4 in the example we have been working with three most common ciphers are Y, P, and W appearing 4, 4, and 3 times respectively. Repeat the steps from the previous two questions to try and find the fourth key letter.

- In the columns labeled 5 in the example we have been working with three most common ciphers are B, W, and M appearing 4, 3, and 3 times respectively. Repeat the steps from the previous two questions to try and find the fourth key letter.

- In the columns labeled 6 in the example we have been working with three most common ciphers are F, V, and U appearing 5, 3, and 3 times respectively. Repeat the steps from the previous two questions to try and find the fourth key letter.

- Put together all the possible key letters we get for all the columns of the message, do they (or at least can they) spell "BERLIN" as we expect them to?

Checkpoint 4.3.3. The following text was enciphered using a key of CIPHER. Follow Kasiski's steps to see if you could have found this for your self.

```
 (1) UQCJIR TIBPWJ UQCNYC CZIYEE UNDYQR VQDUME VWPJSE
 (2) HMHZSI QNIOIF TLTYFR KATTIR WFLHWE QTDUKV TBWLWR
 (3) OMBHRL RBDALR VXTYMF FBWLTC CKTDLZ EPPYED KAWHHY
 (4) GTSPRK JMLVVK JGVVZV TVDYWV UBXTEK KWCDEJ VPPASW
 (5) CXGLPR VMLOSD JMGLWG GKILHR PLPMVZ GVSASN JWBOIF
 (6) YMSHHV DBDMKI CBXAYU GJJARF YPTMIC VPXTWV NNPUME
 (7) HMGPSI CVSALR VIGHQZ UEPZLZ UUPZXV TPTOMD UMAMPZ
 (8) IPILHR NICAII PAJTQF PMSHXL TVZLCR PLHHMU TMIBVE
 (9) KVVASR TIBPWZ CUPACF WZDYHV TABVRJ GQVUIL TIGHQZ
(10) UUTYIC AVDKHV FPXZLV CLPZQL EPPZXF UINCII AODVHR
(11) PLHPKE GLIVLZ OEXALY KAWHRU VWALEU VPTDEP
```

Hint.

1. Look for repeated strings.
2. For each pair of repeated strings write them down and next to them write down the number of characters from the start of one to the start of the next and the factors of that number.
3. Assuming that the most common factor is 6, which we know it should be, use the subset cell to find the most common letters in each column.
4. Use these letters and the key table (Figure 4.3.2) to try and find the key, which should be `CIPHER`.

Checkpoint 4.3.4. Follow Kasiski's steps to find the key for this message. Then use the Vigenère cipher cell to decipher it.

```
 (1) IERSDO EDPRHZ AMWXBU OEFAYX IEOPOO EWWUOG UNYRQJ
 (2) OPFIGX BFHUQP SMMFTG TEZYZJ SYUKWK BIIROE FDILFT
 (3) MVWEXY WFFHGE OYHFFO FYBYWA STZKBZ GGASSI BFSIQR
 (4) QDPZBT TESCSV RUUMKO TWKUKT ZOSZBT TAVCME HSARZS
 (5) USOPWW ZIBYSV DOFWIJ YUMDLR ZFKAFH WCQCJR DPDKBV
 (6) OGFHWP MVHBRR SYCCOF FOLRMW WEETKE JASPFT ZOZVDY
 (7) KWSCVV UNFINO PVYAQW JSOYHJ QLDDPR HGTECS VXGPML
 (8) DSVXVV YWSCNI CZZEUO AJWGKH WCBZZY RUJDPV FXZEOD
 (9) PRHYAU ACFZJZ GSLOFG SEUEFM MRBVYP WBQFIF PEKSZV
(10) TBDAHB QMOGQC GXDVFF MTAYVN WGTOFO EYCZFH WZWJGR
(11) ESAYVF TFGCZK AVQEQT HVITSQ ANSVMM SYIILR BYSUUG
(12) ZOAKDB IEJCWW HUQKAX OUCZNU LKAKCF MYAXOV LNOTDI
(13) EYOGFH WUQEUF IIKRER GQMRLK OEOARO MXLYWZ EEDPKF
(14) ACXELO TPOGML GCAKVR YUKUMK SRDHSN VFRBGB LCMZHU
(15) QRMZWE HUQRWK AFBJTI URPRRH DGWNBY SHZFGB BLBNFE
(16) HRQCWC BELYZV JRMLZS ATVNDA UDMIOA PBABBY DUULAZ
(17) XVPHDI WNNFFR HEJLME SNFHSW IJYBRS LOMCSK ULWNBF
(18) OPAUFD ZPKUQR WDPVAR ZSWOUV RYUTLV MDCEQT ZKVJZN
(19) HEKYNK VRQLWW MEHFBH AVQGDR PEHBQM SQQVWX WWHUQS
(20) GMQVHL AFVKZK OTZAFG PFVNPL GKLVRU UMOSBY VBZOJC
```

Hint.

1. Look for repeated strings.
2. For each pair of repeated strings write them down and next to them write down the number of characters from the start of one to the start of the next and the factors of that number.

3. Use the subset cell to find the most common letters in each "column" as if you had split up the cipher according to the length you found in the previous step.

4. Use these letters and the key table (Figure 4.3.2) to try and find the key.

YouTube: `www.youtube.com/watch?v=TxClRjnRNJw`

Figure 4.3.5: Modern look at Kasiski

Sage cell for calculating subsets of letters in a message:

```
@interact
def subset_count(
                    text=input_box('Place␣the␣text␣that␣you␣
                        would␣like␣to␣analyze␣here!!!',
                    label="Enter␣your␣message", height=3,
                        width=50, type=str),
                    Subsets=[1..10]):
    message =
        AlphabeticStrings().encoding(str(text.encode('ascii','replace')))
    S = Subsets
    sets = [""]*S
    for i in range(0,len(message)):
        sets[i%S] += str(message[i])
    for i in range(S):
        tmp_message = sets[i]
        count = {}
        for i in range(len(tmp_message)):
            c = tmp_message[i:i+1]
            if c in count: count[c] += 1
            else: count[c] = 1
        print "The␣length␣of␣this␣subset␣is␣
            {}".format(len(tmp_message))
        count = sorted([(value, key) for key, value in
            count.items()])
        print "Character\tCount\t\tPercent"
        count.reverse()
        IC = 0
        for c in count[0:min(30,len(count))]:
            if len(c[1])>6:
                tab="\t"
            else:
                tab="\t\t"
            print c[1],tab,c[0],"\t\t␣%.4G" %
                (c[0]/len(tmp_message)*100.0)
            IC += c[0]*(c[0]-1.0)/len(tmp_message)^2
        print "And,␣the␣index␣of␣coincidence␣for␣this␣set␣is␣
            %.4G\n" %  (IC)
```

Sage cell for enciphering and deciphering the Vigenère Cipher:

```
import re
import textwrap
@interact
def _(m=input_box('sage', label="Enter␣your␣message",
    height=3, width=50, type=str),
         key=input_box('sage', label="Enter␣your␣key",
             height=1, width=20, type=str),
         mode = selector(['encipher','decipher'],
             buttons=True),
         spaces = selector(['yes','no'], buttons=True)):
    plain_alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    clean_message = str(m.encode('ascii','replace')).upper()
    if spaces == 'no':
        clean_message = re.sub('[^A-Z]','',clean_message)
    cipher_key =
        re.sub('[^A-Z]','',str(key.encode('ascii','replace')).upper())
    key_list = [plain_alpha.index(ch) for ch in cipher_key]
    if mode == 'decipher':
        key_list = [-1*k for k in key_list]
    cipher_text = ""
    key_counter = 0
    for ch in clean_message:
        try:
            tmp_pos = plain_alpha.index(ch)
            cipher_pos = (tmp_pos+key_list[key_counter])%26
            cipher_text += plain_alpha[cipher_pos]
            key_counter = (key_counter+1)%len(key_list)
        except:
            if spaces == 'yes':
                cipher_text += ch
    print "\nHere␣is␣your␣output:\n"
    if spaces == 'yes':
        print textwrap.fill(cipher_text, 42)
    else:
        for i in xrange(0,len(cipher_text),6):
            print cipher_text[i:i+6],
            if (i+6)%42 == 0: print "\n"
```

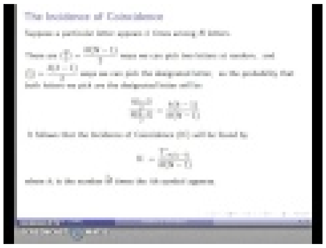
4.3.2 (*)Babbage's Contributions

Insert Information on Babbage's Cryptologic Work ...

4.4 (*)A Statistical Approach

Objectives

- Friedman's Use of Statistics
- Sukhotin's Algorithm



YouTube: `www.youtube.com/watch?v=raNO806R4yc`

Figure 4.4.1: Incidence (or Index) of Coincidence

Do You Feel Logical?

1. Encipher this message by hand using Falconer's transposition cipher and a key of `{ACB, BCA, CBA, BAC}`

> "A quick movement of the enemy will jeopardize six gunboats."

2. Decrypt this transposition cipher with crib `WEATHER`:

```
HIPYEN TAUTCI WSDWPF ETTEEA ARONRN ENPMNC RTEPTH
RERHR  EMTCA  PPHHI  OREOL  FAIEL  TETCA  RNRNF OSIAN
MYWRR  RTNIU  CEDNN  ASIAO  RLSFA  IITHU  TDENA HASOD
HYAEQ  RNTUA  FETRE  TTENR  RITDO  YHNDT  IMOER
```

3. Decrypt this transposition cipher with crib `AFGHANISTAN`:

```
HATSNNI OLHHHIV WLAOINE AYSUMA OPNADA RGTLCF
ERRDRG  YIEHEH  ENHLYS  SGFYOT UPGRIN HITDTI IYRAHN
DHWVAI  CAHEVP  AMOHUA  IIIEEE DWHVEC RDCIBR ONIGEE
```

4. Decrypt this transposition cipher:

```
IQAY ASHE PECV FIIR NUJA UCND
LKX OYD  ZLE  ITA  MWG  BAR
```

5. Encipher this message by hand with a Vigenère and key of `DUNGEON`

> "The wizard quickly jinxed the gnomes before they vaporized."

6. Decrypt this message enciphered with a Vigenère cipher by using the techniques you learned from Falconer and Kasiski.

```
HS OOE BGHE RCI LG JO SOF KCFCXRSF, KYR
KXDDGMESF DSS TYZAU KE BQ. DHC-VGPBVI ZBG
L, MOMOC GVBPSV, YQ KJSNV JO SOF CACMOY OV
YFF OOPHKXR. HJOJ QQXDWUDPR QP L QQEAZG YQ
QQWQCTDLPNO MSF-BZCOC LBF K DWPQWS NKCUG
KTFA CTHVSYU-TYZA, ERPSTPFZNI QITXTGJOO,
OPN TZNEXWPKESF LJ HYY MFQKO KKXOCYC. DC
FODWTKMZG SY SXOCM YKJ KGBP HJO
LDCBEAGXEG, CXO GQ WZRGBLHG NTR VRP HGBXG
UOPA YRPB FSGWFOO PGDHSGX FG, VRLH VRP
PCBROKX HOU MZBEVFRGN FDQX EVG CACV, KYR
YO LH QXNS GXESTOO WPDZ DQCDSUCTCP. DSOV
```

```
FPFA OGSPSYU K WZJGN XM VRTBIC CCWXO TTYX
HJO SCVOW, OPN ZB VRP TQVWCYSYU OYCBKXR
GJOCZQMV VQVXSU PZZNYHSF WP KKDS GGFPFCV
MCZOD OPN ACTDXOPDPOWC. QCT K OOA YC HYY
HS YOCS DEDWNI PARVZMGN TB WXAOEUTBI KYR
NKJWPQ ZIV YFF RBZDGBEM VY EVG LPGV
KOJCXEOIO. EVCD OCPO, HS IBLRWKWZA LPUCX
EC UOEHNO OCYX LBF DZ OEMZAOYOOVO
ZITCPZXOD HQ YFF POH GWBCCWXOWPQD.
```

7. Decrypt this message enciphered with a Vigenère cipher by using the techniques you learned from Falconer and Kasiski.

```
NUTZSK BLYBAU JWUONX BWZFXG OCTUCS BWEXCZ IUSSQO
MFOBGR FAXBHJ IYCBMU GUTBHI JYTUBA HOKOIZ GUSJFE
BHJIUJ PHIFVK FHCFUR UBECOZ BMKSCK TILNCY GIXUOT
FMNBXX FXADYJ ICSUIC BHZUIG WIOENN FGUSNO GCIBNO
PHIPHY FKAFHZ VJUOBO TXOTUY UYXTBK MYLUHK XIXMYG
OMZIYI JNEPZN JMLPLK GUZIYX TUTENU PEAQBO TLKTCJ
FHIFUZ TORMCB BHYJMR BHJOYG SWNBLR FMZPHY POZIWG
SIRJHG UBOTCY MUTECY BPKSSY JHMVFG SITFCZ DITTCY
UMUGFO UNRFYR TYZIUT UBKTYG TUTEUT ECYBVU VNZILK
FGOMYY MITHCZ TVXFUJ UBGUHU QIOONK YWKFXY BKABLZ
FLUGUS JFKJNO TMKQUX BNKEZX PGZIYS BCTMUT EVEBMI
BLIFFE QYXDYV UCHMYI SYKLIU ACTHCZ TQGZNN SIAHBG
XCREYX OYYTIL SYKEMG OXYMCS FULBPU SCZFLK TIXUIL
UBKNUX TBNFHZ IYBFAK UUZJIT BMSJAN UVKTOV QIYFXO
TMIBHZ PLGUFK BMZEQG SZOTBT PNXFYY PZGOSS BATJNA
EYGSYZ PVKTYK OHKBLZ IYCFMZ FLTFRZ SYSJNE XBKSYL
PLZNIA MNXJYY UUTEMG OXCIYX FUXFMU NYSJMK SUHMYL
SUSFVA JFJJHM TNKOUT UYJEOX JHMTOS NYXCSZ IYLVAO
UCBFML SISDBG SFKTNU OXATNG OXLFPK SGGZVK GIAOXO
OXKFXZ IYHSCY UFEQUR NYZUIH VNZIYC IIRFCY MUTEQO
UBZIYK YWKQNO PHUGNN JMCFMZ FLTQIO ONGOXG MCTFIL
IUXEQN JNKCYG DBUONN FMKBWU BMZJMI PPKSYJ XCZIUJ
FHYFOT EYXHLU XNNPZZ IYYXYK UGESNR FMUNOI IJXJTK
EVEUBK IIXUCI VFZVLO TNYPZK OARBHJ UBKTBX VVNFLK
PZZFHG UNGJHY UBKIYO HBZPZL JZZFYT PLZXYT USLFYZ
BHJGIX NMGOUR NIYUCS QYTFNX BVRFWU QJODYH VLZIYT
JHMUBK BCXXCZ ICZTZX BAXBHI F
```

8. Decrypt this message enciphered with a Vigenère cipher by using the techniques you learned from Falconer and Kasiski.

```
REZIAK VVISWB KSHYAL VISJMM CLTSHM QCKVHO OIDZBE
ARQHPF ROHQQQ CMOEDP TTMKHQ GUFVZT TPVVZZ RTPLZX
CEODVM MXTUDV LTJNYT BYEOTT TZVEYI ZVWWXC EETKII
OWDXBZ RBWAJT UFZTTT NZVNTE IMGXJT TTIKXV IZBMEX
JFYNEZ WCEEGM XEMDUC ODEMJA GGZXRO GALYEQ ENTMEW
ORMCOV XCEDTN FVZIRX PRHIOF UQIWOA FIMDTO EPIPVQ
ZTTDLN LDCTHP VLVDTT ZJIGFE JOXINT QSBYIM EPJKZR
BTTTJR GJNUPV TMKHQG BFMOSX DEVWOE XTUVRO SRDZDE
IYTDCI WDLMQW LVZDMI BYMNWA GSRRYF UCICPT WTTVZL
VDDTLL GZDFWM SEXOZX IEJDVQ HGDFJL EIWKLM EQXNVP
OTTPBZ LVDMRK FQKLUH PVHVLX IPRXRA EEWJWD BXTQEX
```

```
CAFLIP ACEZXP RHVRDX DVHVTF WQJVZS GABRRY HMSBVW
OEPXBJ EXCGGI TCDNID ZBMIGU UMCXDN MEWJMO IACBFI
SPQGQD IITIXB YQTNQL SESRLQ SOVSIT TTWCHI UYQMIG
DPTTZW MMSFXE ISOEAJ BDCHEF WWUSAR QSCTXD OZPARW
JRFDNR HYEZSC DXJTTT XRTZRI WQTLDH MSXIIK ADTLWS
MMMGRF VTTTTV ZQVDQP SVCOOO XXYIMA ZSWEIO OPTKZT
CEDQGK LDSFXU VXCEZX OYXRAE LMCPJN MCLKLZ GDTGFJ
ZADAGD SMNUCO NENBQV QERDNS IWJXZA XXVSCO HQTLXI
NORIPV FGIZSA ZAVSZD BJPZEB NPFAZV QGQNEN TADULG
CEJRQK IYTAIP ZRFORH TVIKFA GBYINO XKQEKJ FFWMGV
JBXTUJ IZMQSI CQJSFL QKLDNY NOIENP QMKZXZ DFDIJX
VTQLPZ GCAXBW JXARUV PKIIEP BMSCDT EXVKII SUIGZK
JTDTIU CHYODX PSATTT VLQWED RQGLZR MCLDCI EIAGGV
ZPMGMU OZYIXB YEIERU WIXRHU RPKSJK YTICPH YDTAFP
PTUDVZ AZNFDV JXZAPX TPAMIF XVXMOS BGWGIM LQIBVV
PNPTZV EXHODU SMIAFX WEADTT DCKSIC QAWFOD NSQITO
AODXSE IRTTPB VZZNEW WLPYSA BMFJOH QAMKXZ REQMDM
NPXPKV HDNFWM BITTTT KYEICQ DNIIXO SCQJMI GFWMIM
BHFDVV WROGAL SIGADV MCCDNO GMRWZD NNAVID NSPKFR
NIPTZR FGEZJU SIMORA MKXZRE IWXIOH QGBYII ISAIEG
ZDAKMI XCEIWW CIVNPU WLRYTT PBDEIY AUBYIN YYQWCW
HAPTCG PZTFTZ JADTTH CTLVBM HQJXJW AGSFRO HQGMJX
RAEDVC CGANDC IEAEII MEXVTU KMVJAO DIARRY ITPLTS
MRQRBV HOHQZM PXJASG MVQZNF LQKLNO YTWWXC EODUSM
IAFXWE WDNFWM TMKHQG
```

These will be helpful in completing these problems.

Sage cell for calculating subsets of letters in a message:

```
@interact
def subset_count(
                    text=input_box('Place␣the␣text␣that␣you␣
                        would␣like␣to␣analyze␣here!!!',
                    label="Enter␣your␣message", height=3,
                        width=50, type=str),
                    Subsets=[1..10]):
    message =
        AlphabeticStrings().encoding(str(text.encode('ascii','replace')))
    S = Subsets
    sets = [""]*S
    for i in range(0,len(message)):
        sets[i%S] += str(message[i])
    for i in range(S):
        tmp_message = sets[i]
        count = {}
        for i in range(len(tmp_message)):
            c = tmp_message[i:i+1]
            if c in count: count[c] += 1
            else: count[c] = 1
        print "The␣length␣of␣this␣subset␣is␣
            {}".format(len(tmp_message))
        count = sorted([(value, key) for key, value in
            count.items()])
        print "Character\tCount\t\tPercent"
        count.reverse()
        IC = 0
        for c in count[0:min(30,len(count))]:
            if len(c[1])>6:
                tab="\t"
            else:
                tab="\t\t"
            print c[1],tab,c[0],"\t\t␣%.4G" %
                (c[0]/len(tmp_message)*100.0)
            IC += c[0]*(c[0]-1.0)/len(tmp_message)^2
        print "And,␣the␣index␣of␣coincidence␣for␣this␣set␣is␣
            %.4G\n" %  (IC)
```

Sage cell for enciphering and deciphering the Vigenère Cipher:

```
import re
import textwrap
@interact
def _(m=input_box('sage', label="Enter␣your␣message",
    height=3, width=50, type=str),
        key=input_box('sage', label="Enter␣your␣key",
            height=1, width=20, type=str),
        mode = selector(['encipher','decipher'],
            buttons=True),
        spaces = selector(['yes','no'], buttons=True)):
    plain_alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    clean_message = str(m.encode('ascii','replace')).upper()
    if spaces == 'no':
        clean_message = re.sub('[^A-Z]','',clean_message)
    cipher_key =
        re.sub('[^A-Z]','',str(key.encode('ascii','replace')).upper())
    key_list = [plain_alpha.index(ch) for ch in cipher_key]
    if mode == 'decipher':
        key_list = [-1*k for k in key_list]
    cipher_text = ""
    key_counter = 0
    for ch in clean_message:
        try:
            tmp_pos = plain_alpha.index(ch)
            cipher_pos = (tmp_pos+key_list[key_counter])%26
            cipher_text += plain_alpha[cipher_pos]
            key_counter = (key_counter+1)%len(key_list)
        except:
            if spaces == 'yes':
                cipher_text += ch
    print "\nHere␣is␣your␣output:\n"
    if spaces == 'yes':
        print textwrap.fill(cipher_text, 42)
    else:
        for i in xrange(0,len(cipher_text),6):
            print cipher_text[i:i+6],
            if (i+6)%42 == 0: print "\n"
```

Chapter 5

(*) An Industrial Revolution

Objectives

- Telegraph Codes and Ciphers?
- Enigma
- Lorenz?
- Purple?
- Also included Stinson episode, use Friedman article
- Turing Papers?

5.1 World War I

Objectives

- ADFGVX
- George Zimmermann
- American Cipher Chamber

5.2 Between the Wars

Objectives

- Vernam One-Time Pad and Perfect Secrecy
- Henry L. Stinson, “Gentleman’s Mail”

- Rise of Enigma
- Polish Cipher Bureau

5.3 World War II

Objectives

- Bletchley Park and Enigma
- Lorenz Cipher, Colossus, Electronic Computing
- Japanese Ciphers

Chapter 6

Mathematics to the Rescue

Objectives

- Modular Arithmetic
- Affine
- Matrices
- Hill?

6.1 Affine Ciphers

Objectives

- Biography of Lester Hill
- Modulus
- Affine Cipher

6.1.1 Lester S. Hill

Insert bio here ...

6.1.2 Moduli

The cipher we will focus on here, Hill's Cipher, is an early example of a cipher based purely in the mathematics of number theory and algebra; the areas of mathematics which now dominate all of modern cryptography. Number theory has a long and rich history with many fundamental results dating all the way back to Euclid in 300 BCE, and with results found across the globe in different cultures. Number theory as we understand and use it today is due in large part to Carl Friedrich Gauss and his text *Disquisitiones Arithmeticae* published in 1801 (when Gauss was 24). Algebra (or more properly linear and abstract algebra) as it is going to be used here is much younger tracing its roots back only a couple hundred years to the early nineteenth century; here too much is owed to Gauss.

As with previous topics we will begin by looking at an original source text and trying to understand what it is saying. However, given the importance of this material to the rest of what we will be discussing in subsequent chapters, we will look at the material from a more modern perspective.

Cryptography in an Algebraic Alphabet.

By Lester S. Hill, Hunter College

1. *The Bi-Operational Alphabet*

Let $a_0,\ a_1,\ \ldots,\ a_{25}$ denote any permutation of the letters of the English alphabet; and let us associate the letter a_i with the integer i. We define operations of *modular addition* and *multiplication* (modulo 26) over the alphabet as follows:

$$a_i + a_j = a_r,$$
$$a_i\, a_j = a_t,$$

where r is the remainder obtained upon dividing the integer $i + j$ by the integer 26 and t is the reaminder obtained on dividing ij by 26. The integers i and j may be the same or different.

It is easy to verify the following salient propositions concerning the bi-operational alphabet thus set up:

(1) If $\alpha,\ \beta,\ \gamma$ are letters of the alphabet,

- $\alpha + \beta = \beta + \alpha$ and $\alpha\beta = \beta\alpha$ [commutative law]
- $\alpha+(\beta+\gamma) = (\alpha+\beta)+\gamma$ and $\alpha(\beta\gamma) = (\alpha\beta)\gamma$ [associative law]
- $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$ [distributive law]

(2) There is exactly one “zero” letter, namely a_0, characterized by the fact that the equation $\alpha + a_0 = \alpha$ is satisfied whatever the letter denoted by *alpha*.

(3) Given any letter α, we can find exactly one letter β, dependent on α, such that $\alpha + \beta = a_0$. We call β the “negative” of α, and we write: $\beta = -\alpha$.

(4) Given any letters $\alpha,\ \beta$ we can find exactly on letter γ such that $\alpha + \gamma = \beta$ [i.e. $\gamma = \beta - \alpha$ is unique].

(5) Distinguishing the twelve letters,

$$a_1,\ a_3,\ a_5,\ a_7,\ a_9,\ a_{11},\ a_{15},\ a_{17},\ a_{19},\ a_{21},\ a_{23},\ a_{25},$$

with subscripts prime to 26, as “primary” letters, we make the assertion, easily proved: If α is any primary letter and β is any letter, there is exactly one letter γ for which $\alpha\gamma = \beta$.

(6) In any algebraic sum of terms, we may clearly omit terms of which the letter a_0 is a factor; and we need not write the

letter a_1 explicitly as a factor in any product.

2. An Illustration

Let the letters of the alphabet be associated with the integers as follows:

a	b	c	d	e	f	g	h	i	j	k	l	m
5	23	2	20	10	1	8	4	18	25	0	16	13
n	o	p	q	r	s	t	u	v	w	x	y	z
7	3	1	19	6	12	24	21	17	14	22	11	9

or, in another convenient formulation:

0	1	2	3	4	5	6	7	8	9	10	11	12
k	p	c	o	h	a	r	n	g	z	e	y	s
13	14	15	16	17	18	19	20	21	22	23	24	25
m	w	f	l	v	i	q	d	u	x	b	t	j

It will be seen that

$$c + x = t,\ j + w = m,\ f + y = k,\ -f = y,\ -y = f,\ etc.$$
$$an = z,\ hm = k,\ cr = s,\ etc.$$

The zero letter is k, and the unit letter is p. The primary letters are: $a\ b\ f\ j\ n\ o\ p\ q\ u\ v\ y\ z$.

Since this particular alphabet will be used several times, in illustration of further developments, we append the following table of negatives and reciprocals:

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

The solution to the equation $z + \alpha = t$ is $\alpha = t - z$ or $\alpha = t + (-z) = t + v = f$.

The system of linear equations: $o\,\alpha + u\,\beta = x$, $n\,\alpha + i\,\beta = q$ has solution $\alpha = u$, $\beta = o$, which may be obtained by the familiar method of elimination or by formula. [5, pp.306-308]

Reading Questions:

- Hill starts by describing how we will add and multiply with the alphabet, looking at his description why in his illustration does $j + w$ which should be $25 + 14 = 39$ (see Hill's Correspondence) come out to be m which is 13?
- In his illustration he also says hm which should be 4 times 13, or 52, is k which is 0, why is this the case?

- Along the same lines, why does $f + y$ equal k and why does an (a times n) equal z?

- Thinking about your previous answers, what are the values of the following: $j + z$, nf, $au + j$, and $bv + jw$.

In this section of text Hill has introduced us to the idea of modular arithmetic and modular equivalence, in particular the idea of equivalence modulo 26. This is a concept which will be central to most everything else we do so we need to spend a little more time trying to precisely understand modular equivalence.

Definition 6.1.1 Modular Equivalence. If n is a positive integer then we say that two other integers a and b are **equivalent modulo n** if and only if they have the same remainder when divided by n, or equivalently if and only if $a - b$ is divisible by n, when this is the case we write

$$a \equiv b \pmod{n}.$$

The number n is called the *modulus*.

Example 6.1.2. Suppose that $n = 14$, then $36 \equiv 8 \pmod{n}$ because $36 = 2 \cdot 14 + 8$ and $8 = 0 \cdot (14) + 8$ so we get the same remainder when we divide by $n = 14$. Alternately, we can observe that $36 - 8 = 28$ and $28 = 2 \cdot (14)$ is divisible by $n = 14$.

Using the same value for n we get that $3 \cdot 5 \equiv 1 \pmod{n}$ because $15 = 1 \cdot (14) + 1$, so the remainder when $3 \cdot 5$ is divided by n is 1.

Test your understanding by filling in the rest of this multiplication table:

$\times_{14}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
2	0	2	4	6	8	10	12	0	2	4				
3	0	3	6	9	12	1	4	7	10	13	2	5	8	11
4	0	4	8	12		6		0						
5	0	5	10	1	6	11								
6	0	6	12	4	10	0								
7	0	7	0	7										
8	0													
9	0													
10	0													
11	0													
12	0													
13	0													

Table 6.1.3: Multiplication Modulo 14

- What is strange or different about the row for 7? Why do you think all the remainders come out this way?

- What is the difference between the even and odd rows (excluding row 7)?

Finally, fill in this addition table for addition modulo 14.

$+_{14}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1													
2	2													
3	3													
4	4													
5	5													
6	6													
7	7													
8	8													
9	9													
10	10													
11	11													
12	12													
13	13													

Table 6.1.4: Addition Modulo 14

Definition 6.1.5 Additive Identity. We call 0 the *additive identity* because for all a and all possible moduli n we get

$$a + 0 \equiv a \pmod{n}.$$

Definition 6.1.6 Additive Inverse. We say that a and b are *additive inverses* modulo n if

$$a + b \equiv 0 \pmod{n},$$

and we write $b = -a$.

Definition 6.1.7 Multiplicative Identity. We call 1 the *multiplicative identity* because for all a and all possible moduli n we get

$$a \cdot 1 \equiv a \pmod{n}.$$

Definition 6.1.8 Multiplicative Inverse. We say that a and b are *multiplicative inverses* modulo n if

$$a \cdot b \equiv 1 \pmod{n},$$

and we write $b = a^{-1}$.

Checkpoint 6.1.9. Look back at Example 6.1.2 and write down the pairs of additive and multiplicative inverses. Do all the numbers modulo 14 have additive inverses? Do all of them have multiplicative inverses?

Checkpoint 6.1.10. Write down another multiplication and addition table as you did in Example 6.1.2 but with a modulus of $n = 10$, so when you multiply and add you will always divide by 10 afterwards and write down the remainder. After you write down the tables write down the pairs of multiplicative and additive inverses. Do all the numbers modulo 10 have additive inverses? Do all of them have multiplicative inverses?

Reflection Questions: Look back at what Hill had to say and at the examples you have worked through when you used moduli of $n = 14$ and $n = 10$ as you think about the following questions.

- No matter which modulus you use, do all the numbers have additive inverses, i.e. numbers you can add to them in order to get 0?

- No matter which modulus you use, do all the numbers have multiplicative inverses, i.e. numbers you can multiply them by in order to get 1?

- If you look at the numbers which do have multiplicative inverses how do they relate to those which Hill described as prime to 26?

Definition 6.1.11 Relatively Prime. We say that two integers are *relatively prime* if the largest positive integers which divided them both, their *greatest common divisor*, is 1. For example the greatest common divisor of 7 and 36 is 1 so they are relatively prime, however the greatest common divisor of 30 and 36 is 6 so they are not relatively prime.

Checkpoint 6.1.12. Which numbers, other than 7, that are less than 36 are *relatively prime* to 36?

Checkpoint 6.1.13. Which numbers less than 14 are *relatively prime* to 14? How do these compare to the list of numbers which have *multiplicative inverses*?

Checkpoint 6.1.14. Which numbers less than 10 are *relatively prime* to 10? How do these compare to the list of numbers which have *multiplicative inverses*?

Checkpoint 6.1.15. Which numbers less than 26 are *relatively prime* to 26? How do these compare to the list of numbers which have *multiplicative inverses*? (You will want to use Figure C.0.13.)

6.1.3 Affine Cipher

Definition 6.1.16 Affine Cipher. An *affine cipher* is a cipher with a two part key, a multiplier m and a shift s and calculations are carried out using modular arithmetic; typically the modulus is $n = 26$. Characters of the plain text are enciphered with the formula

$$CIPHER \equiv m(plain) + s \pmod{26},$$

and characters of the cipher text are deciphered with the formula

$$plain \equiv m^{-1}(CIPHER - s) \pmod{26},$$

or

$$plain \equiv m^{-1}CIPHER - m^{-1}s \pmod{26}.$$

Note that the multiplier m must be relatively prime to the modulus so that it has a multiplicative inverse.

Checkpoint 6.1.17. Let's encipher the message "`hello world`" with an affine cipher and a key of $m = 5$ and $s = 16$; assume that we match up the alphabet with the integers from 0 to 25 in the usual way so that `a` is 0, `b` is 1, `c` is 2, etc.. In this way the letter `h` is replaced by the number 7 and when we encipher it we get

$$5 \cdot 7 + 16 \equiv 25 \pmod{26}$$

and 25 is `Z`, so plain `h` becomes cipher `Z`. Next `e` is replaced by 4 and we get

$$5 \cdot 4 + 16 \equiv 10 \pmod{26}$$

and 10 is `K`, so plain `e` becomes cipher `K`. The plain `l` corresponds to 11 and

$$5 \cdot 11 + 16 \equiv 19 \pmod{26},$$

which is `T`, that is plain `l` is replaced by cipher `T`. Try to encipher the rest of the message on your own, you will want to use Figure C.0.13 to help you with the multiplication modulo 26.

Checkpoint 6.1.18. Now let's decipher the message `AJINF CVCSI JCAKU` which was enciphered using an affine cipher and a key of $m = 11$ and $s = 4$. Note that $m^{-1} \equiv 19 \pmod{26}$ and $-s \equiv 22 \pmod{26}$. Take the `A` and replace it by 0 and then using the formula above we get

$$19(0 + 22) \equiv 2 \pmod{26}$$

so we replace cipher `A` with plain text `c`. The `J` is replaced by 9 and

$$19(9 + 22) \equiv 17 \pmod{26}$$

therefore cipher `J` becomes plain `r`. To use the other formula for deciphering we need $m^{-1}s \equiv 2 \pmod{26}$. Then converting the cipher `I` to 8 we get

$$19(8) + 2 \equiv 24 \pmod{26}$$

which is plain `y` or with the next letter `N` we get

$$19(13) + 2 \equiv 15 \pmod{26}$$

which is `p`. Try to decipher the remaining characters in the message on your own.

Checkpoint 6.1.19. Encipher the message "`a fine affine cipher`" using the key $m = 17$ and $s = 12$.

Checkpoint 6.1.20. Decipher the message `RXGTM CHUHJ CFWM` which was enciphered using the key $m = 3$ and $s = 7$.

To *decrypt*, as opposed to just decipher, an affine cipher you can use the techniques we learned in Chapter 2 since they are a type of monoalphabetic substitution cipher. However, we can also take advantage of the fact that it is an affine cipher.

```
DZIUI UDZYH ILUDO HHIBY GITZY LSYUU OQYDZ ODEYE IJJPY GLWTD
IDEOU YBGIT ZYLYP EIDZO AYWKH BIBYH KLDZY SMJDI TJIYL OBPHK
MLDYY BHKLD ZYUZI HDUIB GYIDI UOSKB KOJTZ OXYDI GEYGO BODDO
GAIDE IDZXO UIGHL YCMYB GWOBO JWUIU XMDEY GOBOJ UKMUY DZODO
BOJWU IUDKL YGKVY LDZYK LIQIB OJAYW UKDZO DEYGO BPYGI TZYLD
ZYSYU UOQY
```

Analyzing this we get that the most common characters are `Y`, `D`, `I`, `O` and `U`; the most common bigrams are `DZ`, `ZY`, `YG`, and `OB`; the most common trigrams are `DZY`, `OBO`, `LDZ`, and `DZO`. Therefore it is reasonable to assume that `DZY` is `the`, `Y` is `e`, and `D` is `t`. So when this was enciphered we have to of had

$$\begin{aligned} 24 &\equiv m \cdot 4 + s \pmod{26} \\ 3 &\equiv m \cdot 19 + s \pmod{26} \end{aligned}$$

Subtracting the second expression from the first we get

$$21 \equiv m \cdot -15 \pmod{26}$$

or

$$21 \equiv m \cdot 11 \pmod{26}.$$

Looking at the multiplication table modulo 26 we can see that $m = 9$ since $9 \cdot 11 \equiv 21 \pmod{26}$. Substituting $m = 9$ into the first equation above we get

$$24 \equiv 9 \cdot 4 + s \pmod{26}$$

which simplifies to

$$24 - 10 \equiv s \pmod{26}$$

so that $s = 14$. We can then get the inverse keys $m^{-1} \equiv 3 \pmod{26}$ and $-m^{-1}s \equiv 10 \pmod{26}$. Using these with the affine cipher cell we get the deciphered message:

```
thisi sthef irsta ffine ciphe rmess ageth atwew illde crypt
itwas encip hered witha keyof ninef orthe multi plier andfo
urtee nfort heshi ftsin ceiti samon oalph abeti cweca natta
ckitw ithba sicfr equen cyana lysis butwe canal souse thata
nalys istor ecove rtheo rigin alkey sotha tweca ndeci phert
hemes sage
```

Or, in a more readable form

> “this is the first affine cipher message that we will decrypt ...”

Checkpoint 6.1.21. Try to decrypt this message which was enciphered using an affine cipher.

```
IPAGS WWANP YMFZC TAEWI PZFGC TZANW QWCNN YCMAY ECGGA CZAHV
QWQIE PFACN ZFGWI KYPMC DCGZB YBACP HPIHH YPMIL ANCPI LADTI
NWQHC QGEIN KFCHV AAPCP AXFCS GZYPM IPAWQ EYTAF CHCDN ACHQM
IPASB GZCYN GCPHZ FAGIS PHITZ FADIO KYPMI TZFAF CDDHI INGIW
AZYWA VATIN AZIDH WAZFC ZZFAG ANLCP ZGFCH CDGIN AZYNA HYFCH
NYGAP TNIWW QGACZ CPHEC GKPIO KYPMI SZZFA CGFAG ITWQB YBAEF
APYGS HHAPD QFACN HZFAO DCPMI TZFAV ADDYD IIKAH CZZFA ODIOK
YZECG CUSCN ZANZI ZEADL AZFYG OISDH PIZVA CLYGY ZINCZ GIDCZ
ACPFI SNCBC ZYAPZ ALYHA PZDQC PHBIG GYVDQ CPCDD PYMFZ GYZZY
PMEYZ FCENQ TCOAY EAPZI SZYPZ IZFAF CDDCP HIBAP AHZFA HIINZ
IWQCG ZIPYG FWAPZ YZECG GFAND IOKFI DWAGE FIGZI IHSBI PWQGZ
AB
```

Hint.

- First use frequency analysis to identify at least two of the letters in the message.
- With your two letters set up two equations like this:

$$CIPHER \equiv m(plain) + s \pmod{26}.$$

- Subtract the second equation from the first and try to find m.
- Substitute your value for m into the first equation and use it to find s.
- Now that you have the key you should be able to decipher the message as you had previously.

Sage N-Gram Cell:

```
@interact
def ngram_count_simple(
                    text=input_box('Place␣the␣text␣that␣you␣
                        would␣like␣to␣analyze␣here!!!',
                    label="Enter␣your␣message", height=5,
                        width=50, type=str),N=[1..6]):
    message =
        AlphabeticStrings().encoding(str(text.encode('ascii','replace')))
    count = {}
    for i in range(len(message)-N+1):
        c = message[i:i+N]
        if c in count: count[c] += 1
        else: count[c] = 1
    print "The␣length␣of␣your␣message␣is␣
        {}".format(len(message))
    count = sorted([(value, key) for key, value in
        count.items()])
    print "\tRank\tN-Gram\tCount\tPercent"
    count.reverse()
    rank = 1
    for c in count[0:min(30,len(count))]:
        if len(c[1])>6:
            tab="␣"
        else:
            tab="\t"
        print str("\t"+str(rank)+"."),tab,c[1],tab,c[0],"\t␣
            %.4G" % (c[0]/len(message)*100.0)
        rank += 1
    if len(count)==26 and N==1: print "Pangram!!!"
```

Affine Cipher Cell:

You can use this Sage Cell to encipher and decipher messages that used an *affine cipher*. To decipher you will need to use the second formula listed in Definition 6.1.16. Also, be sure you understand how to encipher and decipher by hand.

```
@interact
def _(me=input_box('sage', label="Enter␣your␣message",
    height=3, width=50, type=str),m=[1..25], s=[0..25]):
    S = AffineCryptosystem(AlphabeticStrings())
    clean_text =
        S.encoding(str(me.encode('ascii','replace')))
    try:
        e = S(m,s)
        cipher_text = e(clean_text)
        print "Your␣affine␣enciphering␣text␣is\n"
        for i in xrange(0,len(cipher_text),5):
            print cipher_text[i:i+5],
            if (i+5)%50 == 0: print "\n"
    except:
        print "error,␣likely␣in␣your␣key"
        cipher_text = clean_text
```

6.2 Hill's Cipher

Objectives

- Matrices
- Hill's Cipher

6.2.1 Matrices

Definition 6.2.1 Matrix. A *matrix* is a rectangular array of numbers such as

$$\begin{pmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{pmatrix}$$

which, in this setting, we will use to represent a transformation for enciphering and deciphering.

The cipher we will look at in this section, *Hill's Cipher* will work much like an affine cipher but will use matrices for the multiplier and shift and not just numbers. All the matrices we will use will be either square like this

$$\begin{pmatrix} 1 & 7 \\ 0 & 5 \end{pmatrix}$$

so that it has the same number of rows and columns or will be a special type of matrix called a **vector** which has only one column like this

$$\begin{pmatrix} 18 \\ 7 \end{pmatrix}.$$

Finally, all operations will be done using arithmetic modulo 26.

We can add and multiply matrices with one another or with vectors as long as the dimensions match.

Two 2 × 2 square matrices can be added like so

$$\begin{pmatrix} 23 & 3 \\ 0 & 15 \end{pmatrix} + \begin{pmatrix} 9 & 2 \\ 5 & 19 \end{pmatrix} = \begin{pmatrix} 23+9 & 3+2 \\ 0+5 & 15+19 \end{pmatrix}$$

where the corresponding entries are added together. Then completing the addition and reducing modulo 26 we get

$$\begin{pmatrix} 32 & 5 \\ 5 & 34 \end{pmatrix} \equiv \begin{pmatrix} 6 & 5 \\ 5 & 8 \end{pmatrix} \pmod{26}.$$

Likewise we can add two vectors

$$\begin{pmatrix} 17 \\ 12 \end{pmatrix} + \begin{pmatrix} 14 \\ 8 \end{pmatrix} \equiv \begin{pmatrix} 5 \\ 20 \end{pmatrix} \pmod{26}.$$

But, we cannot add a matrix to a vector since they are not the same size.

In order to multiply a matrix with a vector or a matrix with another matrix we need to multiply rows by columns like so

$$\begin{pmatrix} 8 & 1 \\ 5 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 10 \end{pmatrix} = \begin{pmatrix} 8 \cdot 1 + 1 \cdot 10 \\ 5 \cdot 1 + 3 \cdot 10 \end{pmatrix}.$$

Breaking that down a little more, the first entry on top is

$$\begin{pmatrix} 8 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 10 \end{pmatrix} = \begin{pmatrix} 8 \cdot 1 + 1 \cdot 10 \end{pmatrix}$$

and the second entry on bottom is

$$\begin{pmatrix} 5 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 10 \end{pmatrix} = \begin{pmatrix} 5 \cdot 1 + 3 \cdot 10 \end{pmatrix}.$$

When we simplify the result modulo 26 we get

$$\begin{pmatrix} 8 \cdot 1 + 1 \cdot 10 \\ 5 \cdot 1 + 3 \cdot 10 \end{pmatrix} \equiv \begin{pmatrix} 18 \\ 9 \end{pmatrix} \pmod{26}.$$

Similarly we can multiply two matrices together

$$\begin{aligned} \begin{pmatrix} 3 & 8 \\ 5 & 15 \end{pmatrix} \cdot \begin{pmatrix} 1 & 7 \\ 13 & 16 \end{pmatrix} &= \begin{pmatrix} 3 \cdot 1 + 8 \cdot 13 & 3 \cdot 7 + 8 \cdot 16 \\ 5 \cdot 1 + 15 \cdot 13 & 5 \cdot 7 + 15 \cdot 16 \end{pmatrix} \\ &\equiv \begin{pmatrix} 3 & 19 \\ 18 & 15 \end{pmatrix} \pmod{26}. \end{aligned}$$

Before moving on to see how Hill used these ideas to create a cipher practice your arithmetic with matrices and vectors, you will want to use Figure C.0.13 so that the multiplication is not so tedious.

Checkpoint 6.2.2. Carry out the arithmetic operations and simplify the results modulo 26.

$$\begin{pmatrix} 17 \\ 5 \end{pmatrix} + \begin{pmatrix} 13 \\ 9 \end{pmatrix}$$

Checkpoint 6.2.3. Carry out the arithmetic operations and simplify the results modulo 26.

$$\begin{pmatrix} 3 & 11 \\ 17 & 5 \end{pmatrix} + \begin{pmatrix} 9 & 25 \\ 1 & 4 \end{pmatrix}$$

Checkpoint 6.2.4. Carry out the arithmetic operations and simplify the results modulo 26.

$$\begin{pmatrix} 4 & 19 \\ 15 & 22 \end{pmatrix} \cdot \begin{pmatrix} 19 \\ 4 \end{pmatrix}$$

Checkpoint 6.2.5. Carry out the arithmetic operations and simplify the results modulo 26.

$$\begin{pmatrix} 21 & 17 \\ 0 & 11 \end{pmatrix} \cdot \begin{pmatrix} 14 \\ 5 \end{pmatrix}$$

Checkpoint 6.2.6. Carry out the arithmetic operations and simplify the results modulo 26.

$$\begin{pmatrix} 16 & 3 \\ 7 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 17 \end{pmatrix} + \begin{pmatrix} 2 \\ 7 \end{pmatrix}$$

Hint. Order of operations is the same for matrices and vectors as it is for numbers, so first multiply and then add.

6.2.2 Hill's Cipher

6.2.2.1 Enciphering with Matrices

Let's see how we can encipher `hill cipher` using a key matrix

$$m = \begin{pmatrix} 7 & 12 \\ 3 & 3 \end{pmatrix}.$$

Since it is a 2 × 2 matrix we will first break the message into groups of two letters

```
(hi) (ll) (ci) (ph) (er)
```

then to encipher each block we will convert each of these to a vector using $a = 0$, $b = 1$, $c = 2$, $\ldots$ *etc.* and multiply it by the matrix m. That is `(hi)` becomes

$$\begin{aligned} m \begin{pmatrix} h \\ i \end{pmatrix} &= \begin{pmatrix} 7 & 12 \\ 3 & 3 \end{pmatrix} \begin{pmatrix} 7 \\ 8 \end{pmatrix} \\ &\equiv \begin{pmatrix} 15 \\ 19 \end{pmatrix} \pmod{26} \\ &\equiv \begin{pmatrix} P \\ T \end{pmatrix}. \end{aligned}$$

Similarly we can encipher the `(ll)`:

$$\begin{aligned} m \begin{pmatrix} l \\ l \end{pmatrix} &= \begin{pmatrix} 7 & 12 \\ 3 & 3 \end{pmatrix} \begin{pmatrix} 11 \\ 11 \end{pmatrix} \\ &\equiv \begin{pmatrix} 1 \\ 14 \end{pmatrix} \pmod{26} \\ &\equiv \begin{pmatrix} B \\ O \end{pmatrix}. \end{aligned}$$

And for `(ci)` we get:

$$\begin{aligned} m \begin{pmatrix} c \\ i \end{pmatrix} &= \begin{pmatrix} 7 & 12 \\ 3 & 3 \end{pmatrix} \begin{pmatrix} 2 \\ 8 \end{pmatrix} \\ &\equiv \begin{pmatrix} 6 \\ 4 \end{pmatrix} \pmod{26} \\ &\equiv \begin{pmatrix} G \\ E \end{pmatrix}. \end{aligned}$$

Checkpoint 6.2.7. Finish enciphering `hill cipher` by enciphering the `(ph)` and the `(er)` just as we enciphered the other bi-grams.

Hint.

1. Change each pair of letters into a pair of numbers, this is a vector.
2. Multiply the vector by the matrix m.
3. Reduce the result modulo 26 and change back to letters, this is your cipher text.

Checkpoint 6.2.8. Try to encipher the message `linear madness` using the matrix

$$m = \begin{pmatrix} 15 & 0 \\ 17 & 7 \end{pmatrix},$$

note that since the message has odd length you will need to tack on an extra null letter, like a z.

Hint.

1. Change each pair of consecutive letters into a pair of numbers, this is a vector.
2. Multiply the vector by the matrix m.
3. Reduce the result modulo 26 and change back to letters, this is your cipher text.

Checkpoint 6.2.9. For a variation to Hill's Cipher you can both multiply by a matrix and add a shift, this make Hill's behave very much like an affine cipher. Try to encipher the word `shifty` by first multiplying by the matrix

$$m = \begin{pmatrix} 23 & 14 \\ 7 & 9 \end{pmatrix},$$

and then adding the vector

$$s = \begin{pmatrix} 10 \\ 3 \end{pmatrix}.$$

6.2.2.2 Matrix Inverses and Deciphering

We will be able to decipher messages which used Hill's cipher in exactly the same way we enciphered them once we can find **inverse matrices**.

Definition 6.2.10 Inverse Matrix. Given a matrix m the *inverse matrix* m^{-1} is the unique matrix such that

$$m \cdot m^{-1} = m^{-1} \cdot m = I$$

where I is the *identity matrix* (i.e. $I \cdot m = m \cdot I = m$ for all matrices m).

Consider the matrix

$$m = \begin{pmatrix} 8 & 1 \\ 5 & 3 \end{pmatrix},$$

the inverse for this will be

$$m^{-1} = \begin{pmatrix} 7 & 15 \\ 23 & 10 \end{pmatrix},$$

and if we multiply them together we get

$$\begin{aligned} m \cdot m^{-1} = \begin{pmatrix} 8 & 1 \\ 5 & 3 \end{pmatrix} \cdot \begin{pmatrix} 7 & 15 \\ 23 & 10 \end{pmatrix} &= \begin{pmatrix} 8 \cdot 7 + 1 \cdot 23 & 8 \cdot 15 + 1 \cdot 10 \\ 5 \cdot 7 + 3 \cdot 23 & 5 \cdot 15 + 3 \cdot 10 \end{pmatrix} \\ &\equiv \begin{pmatrix} 4 + 23 & 16 + 10 \\ 9 + 17 & 23 + 4 \end{pmatrix} \pmod{26} \end{aligned}$$

$$\equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \pmod{26}$$

which is the identity matrix since

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

for any values of a, b, c, and d. The matrix inverses and the identity matrix work just like the multiplicative inverses and multiplicative identities we discussed when we learned about affine ciphers and modular arithmetic; just as in that case, not all matrices will have a multiplicative inverse.

Definition 6.2.11 Determinant. The *determinant* of a 2×2 matrix

$$m = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

is equal to

$$det(m) = ad - bc.$$

The matrix m will have a multiplicative inverse m^{-1} if and only if $det(m)$ has a multiplicative inverse, in which case we get

$$m^{-1} = (det(m))^{-1} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

Going back to our previous example of

$$m = \begin{pmatrix} 8 & 1 \\ 5 & 3 \end{pmatrix},$$

we calculate the inverse as follows

$$\begin{aligned} m^{-1} &= (8 \cdot 3 - 5 \cdot 1)^{-1} \begin{pmatrix} 3 & -1 \\ -5 & 8 \end{pmatrix} \\ &\equiv 19^{-1} \begin{pmatrix} 3 & 25 \\ 21 & 8 \end{pmatrix} && \pmod{26} \\ &\equiv 11 \cdot \begin{pmatrix} 3 & 25 \\ 21 & 8 \end{pmatrix} && \pmod{26} \\ &\equiv \begin{pmatrix} 11 \cdot 3 & 11 \cdot 25 \\ 11 \cdot 21 & 11 \cdot 8 \end{pmatrix} && \pmod{26} \\ &\equiv \begin{pmatrix} 7 & 15 \\ 23 & 10 \end{pmatrix} && \pmod{26} \end{aligned}$$

which is what we had before.

Checkpoint 6.2.12. Check your understanding by finding the inverse of

$$m = \begin{pmatrix} 17 & 21 \\ 15 & 8 \end{pmatrix}.$$

Hint.

$$det(m) = ad - bc.$$

and

$$m^{-1} = (det(m))^{-1} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

We are now, finally, ready to decipher a message. Suppose the message `NDGUF SJV` had been enciphered using Hill's cipher with multiplier

$$m = \begin{pmatrix} 8 & 1 \\ 5 & 3 \end{pmatrix},$$

to decipher we split the message into pairs and multiply those pairs by

$$m^{-1} = \begin{pmatrix} 7 & 15 \\ 23 & 10 \end{pmatrix}.$$

So for the first two letters we get

$$m^{-1}\begin{pmatrix} N \\ D \end{pmatrix} = \begin{pmatrix} 7 & 15 \\ 23 & 10 \end{pmatrix}\begin{pmatrix} 13 \\ 3 \end{pmatrix} \equiv \begin{pmatrix} 6 \\ 17 \end{pmatrix} \pmod{26} \equiv \begin{pmatrix} g \\ r \end{pmatrix}.$$

And, for the next two letters of cipher text

$$m^{-1}\begin{pmatrix} G \\ U \end{pmatrix} = \begin{pmatrix} 7 & 15 \\ 23 & 10 \end{pmatrix}\begin{pmatrix} 6 \\ 20 \end{pmatrix} \equiv \begin{pmatrix} 4 \\ 0 \end{pmatrix} \pmod{26} \equiv \begin{pmatrix} e \\ a \end{pmatrix}.$$

Checkpoint 6.2.13. Finish deciphering the message `NDGUF SJV`.

Hint.

1. Change each pair of consecutive letters into a pair of numbers, this is a vector.
2. Multiply the vector by the matrix m^{-1}.
3. Reduce the result modulo 26 and change back to letters, this is your plain text.

Checkpoint 6.2.14. Decipher the message `TYEMC QDE` which was enciphered with the matrix

$$m = \begin{pmatrix} 7 & 4 \\ 0 & 3 \end{pmatrix}.$$

Hint.

1. Find the inverse matrix m^{-1}.
2. Change each pair of consecutive letters into a pair of numbers, this is a vector.
3. Multiply the vector by the matrix m^{-1}.
4. Reduce the result modulo 26 and change back to letters, this is your plain text.

Checkpoint 6.2.15. Decipher the message `URYMB FXFLS` which was enciphered by multiplying the plain text by

$$m = \begin{pmatrix} 7 & 4 \\ 0 & 3 \end{pmatrix},$$

and then adding the vector

$$s = \begin{pmatrix} 7 \\ 5 \end{pmatrix}.$$

Hint.

1. Find the inverse matrix m^{-1}.
2. Change each pair of consecutive letters into a pair of numbers, this is a vector.
3. Subtract the vector s from this vector.
4. Multiply the new vector by the matrix m^{-1}.
5. Reduce the result modulo 26 and change back to letters, this is your plain text.

In conclusion we can give Hill's cipher the following general definition.

Definition 6.2.16 Hill's Cipher. *Hill's cipher* is a cipher with a two part key, a multiplier m which is a square $n\times n$ matrix and a shift s which is a vector with n entries; typically all the arithmetic is done modulo 26. Characters of the plain text are enciphered n at a time with the formula

$$CIPHER \equiv m(plain) + s \pmod{26},$$

where *plain* is a vector of n characters. Likewise, characters of the cipher text are deciphered n at a time with the formula

$$plain \equiv m^{-1}(CIPHER - s) \pmod{26},$$

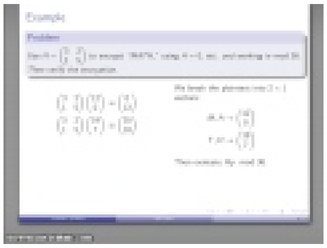
or

$$plain \equiv m^{-1}CIPHER - m^{-1}s \pmod{26}.$$

Note that the multiplier m must have a multiplicative inverse.

We end with two important observations. First, modern explanations of Hill's cipher focus on the simplest case when the matrix has dimension 2 × 2 and there is no shift. Second, you should be able to see that Hill's cipher is a variation on a affine cipher which enciphers multiple characters at a time in order to mix up the frequencies, this means it is a type of **polygraphic cipher**.

6.2.2.3 Alternate Explanation



YouTube: `www.youtube.com/watch?v=4RhLNDqcjpA`

Figure 6.2.17: Modern look at Hill's Cipher

Up Hill struggle?

1. Construct an addition and multiplication table modulo 6. Which pairs of numbers are additive inverses? Which pairs are multiplicative inverses, and how do they relate to 6?

2. Construct an addition and multiplication table modulo 13. Which pairs of numbers are additive inverses? Which pairs are multiplicative inverses, and how do they relate to 13?

3. Encipher `zygomorphic` using an affine cipher with key $m = 7$ and $s = 12$.

4. Encipher `quaquaversal` using an affine cipher with key $m = 17$ and $s = 3$.

5. Decipher `TWWFS NLVBY FVNBX GPNWY XWZVN BKLVM FSHFA FKWT` which was enciphered using an affine cipher with key $m = 15$ and $s = 23$.

6. Decipher `YTGXZ GHACB HQGBN XABFU ZAGXN TJJ` which was enciphered using an affine cipher with key $m = 3$ and $s = 7$.

7. Decrypt this message which was enciphered using an affine cipher:

```
EHWQC CIYKH EYKVK PIXKH ZKZKQ THZIP YGPXE KDFYX CZKXT IAOZI
TYKCX KAIRK XKFPX IYHZK CHXQE DAQMC KFVGZ ECEYY KDCKK BKXHE
IDCED HZKCN XEDUI PHZKW ZITKS MKCHE IDIPH ZKDKH ZKXTQ DFCMY
QHXQA IYNQD GQDFI PHZKA ITICC QTCAZ KYKCI PVQXI DYQMN KXHME
CQXKH IIXKA KDHED HZKYE DFCIP HZKNM VTEAQ DFQXK HIIED HEYQH
KTGAI DAKXD KFWEH ZNITE HEACQ DFPED QDAKH IVKPE HHEDU CMVJK
AHCPI XHZEC CKXEK CIPCO KHAZK CHZKG TKFZI WKRKX EDQDE DFEXK
AHPQC ZEIDH IQCED UMTQX QDFAI YNTKB NXIVT KYWZE AZUQR KYGPX
EKDFQ DINNI XHMDE HGIPF KYIDC HXQHE DUHZK RQTMK IPQPX KCZWK
QNIDQ YIDUH ZKYQD GWEHZ WZEAZ ZKWQU KFZEC TEPKT IDUVQ HHTKQ
UQEDC HAXEY K
```

8. Decrypt this message which was enciphered using an affine cipher:

```
ZTWFB JOJLT ZZZTW XVAJO KOJWZ MHOLJ KTHWT WKHJF HWMBZ ZHOHM
KIBHO QBOUJ TZBLB JVCRJ ZWHAH WXBOK IBZJL BLJWV SKHKI JKSBO
THQKI BSAJF BRITF IJOJL TZIJQ IBAQT WKIBR HOKIN XHGBO WHOZB
ZKTLJ KTHWR JZKIJ KHMJS OBAJK BRIHL IBOBZ SBFKB QJWQJ MOTBW
QKHRI HLIBH RBQJQ BUKHM XOJKT KVQBU VKWHR IBMBA KITLZ BAMJW
TWMBO THOJW QKIJK JOJLT ZRJZI TZLJZ KBOIB ITLZB AMATX IKBQJ
AJWKB OWZVL LHWBQ JKVOW PBNJW QZJTQ OBKVO WTWXK HJOJL TZTJL
JKNHV OHOQB OZLHW ZBTXW BVOJO JLTZL BOBAN WHQQB QITZI BJQJZ
LVFIJ ZKHZJ NGBON XHHQJ WQZTX WBQKH ITLRT KIITZ IJWQK HABJQ
KIBRJ N
```

9. Encipher `order more pizza` using Hill's cipher and the matrix,

$$m = \begin{pmatrix} 1 & 5 \\ 7 & 10 \end{pmatrix},$$

be sure to how your work.

10. Encipher `we need soda too` using Hill's cipher and the matrix,

$$m = \begin{pmatrix} 9 & 6 \\ 13 & 7 \end{pmatrix},$$

be sure to how your work.

11. Decipher `ZEMUM KJDIK NEWSX XHM` which was enciphered using Hill's cipher and the matrix,

$$m = \begin{pmatrix} 13 & 19 \\ 1 & 6 \end{pmatrix},$$

be sure to how your work.

12. Decipher `MRUKD QFDWI DHNL` which was enciphered using Hill's cipher and the matrix,

$$m = \begin{pmatrix} 7 & 17 \\ 2 & 5 \end{pmatrix},$$

be sure to how your work.

13. Decrypt this quote about "Chuck Norris" which was enciphered using Hill's cipher: `FYQAB YTAHX XYERO MQMFP NNBXN ARJFK HLSNK QVVME IFDUJ AVOUT ZYLA`

14. Decrypt this quote about "Molly Weasley" which was enciphered using Hill's cipher: `CQFUM OEAZH YUMAW MYGCV GEQDD MKCEA BIKCU ZSMGN VUGC`

Affine Cipher Cell:

This SAGE cell can help you check your work when you encipher and decipher with a affine cipher, but you should be able to do the basic calculations your self.

```
@interact
def _(me=input_box('sage', label="Enter␣your␣message",
    height=3, width=50, type=str),m=[1..25], s=[0..25]):
    S = AffineCryptosystem(AlphabeticStrings())
    clean_text =
        S.encoding(str(me.encode('ascii','replace')))
    try:
        e = S(m,s)
        cipher_text = e(clean_text)
        print "Your␣affine␣enciphering␣text␣is\n"
        for i in xrange(0,len(cipher_text),5):
            print cipher_text[i:i+5],
            if (i+5)%50 == 0: print "\n"
    except:
        print "error,␣likely␣in␣your␣key"
        cipher_text = clean_text
```

Hill's Cipher Cell:

This SAGE cell can help you check your work when you encipher and decipher with a basic Hill's cipher, but you should be able to do the basic calculations your self.

```
@interact
def _(m=input_box('sage', label="Enter␣your␣message",
    height=3,
    width=50, type=str),
    key=input_grid(2,2,default=[[7,12],[3,3]],type=int,width=2)):
    S = HillCryptosystem(AlphabeticStrings(),2)
    M = MatrixSpace(IntegerModRing(26),2,2)
    e = S(M(key).transpose())
    clean_text = S.encoding(str(m.encode('ascii','replace')))
    if len(clean_text)%2 == 1:
        clean_text = S.encoding(str(m)+"Z")
    cipher_text = e(clean_text)
    print str("Your␣Hill␣enciphering␣text␣is\n")
    for i in xrange(0,len(cipher_text),5):
        print cipher_text[i:i+5],
        if (i+5)%50 == 0: print "\n"
```

Chapter 7

(*) The Modern Age

Objectives

- Symmetric ciphers (DES, AES) and the roll of group theory and linear algebra
- Asymmetric ciphers (Diffie-Hellman, RSA, El Gamal?, etc.) and the roll of number theory
- Absolute secrecy

7.1 Going Public

Objectives

- Biographies for Diffie and Hellman
- Public Key Cryptography Overview

7.2 Asymmetric Ciphers

Objectives

- Biographies for Rivest, Shamir, and Adelman
- A Little More Math
- RSA - Encryption
- Diffie-Hellman Key Exchange

7.3 Symmetric Ciphers

Objectives

- DES and Lucifer
- AES
- PGP

Appendix A

Sage Cell Analysis Utilities

Word Count Cell

```
import re
@interact
def word_count_simple(
                    text=input_box('Place␣the␣text␣that␣you␣
                        would␣like␣to␣analyze␣here!!!',
                    label="Enter␣your␣message", height=3,
                        width=50, type=str)):
    message = re.sub('[^A-Z]','␣
        ',str(text.encode('ascii','replace')).upper())+"␣"
    count = {}
    words = 0
    c=""
    for l in message:
        if l=="␣":
            if c in count:
                words += 1
                count[c] += 1
            elif c!="":
                count[c] = 1
                words += 1
            c=""
        else: c=c+l
    print "The␣total␣number␣of␣words␣in␣your␣message␣is␣
        {}".format(words)
    print "The␣number␣of␣distinct␣words␣in␣your␣message␣is␣
        {}".format(len(count))
    count = sorted([(value, key) for key, value in
        count.items()])
    print "Word\t\tCount\t\tPercent"
    count.reverse()
    for c in count[0:min(30,len(count))]:
        if len(c[1])>6:
            tab="\t"
        else:
            tab="\t\t"
        print c[1],tab,c[0],"\t\t␣%.4G" % (c[0]/words*100.0)
```

String Count Cell

```
import re
@interact
```

```
def string_count_simple(
                    text=input_box('Place␣the␣text␣that␣you␣
                        would␣like␣to␣analyze␣here!!!',
                    label="Enter␣your␣message", height=3,
                        width=50, type=str),
                    key=input_box('sage', label="String␣to␣
                        search", height=1, width=20,
                        type=str)):
    message =
        re.sub('[^A-Z]','',str(text.encode('ascii','replace')).upper())+"␣
        "
    count = 0
    K = re.sub('[^A-Z]','␣
        ',str(key.encode('ascii','replace')).upper())
    N = len(K)
    c=""
    for i in range(len(message)-N+1):
        c = message[i:i+N]
        if c == K: count += 1
    print "Your␣string␣appears␣",str(count),"␣times."
```

N-Gram Analysis Cell

```
@interact
def ngram_count_simple(
                    text=input_box('Place␣the␣text␣that␣you␣
                        would␣like␣to␣analyze␣here!!!',
                    label="Enter␣your␣message", height=3,
                        width=50, type=str),N=[1..6]):
    message =
        AlphabeticStrings().encoding(str(text.encode('ascii','replace')))
    count = {}
    for i in range(len(message)-N+1):
        c = message[i:i+N]
        if c in count: count[c] += 1
        else: count[c] = 1
    print "The␣length␣of␣your␣message␣is␣
        {}".format(len(message))
    count = sorted([(value, key) for key, value in
        count.items()])
    print "Rank\t\tN-Gram\t\tCount\t\tPercent"
    count.reverse()
    rank = 1
    for c in count[0:min(30,len(count))]:
        if len(c[1])>6:
            tab="\t"
        else:
            tab="\t\t"
        print str(str(rank)+"."),tab,c[1],tab,c[0],"\t\t␣
            %.4G" % (c[0]/len(message)*100.0)
        rank += 1
    if len(count)==26 and N==1: print "Pangram!!!"
```

Subset Analysis Cell

```
@interact
def subset_count(
                    text=input_box('Place␣the␣text␣that␣you␣
                        would␣like␣to␣analyze␣here!!!',
                    label="Enter␣your␣message", height=3,
                        width=50, type=str),
                    Subsets=[1..10]):
    message =
        AlphabeticStrings().encoding(str(text.encode('ascii','replace')))
    S = Subsets
    sets = [""]*S
    for i in range(0,len(message)):
        sets[i%S] += str(message[i])
    for i in range(S):
        tmp_message = sets[i]
        count = {}
        for i in range(len(tmp_message)):
            c = tmp_message[i:i+1]
            if c in count: count[c] += 1
            else: count[c] = 1
        print "The␣length␣of␣this␣subset␣is␣
            {}".format(len(tmp_message))
        count = sorted([(value, key) for key, value in
            count.items()])
        print "Character\tCount\t\tPercent"
        count.reverse()
        IC = 0
        for c in count[0:min(30,len(count))]:
            if len(c[1])>6:
                tab="\t"
            else:
                tab="\t\t"
            print c[1],tab,c[0],"\t\t␣%.4G" %
                (c[0]/len(tmp_message)*100.0)
            IC += c[0]*(c[0]-1.0)/len(tmp_message)^2
        print "And,␣the␣index␣of␣coincidence␣for␣this␣set␣is␣
            %.4G\n" %  (IC)
```

Appendix B

Sage Cell Cipher Utilities

Substitution Cipher Cell

```
import textwrap
@interact
def _(p =
    input_box('abcdefghijklmnopqrstuvwxyz',label='Plain',
    type=str,width=50,height=1),
            c =
                input_box('ZYXWVUTSRQPONMLKJIHGFEDCBA',label='Cipher',
                type=str,width=50,height=1),
            shift=[0..25],
            mode=selector(['encipher','decipher'],
                buttons=True),
            spaces = selector(['yes','no'], buttons=True),
            m=input_box('sage', label="Message", height=3,
                width=50, type=str)):
    P = str(p.encode('ascii','replace')).upper()
    C = str(c.encode('ascii','replace')).upper()
    C = C[shift:]+C[:shift]
    Message = str(m.encode('ascii','replace')).upper()
    print "\nYour␣Plain␣alphabet␣is:␣␣", str(P).lower()
    print "Your␣Cipher␣alphabet␣is:␣", str(C)
    if len(C)!=len(P):
        print "Key␣lengths␣do␣not␣match."
    else:
        if mode == 'encipher':
            inText = P
            outText = C
        else:
            inText = C.lower()
            outText = P.lower()
            Message = Message.lower()
        output = ""
        for char in Message:
            try:
                position = inText.index(char)
                output += outText[position]
            except:
                if spaces=='yes': output += char
                else: pass
        print "\nHere␣is␣your␣output:\n"
        if spaces == 'yes':
            print textwrap.fill(output, 50)
        else:
            for i in xrange(0,len(output),5):
                print output[i:i+5],
                if (i+5)%50 == 0: print "\n"
```

Affine Cipher Cell

```
@interact
def _(me=input_box('sage', label="Enter␣your␣message",
    height=3, width=50, type=str),m=[1..25], s=[0..25]):
    S = AffineCryptosystem(AlphabeticStrings())
    clean_text =
        S.encoding(str(me.encode('ascii','replace')))
    try:
        e = S(m,s)
        cipher_text = e(clean_text)
        print "Your␣affine␣enciphering␣text␣is\n"
        for i in xrange(0,len(cipher_text),5):
            print cipher_text[i:i+5],
            if (i+5)%50 == 0: print "\n"
    except:
        print "error,␣likely␣in␣your␣key"
        cipher_text = clean_text
```

Vigenere Cipher Cell

```
import re
import textwrap
@interact
def _(m=input_box('sage', label="Enter␣your␣message",
    height=3, width=50, type=str),
        key=input_box('sage', label="Enter␣your␣key",
            height=1, width=20, type=str),
        mode = selector(['encipher','decipher'],
            buttons=True),
        spaces = selector(['yes','no'], buttons=True)):
    plain_alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    clean_message = str(m.encode('ascii','replace')).upper()
    if spaces == 'no':
        clean_message = re.sub('[^A-Z]','',clean_message)
    cipher_key =
        re.sub('[^A-Z]','',str(key.encode('ascii','replace')).upper())
    key_list = [plain_alpha.index(ch) for ch in cipher_key]
    if mode == 'decipher':
        key_list = [-1*k for k in key_list]
    cipher_text = ""
    key_counter = 0
    for ch in clean_message:
        try:
            tmp_pos = plain_alpha.index(ch)
            cipher_pos = (tmp_pos+key_list[key_counter])%26
            cipher_text += plain_alpha[cipher_pos]
            key_counter = (key_counter+1)%len(key_list)
        except:
            if spaces == 'yes':
                cipher_text += ch
    print "\nHere␣is␣your␣output:\n"
    if spaces == 'yes':
        print textwrap.fill(cipher_text, 42)
    else:
        for i in xrange(0,len(cipher_text),6):
            print cipher_text[i:i+6],
            if (i+6)%42 == 0: print "\n"
```

Falconer Cipher Cell

```
import textwrap
import re
@interact
def falconer(message=input_box("The␣quick␣brown␣fox␣jumps␣
    over␣the␣lazy␣sleeping␣dog.",
                                  label="Message:",
                                     type=str, width=50,
                                     height=3),
             keys=input_grid(1,6,default=["CBA", "CAB",
                 "ACB", "BCA","BAC",""],
                    label="Keys:", to_value=list, type=str),
            chars=[3..5]):
    text =
        re.sub('[^A-Z]','',str(message.encode('ascii','replace')).upper())
    columns = "ABCDE"
    key = keys[0]
    while "" in key: key.remove("")
    message_table = [["" for x in range(chars)] for y in
        range(len(key))]
    for i in xrange(0,len(text),chars):
        row = (i/chars)%len(key)
        for j in range(chars):
            try:
                col = columns.index(key[row][j])
            except:
                col = chars-1 #pass
            try:
                message_table[row][col] += str(text[i+j])
            except:
                pass
    out_message = ""
    print "Chracters␣in␣text:␣",len(text)
    print "Cipher␣Table:"
    for k in range(len(key)):
        print
            "\t",str(key[k][0:chars]),":\t","\t".join(message_table[k])
        for i in range(chars):
            out_message += str(message_table[k][i])+"␣"
    print "Completed␣Message:"
    #for i in xrange(0,len(out_message),50):
    #    print
        "\t",out_message[i:min(i+50,len(out_message))].strip()
    print textwrap.fill(out_message, 50)
```

Hill Cipher Cell

```
@interact
def _(m=input_box('sage', label="Enter␣your␣message",
    height=3,
    width=50, type=str),
    key=input_grid(2,2,default=[[7,12],[3,3]],type=int,width=2)):
    S = HillCryptosystem(AlphabeticStrings(),2)
    M = MatrixSpace(IntegerModRing(26),2,2)
    e = S(M(key).transpose())
    clean_text = S.encoding(str(m.encode('ascii','replace')))
    if len(clean_text)%2 == 1:
        clean_text = S.encoding(str(m)+"Z")
    cipher_text = e(clean_text)
    print str("Your␣Hill␣enciphering␣text␣is\n")
    for i in xrange(0,len(cipher_text),5):
        print cipher_text[i:i+5],
        if (i+5)%50 == 0: print "\n"
```

Appendix C

Blank Tables, Charts, and Images

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

Table C.0.1: Monoalphabetic Substitution Table

1	2	3	4	5	6	7	8	9
a	b	c	d	e	f	g	h	i
10	20	30	40	50	60	70	80	90
j	k	l	m	n	o	p	q	r
100	200	300	400	500	600	700	800	900
s	t	u	v	w	x	y	z	

Table C.0.2: A Numerical Alphabet

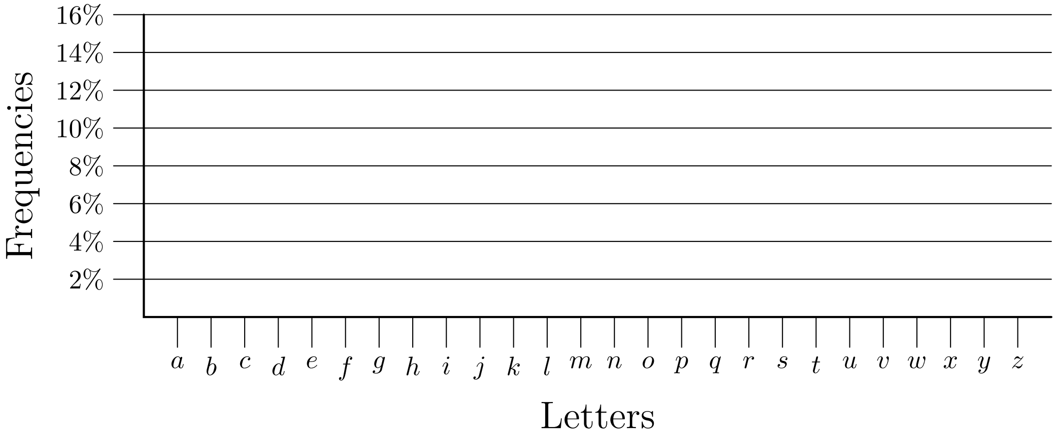


Figure C.0.3: Axes for Mapping Letter Frequencies

	Letter Freq.
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	

Figure C.0.4: Blank Tables for Frequency Counts

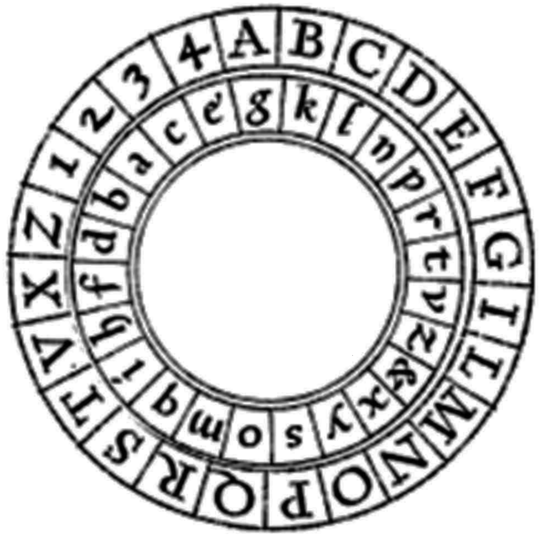

A B C D E F G I L M N O P Q R S T V X Z 1 2 3 4
g k l n p r t v z & x y s o m q i h f d b a c e

Figure C.0.5: Alberti's Cipher Disk or "Formula"

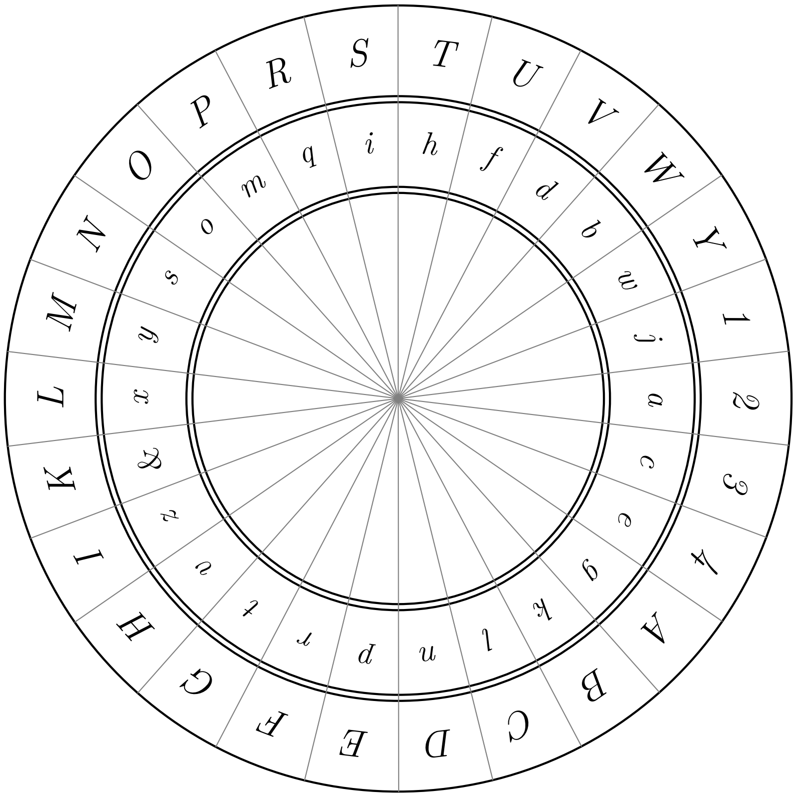
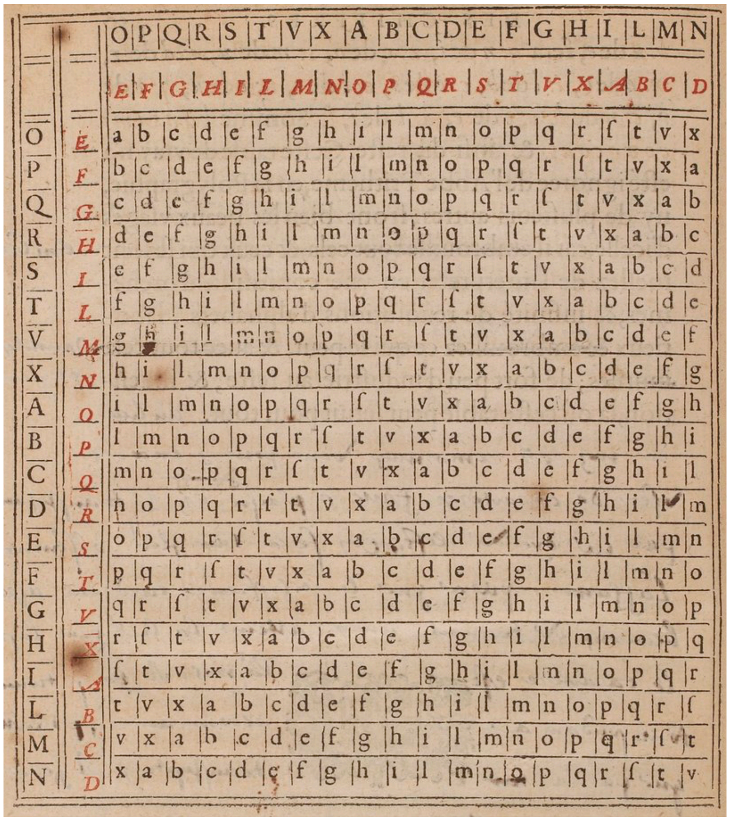

T U V W Y 1 2 3 4 A B C D E F G H I K L M N O P R S
h f d b w j a c e g k l n p r t v z & x y s o m q i

Figure C.0.6: Modern Cipher Disk or "Formula"

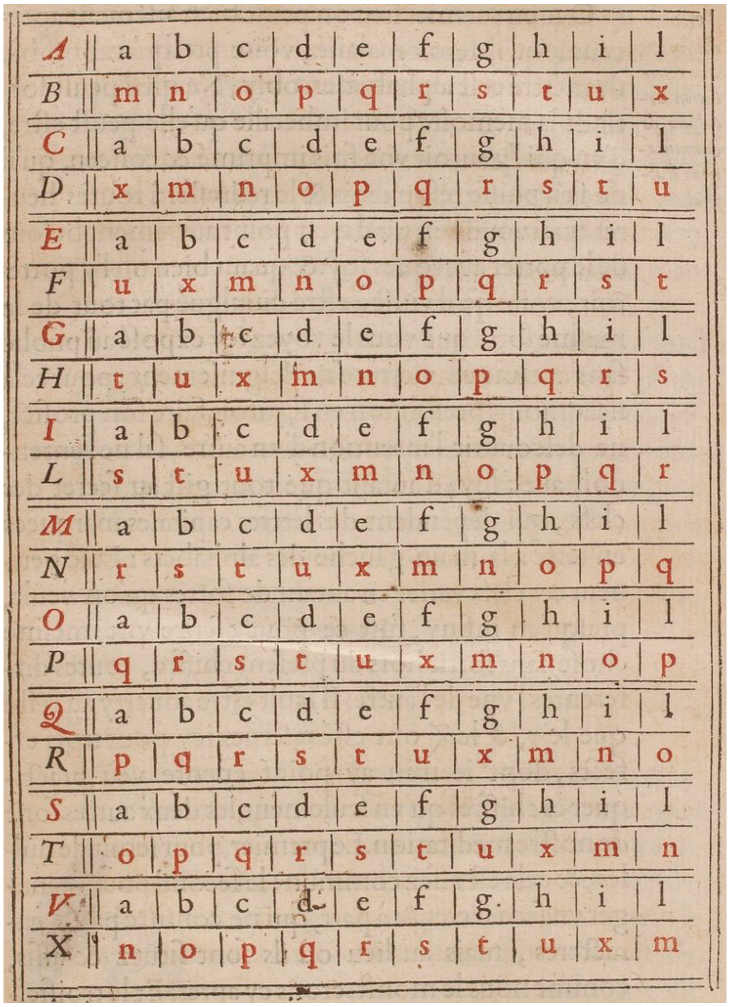


		O	P	Q	R	S	T	V	X	A	B	C	D	E	F	G	H	I	L	M	N
		E	F	G	H	I	L	M	N	O	P	Q	R	S	T	V	X	A	B	C	D
O	E	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x
P	F	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a
Q	G	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b
R	H	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c
S	I	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d
T	L	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e
V	M	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f
X	N	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g
A	O	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h
B	P	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i
C	Q	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l
D	R	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m
E	S	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n
F	T	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o
G	V	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p
H	X	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q
I	A	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r
L	B	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ
M	C	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t
N	D	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v

Figure C.0.7: Vigenère's Tableau

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

Figure C.0.8: Updated Vigenère Tableau



A	a	b	c	d	e	f	g	h	i	l
B	m	n	o	p	q	r	s	t	u	x
C	a	b	c	d	e	f	g	h	i	l
D	x	m	n	o	p	q	r	s	t	u
E	a	b	c	d	e	f	g	h	i	l
F	u	x	m	n	o	p	q	r	s	t
G	a	b	c	d	e	f	g	h	i	l
H	t	u	x	m	n	o	p	q	r	s
I	a	b	c	d	e	f	g	h	i	l
L	s	t	u	x	m	n	o	p	q	r
M	a	b	c	d	e	f	g	h	i	l
N	r	s	t	u	x	m	n	o	p	q
O	a	b	c	d	e	f	g	h	i	l
P	q	r	s	t	u	x	m	n	o	p
Q	a	b	c	d	e	f	g	h	i	l
R	p	q	r	s	t	u	x	m	n	o
S	a	b	c	d	e	f	g	h	i	l
T	o	p	q	r	s	t	u	x	m	n
V	a	b	c	d	e	f	g	h	i	l
X	n	o	p	q	r	s	t	u	x	m

Figure C.0.9: Vigenère's Autokey Tableau

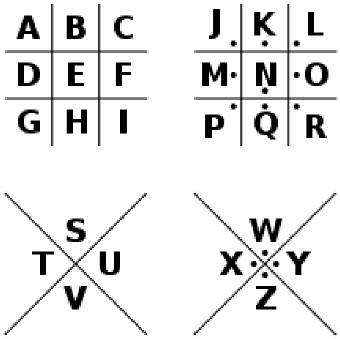


Figure C.0.10: Pigpen Cipher Key

		A	B	C
1				
2				
3				
4				
5				
6				

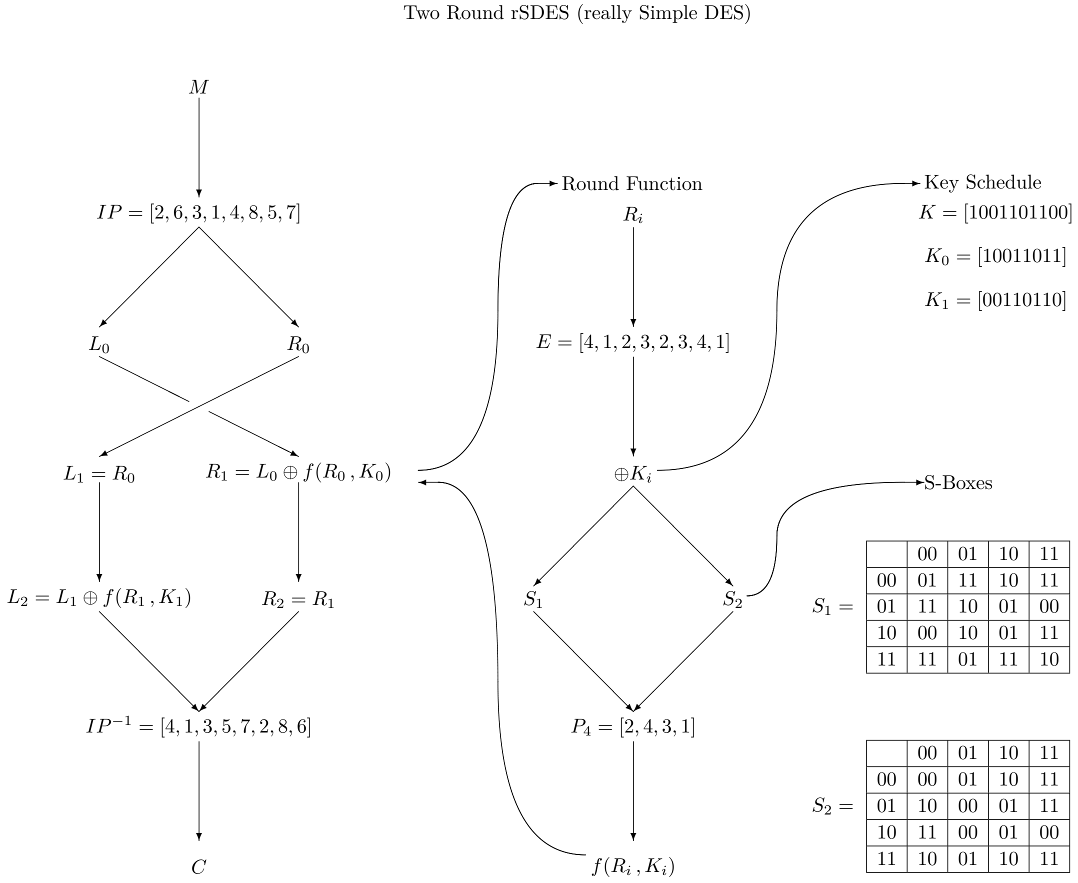
Table C.0.11: Falconer's Transposition Table

		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
12.22%	e	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
9.28%	t	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
8.06%	a	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
7.62%	o	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
7.10%	i	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
6.82%	n	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
6.45%	s	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
5.91%	r	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
5.76%	h	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
4.19%	l	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
3.93%	d	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V

Figure C.0.12: Kasiski Key Table

	$\times_{26}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
C	2	0	2	4	6	8	10	12	14	16	18	20	22	24	0	2	4	6	8	10	12	14	16	18	20	22	24
D	3	0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23
E	4	0	4	8	12	16	20	24	2	6	10	14	18	22	0	4	8	12	16	20	24	2	6	10	14	18	22
F	5	0	5	10	15	20	25	4	9	14	19	24	3	8	13	18	23	2	7	12	17	22	1	6	11	16	21
G	6	0	6	12	18	24	4	10	16	22	2	8	14	20	0	6	12	18	24	4	10	16	22	2	8	14	20
H	7	0	7	14	21	2	9	16	23	4	11	18	25	6	13	20	1	8	15	22	3	10	17	24	5	12	19
I	8	0	8	16	24	6	14	22	4	12	20	2	10	18	0	8	16	24	6	14	22	4	12	20	2	10	18
J	9	0	9	18	1	10	19	2	11	20	3	12	21	4	13	22	5	14	23	6	15	24	7	16	25	8	17
K	10	0	10	20	4	14	24	8	18	2	12	22	6	16	0	10	20	4	14	24	8	18	2	12	22	6	16
L	11	0	11	22	7	18	3	14	25	10	21	6	17	2	13	24	9	20	5	16	1	12	23	8	19	4	15
M	12	0	12	24	10	22	8	20	6	18	4	16	2	14	0	12	24	10	22	8	20	6	18	4	16	2	14
N	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13
O	14	0	14	2	16	4	18	6	20	8	22	10	24	12	0	14	2	16	4	18	6	20	8	22	10	24	12
P	15	0	15	4	19	8	23	12	1	16	5	20	9	24	13	2	17	6	21	10	25	14	3	18	7	22	11
Q	16	0	16	6	22	12	2	18	8	24	14	4	20	10	0	16	6	22	12	2	18	8	24	14	4	20	10
R	17	0	17	8	25	16	7	24	15	6	23	14	5	22	13	4	21	12	3	20	11	2	19	10	1	18	9
S	18	0	18	10	2	20	12	4	22	14	6	24	16	8	0	18	10	2	20	12	4	22	14	6	24	16	8
T	19	0	19	12	5	24	17	10	3	22	15	8	1	20	13	6	25	18	11	4	23	16	9	2	21	14	7
U	20	0	20	14	8	2	22	16	10	4	24	18	12	6	0	20	14	8	2	22	16	10	4	24	18	12	6
V	21	0	21	16	11	6	1	22	17	12	7	2	23	18	13	8	3	24	19	14	9	4	25	20	15	10	5
W	22	0	22	18	14	10	6	2	24	20	16	12	8	4	0	22	18	14	10	6	2	24	20	16	12	8	4
X	23	0	23	20	17	14	11	8	5	2	25	22	19	16	13	10	7	4	1	24	21	18	15	12	9	6	3
Y	24	0	24	22	20	18	16	14	12	10	8	6	4	2	0	24	22	20	18	16	14	12	10	8	6	4	2
Z	25	0	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Figure C.0.13: Multiplication table modulo 26


Two Round rSDES (really Simple DES)
M
$IP = [2, 6, 3, 1, 4, 8, 5, 7]$
L_0
R_0
$L_1 = R_0$
$R_1 = L_0 \oplus f(R_0\,, K_0)$
$L_2 = L_1 \oplus f(R_1\,, K_1)$
$R_2 = R_1$
$IP^{-1} = [4, 1, 3, 5, 7, 2, 8, 6]$
C
Round Function
R_i
$E = [4, 1, 2, 3, 2, 3, 4, 1]$
$\oplus K_i$
S_1
S_2
$P_4 = [2, 4, 3, 1]$
$f(R_i\,, K_i)$
Key Schedule
$K = [1001101100]$
$K_0 = [10011011]$
$K_1 = [00110110]$
S-Boxes

$S_1 =$

	00	01	10	11
00	01	11	10	11
01	11	10	01	00
10	00	10	01	11
11	11	01	11	10

$S_2 =$

	00	01	10	11
00	00	01	10	11
01	10	00	01	11
10	11	00	01	00
11	10	01	10	11

Figure C.0.14: A Highly Simplified two step DES

Appendix D

Quotations Appendix

D.1 Chapter 1 Quotes:

Julius Caesar Shift:

> "There are extant likewise some letters from him to Cicero and others to his friends concerning his domestic affairs in which if there was occasion for secrecy he wrote in cyphers, that is he used the alphabet in such a manner that not a single ward could be made out. The way to decipher those epistles was to substitute the fourth for the first letter as d for a and so for the other letters respectively." [12, p. 37]

Augustus Caesar Shift:

> "When he had occasion to write in cypher he put b for a, c for b [and] so forth and instead of z, aa." [12, p. 134]

Falconer on 26!:

> *Schottus* demonstrates, (though the calculation in his book be not exact) that a thousand million of men in as many years could not write down all those different transpositions of the alphabet, granting every one should complete forty pages a day, and every page contain forty several positions: For if one writer in one day write forty pages, everyone containing forty combinations, 40 multiplied by 40, gives 1,600, the number he completes in one day, which multiplied by 366, the number (and more) of days in a year; a writer in one year shall compass 585,600 distinct rows. Therefore in a thousand million of years he could write
>
> $$585,600,000,000,000,$$
>
> which being multiplied by 1,000,000,000, the number of writers supposed, the product will be
>
> $$585,600,000,000,000,000,000,000,$$

which wants of the number of combinations no less than

$$348,484,017,332,394,393,600,000.$$

[4, pages 5-6]

Ibn ad-Durayhim on Numerical Ciphers:

”***5. On the replacement of letters using the decimally-weighted numerical alphabet:***

- By substituting decimal numerical alphabet for letters in four different ways: by writing the numbers in words as pronounced; or by finger-bending, using the fingers to communicate the message visually to a recipient; or by writing the numbers as numerals such as writing *(mhmd: forty, eight, forty, four)*; or by giving the cryptogram a semblance of a page of a financial register.
- By recovering the cryptogram numeral into a number of letters - a method of encipherment which involves more sophistication. There are many combinations that can be used in this method; for example in *(mhmd: jl, fb, jl, ca)* or *(kk, ga, kk, bb)* . One can even form delusive words such as *(mhmd: lead, cad, deal, baa)*, or substitute two words for a letter, e.g. *(ali:* $\overline{dig\ fad}$*,* $\overline{cab\ ab}$*)*, in which case a line is to be drawn over two words to denote that they represent one letter.
- By multiplying the number representing the letter by two, and so write *(mhmd: q, jf, q, h)* and *(ali: ob, jh)*, etc; or multiply it by three, thus writing *(mhmd: sk, kd, sk, jb)* and *(ali: rc, kg)*. Numbers can also be multiplied by four or five.” [a] [9, vol. 3, pp. 69-70]

[a] The examples here are very loosely based on the Arabic examples in the translation. The ”mhmd” is Mohamed since Arabic is written without vowels, and for ”ali” the a and l together are treated as a single letter.

D.2 Chapter 2 Quotes:

Al-Kindi on Frequency Analysis:

”***Algorithms of Cryptanalysis***

So we say, the enciphered letters are either in numerical proportions, that is poetry -because poetic meter, ipso facto, sets measures to the number of letters in each line-, or they are not. Non- poetry can be cryptanalyzed using either quantitative or qualitative expedients.

The ***quantitative*** expedients include determining the most frequently occurring letters in the language in which cryptograms are to be cryptanalyzed. If vowels functioned as the material from which any language is made, and non-vowels functioned as the shape of any language, and since many shapes can be made from the same material, then the number of vowels in any language would be greater than non-vowels. For instance, gold is the material of many shapes of finery and vessels; it may cover crowns, bangles, cups, etc.. The gold in these realizations is more than the shapes made of it. Similarly, the vowels which are the material of any kind of text are more than the non- vowels in any language. I mean by vowels the letters: (a), (y or i or e) and (o or u). Therefore the vowels in any language, inevitably, exceed in number the non-vowels in a text of that language. It happens that in certain languages some vowels are greater in number than some other vowels, while non-vowels may be frequent or scarce according to their usage in each language, such as the letter (s), of which frequency of occurrence is high in Latin.

Among the expedients we use in cryptanalyzing a cryptogram if the language is already known, is to acquire a fairly long plaintext in that language, and count the number of each of its letters. We mark the most frequent letter ”first”, the second most frequent ”second”, and the following one ”third”, and so forth until we have covered all its letters. Then we go back to the message we want to cryptanalyze, and classify the different symbols, searching for the most frequent symbol of the cryptogram and we regard it as being the same letter we have marked ”first” -in the plaintext-; then we go to the second frequent letter and consider it as being the same letter we have termed ”second”, and the following one ”third”, and so on until we exhaust all the symbols used in this cryptogram sought for cryptanalysis.

It could happen sometimes that short cryptograms are encountered, too short to contain all the symbols of the alphabet, and where the order of letter frequency cannot be applied. Indeed the order of letter frequency can normally be applied in long texts, where the scarcity of letters in one part of the text is compensated for by their abundance in another part.

Consequently, if the cryptogram was short, then the correlation between the order of letter frequency in it and in that of the language would no longer be reliable, and thereupon you should use another, ***qualitative*** expedient in cryptanalyzing the letters. It is to detect in the language in which cryptograms are enciphered the associable letters and the dissociable ones. When you discern two of them using the letter order of frequency, you see whether they are associable in that language. If so, you seek each of them elsewhere in the cryptogram, comparing it with the preceding and following dissociable letters by educing from the order of frequency of letters, so as to see whether they are combinable or non-combinable. If you find that all these letters are combinable with that letter, you look for letters combinable with the second letter. If found really combinable, so they are the expected letters suggested by the combination and non-combination of letters, and also by their order of frequency. Those expected letters are correlated with words that make sense. The same procedure is repeated elsewhere in the ciphertext until the whole message is cryptanalyzed." [9, vol. 1, pp. 121-123]

Falconer's Analysis Step 1-4:

First, Distinguish the Vowels from the Consonants.

1. And first, the vowels generally discover themselves by their frequency; for because they are but few in number, and no word made up without some of them, they must frequently be used in any writing.
2. Where you find any character or letter standing by it self, it must be a vowel.
3. If you find any character doubled in the beginning of a word, in any language it is a vowel, as *Aaron, Eel, Jilt, Oogala, Vulture, etc.*, except for some English proper names, as Llandaff or Lloyd.
4. In monosyllables of two letters you may distinguish it from the consonant joined with it by its frequency.
5. In a word of three letters beginning and ending in the same letter the vowel is probably included.
6. When you find a character doubled in the middle of a word of four letters, 'tis probably the vowel *e* or *o*.
7. In Polysyllables, where a character is double in the middle of the word, it is for the most part a consonant; and if so, the precedent letter is always a vowel, and very often the following.[4, pp. 8-9]

Secondly, Distinguish the Vowels from Themselves..

1. Compare their frequency, and *e*, as we observed before, is generally the most used in the English tongue, next *o*, then *a* and *i*; but *u* and *y* are not so frequently used as some of the consonants.
2. It is remarkable that amongst the vowels, *e* and *o* are often doubled, the rest seldom or never.
3. *e* is very often a terminal letter, and *y* terminates words, but they are distinguishable, because there is no proportion to their frequency: *o* is not often in the end of words, except in monosyllables.

4. *e* is the only vowel that can be doubled in the end of an English word, except *o* in *too*, etc.
5. You may consider which of the vowels, in any language, can stand alone, as *a*, *i*, and sometimes *o* in English, *a*, *e*, *o*, in Latin or *i* the imperative of *eo*.[4, pp. 9-10]

Distinguish the Consonants Amongst Themselves..

1. As before observe their frequency. Those of most use in English are *d, h, n, r, s, t*, and next to those may be reckoned *c, f, g, l, m, w*, in third rank may be placed *b, k, p*, and lastly *q, x, z...*
2. You may consider which consonants may be doubled in the middle or end of words.
3. What are terminal letters, etc.
4. The number and nature of consonants and vowels that fall together, or do usually fall together.[4, p. 10]

Additional Observations.

1. A word of three letters, beginning and ending with the same, may be supposed *did*
2. A word consisting of four characters, with the same letter in the beginning and end, is probably *that* or *hath*
3. A word consisting of five letters, when the second and last are the same, is commonly *which*, though it may be otherways, as in *known*, *serve*, etc. And you may judge of the truth of such suppositions by the frequency of the letters in the word supposed.

Next you may compare words one with another, as *on* and *no*, each being the other reversed; so *of* and *for*, the last being the first reversed with the addition of a letter; for and from will discover each other, etc.

You may also likewise observe some of the usual propositions and terminations of words, such as *com, con, ing, ed*, etc. Note that *t* and *h* are often joined in the beginning and end of English words, and sometimes in the middle.[4, pp. 11-12]

D.3 Chapter 3 Quotes:

Alberti's Polyalphabetic Cipher:

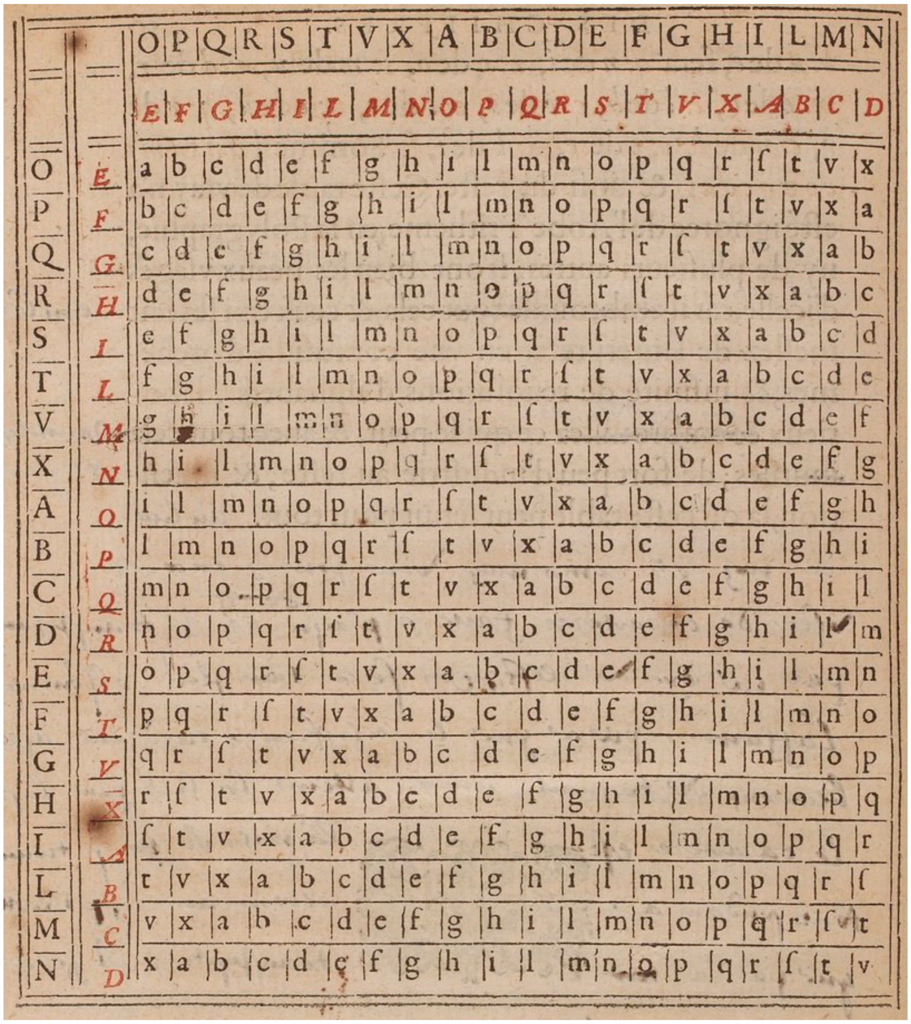
"Say for example we mutually establish k as the index of the mobile circle. Writing, the formulae are positioned at will, say such k lies under the uppercase B [Figure 3.1.2] and the next letter corresponds to the letter that comes next. In writing to you, I will first put the uppercase B under which lies the index k in the formula; this is to signal you far away, wanting to read what I have written, that you should set up the twin formula in your keeping, positioning the mobile circle so that the B sits over the index k. Then all of the rest of the lowercase letters present in the coded text will take their meaning and sound from those of the fixed circle above them."

"After I have written three or four words I will mutate the position of the index in our formula, rotating the disk let's say, so that the index k falls below the upper case R. Then in this missive I write an uppercase R to indicate that k no longer refers to B, but to R, and the letters that will follow will assume new meanings." - Alberti [14, p. 181]

Vigenère's Polyalphabetic Cipher:

"... set the capitals which run across the top [of the table] for the message to be conveyed & those that run perpendicularly down the left for the keys. I have put two rows of capitals here, one black and the other red, to show that alphabets of the text as, the keys, may be transposed and changed in as many ways as you want to keep knowledge of them from all others except ones correspondents. Therefore, to encipher the saying we used previously *"au nom de l'eternel,"* with the key *"le jour obscur"* proceed in this way; from *a* in the alphabet across the top in red, come to the row with *l* and the box with *b*: u from e will be q: n from i, n: o from o, s: m from u, a: d from r, m: e from o, i: l from b, c: e from s, o: t from c, n: e from u, q: r from r, c: n from l, o: e from e, a: l from i, l. Altogether we get *bqnsamiconcoal* & so on for the remainder." [13, pp. 49b, 50, 50b (100-102)], [8, p. 110][a]

[a] Most of this translation is due to Mendelsohn [8], I have only filled in with my own at the end which he had not translated.



		O	P	Q	R	S	T	V	X	A	B	C	D	E	F	G	H	I	L	M	N
		E	F	G	H	I	L	M	N	O	P	Q	R	S	T	V	X	A	B	C	D
O	E	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x
P	F	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a
Q	G	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b
R	H	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c
S	I	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d
T	L	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e
V	M	g	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f
X	N	h	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g
A	O	i	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h
B	P	l	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i
C	Q	m	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l
D	R	n	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m
E	S	o	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n
F	T	p	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o
G	V	q	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p
H	X	r	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q
I	A	ſ	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r
L	B	t	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ
M	C	v	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t
N	D	x	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	ſ	t	v

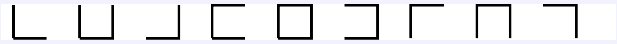
Figure D.3.1: Vigenère's Tableau

Agripa's Pigpen Cipher

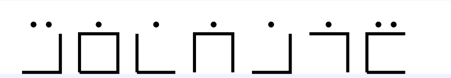
One may frame nine chambers by the intersection of four parallel lines intersectiong themselves at right angles as expressed in the figure:

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

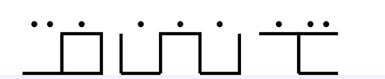
Which being dissected into parts generates nine the particular figures:



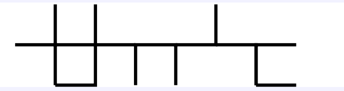
from the nine chambers. Characterize each letter in a chamber by the notation of one point to show the first letter in the chamber; two for the second letter; three for the third letter; so that the characters of Michael may be written in seven characters:



Which may be written one after another drawn as three figures:



Which written one after another drawn as one figure, omitting the usual marks, gives the characters of Michael as:

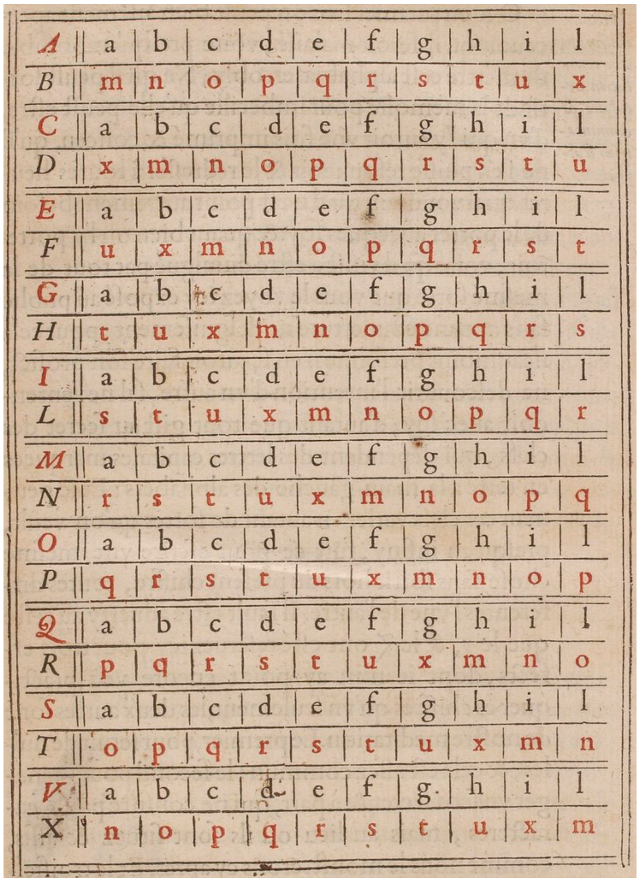


[1, Vol. 3, CCLXXV (p.279)] [2, Vol.3, Chapter XXX] [a]

[a] The translation is based, for the most part, on [2] with reference to the original text. The example was changed some in order to make the work more accessible.'

Vigenère's Autokey Cipher

"Each letter may be enciphered by the preceding letter, thus: with the . . . text. . . . *"Au nom de l'eternel"* and the key D, we say, a [the first letter of the text] from D [the key letter] gives x; u [the second letter of the text] from A [the first letter of the text, which now becomes the key] gives i; n from U, a; o from N, h; m from O, g; d from M, u; e from D, p; l from E, t; e from L; m, t from E, l; e from T, s; r from E, h; n from R, i; e from N, x; l from E, t. By which method, taking D for the key, we would arrive at *DXIAHGUPTMLSHIXT*. The other method, which is more secret, is to encipher each letter of the clear text not by the letter [of that text] which precedes it, but by the letter by which the preceding letter is enciphered. Thus, a from D, which is the key, gives x; u from X [the first letter of the cipher text] h; n from H, e; o from E, e; m from E, c; d from C, o; e from O, u; l from U, m; ... etc." [13, pp. 49-49b] [8, p. 128]



A	a	b	c	d	e	f	g	h	i	l
B	m	n	o	p	q	r	s	t	u	x
C	a	b	c	d	e	f	g	h	i	l
D	x	m	n	o	p	q	r	s	t	u
E	a	b	c	d	e	f	g	h	i	l
F	u	x	m	n	o	p	q	r	s	t
G	a	b	c	d	e	f	g	h	i	l
H	t	u	x	m	n	o	p	q	r	s
I	a	b	c	d	e	f	g	h	i	l
L	s	t	u	x	m	n	o	p	q	r
M	a	b	c	d	e	f	g	h	i	l
N	r	s	t	u	x	m	n	o	p	q
O	a	b	c	d	e	f	g	h	i	l
P	q	r	s	t	u	x	m	n	o	p
Q	a	b	c	d	e	f	g	h	i	l
R	p	q	r	s	t	u	x	m	n	o
S	a	b	c	d	e	f	g	h	i	l
T	o	p	q	r	s	t	u	x	m	n
V	a	b	c	d	e	f	g	h	i	l
X	n	o	p	q	r	s	t	u	x	m

Figure D.3.2: Table used for Vigenère's Autokey Cipher

D.4 Chapter 6 Quotes:

Hill's Cryptography in an Algebraic Alphabet:

Cryptography in an Algebraic Alphabet.

By Lester S. Hill, Hunter College

1. *The Bi-Operational Alphabet*

Let $a_0,\ a_1,\ \ldots,\ a_{25}$ denote any permutation of the letters of the English alphabet; and let us associate the letter a_i with the integer i. We define operations of *modular addition* and *multiplication* (modulo 26) over the alphabet as follows:

$$a_i + a_j = a_r,$$
$$a_i\, a_j = a_t,$$

where r is the remainder obtained upon dividing the integer $i + j$ by the integer 26 and t is the reamainder obtained on dividing ij by 26. The integers i and j may be the same or different.

It is easy to verify the following salient propositions concerning the bi-operational alphabet thus set up:

(1) If $\alpha,\ \beta,\ \gamma$ are letters of the alphabet,

- $\alpha+\beta=\beta+\alpha$ and $\alpha\beta=\beta\alpha$ [commutative law]
- $\alpha+(\beta+\gamma)=(\alpha+\beta)+\gamma$ and $\alpha(\beta\gamma)=(\alpha\beta)\gamma$ [associative law]
- $\alpha(\beta+\gamma)=\alpha\beta+\alpha\gamma$ [distributive law]

(2) There is exactly one “zero” letter, namely a_0, characterized by the fact that the equation $\alpha+a_0=\alpha$ is satisfied whatever the letter denoted by *alpha*.

(3) Given any letter α, we can find exactly one letter β, dependent on α, such that $\alpha+\beta=a_0$. We call β the “negative” of α, and we write: $\beta=-\alpha$.

(4) Given any letters α, β we can find exactly on letter γ such that $\alpha+\gamma=\beta$ [i.e. $\gamma=\beta-\alpha$ is unique].

(5) Distinguishing the twelve letters,

$$a_1,\ a_3,\ a_5,\ a_7,\ a_9,\ a_{11},\ a_{15},\ a_{17},\ a_{19},\ a_{21},\ a_{23},\ a_{25},$$

with subscripts prime to 26, as “primary” letters, we make the assertion, easily proved: If α is any primary letter and β is any letter, there is exactly one letter γ for which $\alpha\gamma=\beta$.

(6) In any algebraic sum of terms, we may clearly omit terms of which the letter a_0 is a factor; and we need not write the letter a_1 explicitly as a factor in any product.

2. An Illustration

Let the letters of the alphabet be associated with the integers as follows:

a	b	c	d	e	f	g	h	i	j	k	l	m
5	23	2	20	10	1	8	4	18	25	0	16	13
n	o	p	q	r	s	t	u	v	w	x	y	z
7	3	1	19	6	12	24	21	17	14	22	11	9

or, in another convenient formulation:

0	1	2	3	4	5	6	7	8	9	10	11	12
k	p	c	o	h	a	r	n	g	z	e	y	s
13	14	15	16	17	18	19	20	21	22	23	24	25
m	w	f	l	v	i	q	d	u	x	b	t	j

It will be seen that

$$c+x=t,\ j+w=m,\ f+y=k,\ -f=y,\ -y=f,\ etc.$$
$$an=z,\ hm=k,\ cr=s,\ etc.$$

The zero letter is k, and the unit letter is p. The primary letters are: $a\ b\ f\ j\ n\ o\ p\ q\ u\ v\ y\ z$.

Since this particular alphabet will be used several times, in illustration of further developments, we append the following table of negatives and reciprocals:

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

The solution to the equation $z + \alpha = t$ is $\alpha = t - z$ or $\alpha = t + (-z) = t + v = f$.

The system of linear equations: $o\,\alpha + u\,\beta = x$, $n\,\alpha + i\,\beta = q$ has solution $\alpha = u$, $\beta = o$, which may be obtained by the familiar method of elimination or by formula. [5, pp.306-308]

References

[1] Heinrich Cornelius Agrippa (1486?-1535), *De occulta philosophia.* (Coloniae, 1533).
URL: https://lccn.loc.gov/20007812

[2] Heinrich Cornelius Agrippa (1486?-1535) [John French trans.], *Three books of occult philosophy written by Henry Cornelius Agrippa of Nettesheim.* Text Creation Partnership, Ann Arbor, MI / Oxford (UK), (2012)
URL: http://name.umdl.umich.edu/A26565.0001.001, accessed June 2018.

[3] Arthur Conan Doyle, *A study in scarlet.* Project Gutenberg, (2016)
URL: http://www.gutenberg.org/files/244/244-0.txt

[4] John Falconer, *Cryptomenysis patefacta; or, the art of secret information disclosed without a key. containing plain and demonstrative rule, for decyphering.* Daniel Brown, London, (1685).
URL: https://archive.org/details/cryptomenysispat00falc

[5] Lester S. Hill, *Cryptography in an algebraic alphabet.* The American Mathematical Monthly, Vol. 36, No. 6 (Jun. - Jul., 1929), pp. 306-312.
Stable URL: http://www.jstor.org/stable/2298294

[6] David Kahn, *The codebreakers: the story of secret writing.* Scribner, New York, (1996).

[7] Friedrich W. Kasiski [R.W. Pettengill trans.], *Secret writing and the art of deciphering with special consideration of the german and french languages.* Privately Published, Washington D.C., (1954). (Original work published Berlin 1863)
Thanks to Rene Stein at the National Cryptologic Museum for assistance in obtaining a copy of this work.

[8] Charles J. Mendelsohn, *Blaise de Vigenère and the "Chiffre Carré".* Proceedings of the American Philosophical Society, Vol. 82, No. 2 (Mar. 22, 1940), pp. 103-129.
Stable URL: http://www.jstor.org/stable/985011

[9] M. Mrayati, Y. Meer Alam, and M.H. at Tayyan (eds.), *Series on arabic origins of cryptology in six volumes.* KFCRIS and KACST, Riyadh, (2003).

[10] Charles Rocca, *"Falconer's cryptology",* in Research in History and Philosophy of Mathematics, The CSHPM 2014 Annual Meeting in St. Catharines, Ontario, edited by Maria Zack and Elaine Landry, pp.1-13. Birkhäuser, Basel, (2015).

[11] Simon Singh, *The code book: the science of secrecy from ancient egypt to quantum cryptography.* Anchor Books, New York, (2000).

[12] Suetonius, *The lives of the twelve caesars.* Bohn's Classical Library, G.

Bell, (1901.)
URL: http://books.google.com/books?id=pjcbAAAAYAAJ

[13] Blaise de Vigenère, *Traicté des chiffress, ou secrètes manières d'escrire,* Chez Abel L'Angelier, Paris, (1586).
URL: http://gallica.bnf.fr/ark:/12148/bpt6k1040608n/f7.image

[14] K. Williams, L. March, and S. Wassell (eds.), *The mathematical works of leon battista alberti* Birkhäuser, Basel (2010).

Appendix E

Solutions to Selected Exercises

1 Caesar's Shifty Idea

1.1 Simple Early Ciphers (and a little math)

Checkpoint 1.1.3 Answer. `KHOOR ZRUOG`

Checkpoint 1.1.4 Answer. this is a test

Checkpoint 1.1.5 Answer. zombies like zebras

Checkpoint 1.1.6 Answer. Here is a longer message for you to practice on. The shift for this message was by a factor of seven, so the a goes to the h. This doesn't really offer much more security, especially since there are only twenty six possible shift ciphers and one of those doesn't change anything. While you were deciphering this what patterns did you notice? What words stood out? How could we make this a more secure (harder to crack) cipher? These are all questions that we will investigate in more detail throughout the text.

Checkpoint 1.1.7 Answer.

> "Be copy now to men of grosser blood, And teach them how to war. And you, good yeoman, Whose limbs were made in England, show us here The mettle of your pasture; let us swear That you are worth your breeding; which I doubt not; For there is none of you so mean and base, That hath not noble lustre in your eyes. I see you stand like greyhounds in the slips, Straining upon the start. The game's afoot: Follow your spirit, and upon this charge Cry 'God for Harry, England, and Saint George!" - Henry V, William Shakespeare

Checkpoint 1.1.8 Answer.

> "We few, we happy few, we band of brothers; For he to-day that sheds his blood with me Shall be my brother; be he ne'er so vile, This day shall gentle his condition: And gentlemen in England now a-bed Shall think themselves accursed they were not here, And hold their manhoods cheap whiles any speaks That fought with us upon Saint Crispin's day." - Henry V, by William Shakespeare

Checkpoint 1.1.9 Answer. 40,329,146,112,660,563,558,400,000 seconds, 11,202,540,586,850,156,544,000 hours, or 1,275,334,766,262,540,590 and 10/61 years

Checkpoint 1.1.11 Answer.

- 4!=24
- 7!=5040
- 10!=3628800

Checkpoint 1.1.12 Answer. “Before you begin a thing, remind yourself that difficulties and delays quite impossible to foresee are ahead. If you could see them clearly, naturally you could do a great deal to get rid of them but you can’t. You can only see one thing clearly and that is your goal. Form a mental vision of that and cling to it through thick and thin.” - Kathleen Norris”

1.2 Arabic Numerical Ciphers

Checkpoint 1.2.2 Answer.

1. NINE TWO FIVE SIXHUNDRED
2. HA AA CB VT
3. KG DB JE ZXV

Checkpoint 1.2.3 Answer. There are multiple possible encipherments, a couple are `V B S` and another is `QUK AA JJOK`.

Checkpoint 1.2.4 Answer. “When I want to understand what is happening today or try to decide what will happen tomorrow, I look back.” - Omar Khayyam

Checkpoint 1.2.5 Answer. One possible solution is LEAD A SOM CAD.

How Shifty are You?

1.1. **Answer**.

```
``FN'EN JUU QNJAM CQJC J VRUURXW VXWTNHB KJWPRWP XW
J VRUURXW CHYNFARCNAB FRUU NENWCDJUUH ANYAXMDLN
CQN NWCRAN FXATB XO BQJTNBYNJAN. WXF, CQJWTB CX
CQN RWCNAWNC, FN TWXF CQRB RB WXC CADN.'' -
YAXONBBXA AXKNAC BRUNWBTH
```

1.2. **Answer**. There are in fact multiple answers to this, one of them would be:

```
STUVVWXYZ TRJ ST SN U LE ME VS VX
```

1.3. **Answer**. BAA LEEK BAD

1.4. **Answer**.

```
NRHGZPVH ZIV Z KZIG LU YVRMT SFNZM. ZKKIVXRZGV
BLFI NRHGZPVH ULI DSZG GSVB ZIV: KIVXRLFH ORUV
OVHHLMH GSZG XZM LMOB YV OVZIMVW GSV SZIW DZB.
FMOVHH RG?H Z UZGZO NRHGZPV, DSRXS, ZG OVZHG,
LGSVIH XZM OVZIM UILN. - ZO UIZMPVM, LS, GSV
GSRMTH R PMLD, 2002
```

1.5. **Answer**.

> “Now they show you how detergents take out bloodstains, a pretty violent image there. I think if you’ve got a T-shirt with a bloodstain all over it, maybe laundry isn’t your biggest problem. Maybe you should get rid of the body before you do the wash.” - Jerry Seinfeld

1.6. **Answer**.

> “One should guard against preaching to young people success in the customary form as the main aim in life. The most important motive for work in school and in life is pleasure in work, pleasure in its result, and the knowledge of the value of the result to the community.” - Albert Einstein

1.7. **Answer**.

> “In a completely rational society, the best of us would be teachers and the rest of us would have to settle for something less, because passing civilization along from one generation to the next ought to be the highest honor and the highest responsibility anyone could have.” - Lee Iacocca

1.8. **Answer**.

> “The good Christian should beware of mathematicians and all those who make empty prophecies. The danger already exists that mathematicians have made a covenant with the devil to darken the spirit and confine man in the bonds of Hell.” - St. Augustine

1.9. **Answer**.

```
MPKRGOPL AHRM GHP ZHFR NKHF TDGGDGO. VHQK
MPKQOOERM ARSREHI VHQK MPKRGOPLM. TLRG VHQ OH
PLKHQOL LXKAMLDIM XGA ARZDAR GHP PH MQKKRGARK,
PLXP DM MPKRGOPL. - XKGHEA MZLTXKWRGROORK
```

2 Attacking the Alphabet

Bringing it all Together

2.5. **Answer**. “of all that is good, sublimity is supreme. succeeding is the coming together of all that is beautiful. furtherance is the agreement of all that is just. perseverance is the foundation of all actions.” - lao tzu

Solution. When you look at the frequencies you see that `G` and `V` are most common, next most common are `R` and `Z`. So we know that one of the first two is probably `e` while the other is `t`, and one of the latter two is likely `a`, assuming the frequencies are fairly normal.

Then we see that `GSV` and `GSZG` appear frequently, and `GS` is the most common bi-gram (so that it is likely `th`). Putting this together we see that `e` is `V`, `t` is `G`, `a` is `Z`, and `h` is `S`. So that `GSV` and `GSZG` represent `the` and `that`.

We can also see that `ZOO` appears four times with `Z` replaced by `a` this looks like `aOO`, the likely candidate is that `ZOO` is `all`, so that `O` is `l`.

At this point we can start writing down the letters we have in the monoalphabetic substitution table (Table C.0.1). When you do that you will hopefully notice that the cipher letters are in reverse alphabetic order, so this was likely enciphered with atabash which we learned about in Exercise 1.4.

2.6. **Answer**. "far better is it to dare mighty things, to win glorious triumphs, even though checkered by failure ... than to rank with those poor spirits who neither enjoy nor suffer much, because they live in a gray twilight that knows not victory nor defeat." - theodore roosevelt

Solution. Looking at a basic frequency analysis we see that the most common single letters are G, R, B, E, V, U, A, F, N, H, the most common bi-gram is GU, and the most common tri-gram is GUR; from this we may conclude that G is t, U is h, and R is e.

Since the word spacing is preserved we also have the two letter words VF, VG, GB, and VA, and the one letter word N, these allow us to deduce the ciphertext-plaintext pairs N - a, B - o, V - i, and F - s.

When we start writing down the ciphertext we have worked out under the plaintext alphabet (use Table C.0.1) we can see that the letters we have discovered are in the correct order and with the correct spacing so that we appear to have a shift, trying this as a possible solution we see that we are correct and that a was shifted to N.

2.7. **Answer**. "... what is out of the common is usually a guide rather than a hindrance. In solving a problem of this sort, the grand thing is to be able to reason backwards. That is a very useful accomplishment, and a very easy one, but people do not practice it much. In the every-day affairs of life it is more useful to reason forwards, and so the other comes to be neglected. There are fifty who can reason synthetically for one who can reason analytically." - Sherlock Holmes in Study in Scarlet by Sir Arthur Conan Doyle

Solution. As always you need to start with a basic frequency analysis. From this you immediately get that the ten most common letters are L, T, J, S, Q, P, I, K, F, D, the most common bi-gram and tri-gram are TK and TKL, L appears frequently before and after other letters (in particular P), and finally TKLPL and TKLDP appear multiple times; from all of this we get the ciphertext-plaintext pairs L - e, T - t, H - k, P - r, and D - i.

Then we can start filling in bits and pieces of the message and looking for more clues, for example LVLPY becomes eVerY which we reasonably assume is every. Also, with the hint that it is a keyword cipher we can line up the ciphertext letters we have worked out underneath a copy of the plaintext alphabet (Table C.0.1) and look for patterns. Continuing in this way we can finally arrive at the solution and the key is KEYWORD: SHERLOCKED, key letter: a.

3 Mixing Things Up

3.1 Alberti's Great Idea

3.1.2 Alberti's Polyalphabetic Cipher

Checkpoint 3.1.3 Answer. G vbb ke V xsdba m&bbry

Checkpoint 3.1.4 Answer. cats and dogs

3.1.3 Vigenère's Cipher

Checkpoint 3.1.7 Answer. Ciphertext: QOTEASLQACVI

Checkpoint 3.1.8 Answer. Ciphertext: `IIREHXSL`

Checkpoint 3.1.9 Answer. Plaintext: *tour de france*

Checkpoint 3.1.10 Answer. Plaintext: *baguette*

3.2 Variation on a Theme

3.2.2 Pigpen Cipher

Checkpoint 3.2.3 Answer. Here are the first few words of the message:

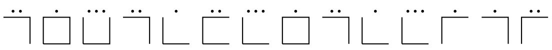


Figure E.0.1: Cipher Text:

Checkpoint 3.2.5 Answer. Plaintext: “Pork ’n Beans”

Checkpoint 3.2.8 Answer. “I am fond of pigs. Dogs look up to us. Cats look down on us. Pigs treat us as equals.” Winston S. Churchill

Checkpoint 3.2.10 Answer.

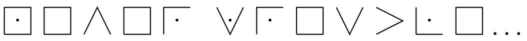


Figure E.0.2: Cipher Text

3.3 An Automatic Hit

3.3.2 The Autokey Cipher

Checkpoint 3.3.2 Answer. Method 1:

```
Plaintext:    a u t o m a t i c c o n f u s i o n
Key Letter:   Q A U T O M A T I C C O N F U S I O
Ciphertext: Q P I G A G R H M U N D H M A F M G H
```

Method 2:

```
Plaintext:    a u t o m a t i c c o n f u s i o n
Key Letter:   Q P E L G D X G R N T A B R F I Q L
Ciphertext: Q P E L G D X G R N T A B R F I Q L F
```

Checkpoint 3.3.3 Answer.

```
Ciphertext: C H F A B H C X N X M G F M R R M D D M R
Key Letter:   C S T O R M T H E B A S T I L L E N U L
Plaintext:    s t o r m t h e b a s t i l l e n u l l
```

Checkpoint 3.3.4 Answer.

```
Ciphertext: Q M P T M G P O H G S O O H L Q X U G C
Key Letter:   Q M P T M G P O H G S O O H L Q X U G
Plaintext:    h i d i n g i n p l a i n s i g h t x
```

6 Mathematics to the Rescue

6.1 Affine Ciphers

6.1.3 Affine Cipher

Checkpoint 6.1.17 Answer. `ZKTTI WIXTF`

Checkpoint 6.1.18 Answer. `cryptology rocks`

Checkpoint 6.1.19 Answer. `MTSZC MTTSZ CUSHB CP`

Checkpoint 6.1.20 Answer. `more than a shift`

Checkpoint 6.1.21 Answer. The message begins with "One summer night, a few months after my ..."

6.2 Hill's Cipher

6.2.1 Matrices

Checkpoint 6.2.2 Answer.

$$\begin{pmatrix} 4 \\ 14 \end{pmatrix} \pmod{26}$$

Checkpoint 6.2.3 Answer.

$$\begin{pmatrix} 12 & 10 \\ 18 & 9 \end{pmatrix} \pmod{26}$$

Checkpoint 6.2.4 Answer.

$$\begin{pmatrix} 22 \\ 9 \end{pmatrix} \pmod{26}$$

Checkpoint 6.2.5 Answer.

$$\begin{pmatrix} 15 \\ 3 \end{pmatrix} \pmod{26}$$

Checkpoint 6.2.6 Answer.

$$\begin{pmatrix} 13 \\ 0 \end{pmatrix} \pmod{26}$$

6.2.2 Hill's Cipher

6.2.2.1 Enciphering with Matrices

Checkpoint 6.2.7 Answer. `(ph)`≡`HO` and `(er)`≡`YL`

Checkpoint 6.2.8 Answer. Your enciphered message should be something like `JJNPA PYWTM IMKN`.

Checkpoint 6.2.9 Answer. `CKEADO`

6.2.2.2 Matrix Inverses and Deciphering

Checkpoint 6.2.12 Answer.

$$m^{-1} = \begin{pmatrix} 20 & 19 \\ 21 & 23 \end{pmatrix}.$$

Checkpoint 6.2.13 Answer. `great job`

Checkpoint 6.2.14 Answer. `nice work`

Checkpoint 6.2.15 Answer. `hello again`

Up Hill struggle?

6.9. Answer. `VIXJZ FVIBW DUZT`

6.11. Answer. `spicy chicken wings`

6.13. Answer. “Chuck Norris threw a grenade and killed 50 zombies, then it exploded.”

Appendix F

Glossary

Block Cipher A *block cipher* repeatedly uses a single key in order to encipher successive pieces of plaintext.

Cipher A *cipher* is a system by which a letter or block of letters in a message is replaced by another letter or block of letters in a systematic way.

Ciphertext The *ciphertext* is the message you wish to transmit after it has been rewritten in order to hide its meaning.

Code A *code* is a system by which words or phrses in a message are replaced by other words, phrases, or symbols in a systematic way.

Crib In cryptology a *crib* is a known piece of plaintext which may be compared to as sample ciphertext in order to aid cryptanalysis.

Cryptoanalysis *Cryptoanalysis* is an attempt to compute the key or determine the meaning of a message from the ciphertext.

Cryptography *Cryptography* focuses on concealing the meaning of a message.

Cryptology *Cryptology* is the science of maintaining secrets by either hiding the meaning or the existence of messages.

Decipher *Deciphering* is the process by which blocks of ciphertext are replaced by plaintext.

Decrypt *Decrypting* is the process by we discover the meaning of a ciphertext without the aid of a cipher key.

Encipher *Enciphering* is the process by which blocks of plaintext are replaced by ciphertext.

Factorial Given a counting number n we define n**-factorial** as the number of ways in which we can arrange n objects and we calculate it by taking the product of all the consecutive positive integers from n down to 1, and we write:

$$n! = n(n-1)(n-2)\cdots 2\cdot 1$$

Note that for a variety of reasons $0! = 1$.

Frequency Analysis *Frequency analysis* is the process of counting the characters, blocks of characters, or words in a text in order to determine how many of each there are relative to the entire length of the text.

Greatest Common Divisor The *greatest common divisor* of two integers is the largest positive integer which divides both. For a variety of reasons it is important to know that it is also defined as the least positive linear combination of the two integers, i.e. 5 is the $gcd(15, 35)$ because not larger positive integer divides both 15 and 35 and because $5 = 1 \cdot (35) - 2 \cdot (15)$ is the smallest possible positive combination of 15 and 35.

Homophonic Cipher A *homophonic cipher* is a cipher in which a single plaintext letter may be replaced by multiple different ciphertext letters throughout a message.

Index of Coincidence The *index of coincidence* for a block of text is the probability that two characters chosen at random are the same.

Modular Equivalence Given a positive integer n two integers are *equivalent modulo* n if they have the same remainder when you divide them by n. Equivalently, two integers a and b are equivalent modulo n if $a - b$ is divisible by n. When this is the case we write $a \equiv b \pmod{n}$

Monoalphabetic Substitution Cipher A *monoalphabetic substitution cipher* is a cipher in which there is a one-to-one correspondence between plaintext letters and the cipher text so that a single plaintext letter is always replaced by the same letter, group of letters, or symbol and every letter, group of letters, or symbol in the ciphertext represents the same plaintext letter.

Multiplicative Identity A number a is the *multiplicative identity* if for all numbers b,

$$a \cdot b = b \cdot a = b.$$

Multiplicative Inverse See Reciprocal

N-Gram An *n-gram* is a block of n consecutive characters.

Nomenclator A *nomenclator* is a system for disguising the meaning of a message which uses both cipher alphabets and a set of code words.

Pangrams *Pangrams* are words or sentences containing every letter of the alphabet at least once.

Plaintext The *plaintext* is the message you wish to transmit written in a readable form.

Polyalphabetic Substitution Cipher A *polyalphabetic substitution cipher* is a cipher in which a single plaintext letter maybe replaced by several differnt ciphertext letters, groups of letters, or symbols and every letter, group of letters, or symbol in the ciphertext may represent more than one plain text letter.

Polygraphic Cipher A *polygraphic cipher* is a cipher in which multiple plaintext characters are enciphered at a time so that how a character is enciphered depends on which other letters it is beside.

Polyphonic Cipher A *polyphonic cipher* is a cipher in which a single ciphertext letter may be represent multiple different plaintext letters throughout a message.

Prime Number A positive integer is *prime* if it has precisely two divisors, one and its self.

Reciprocal The *reciprocal* (or *multiplicative inverse*) of a number x is some number y such that $x \cdot y = 1$.

Relatively Prime Numbers Two integers are *relatively prime* if there greatest common divisor is one. In this case we may say that one of the integers is *prime* to the other.

Steganography *Steganography* focuses on concealing the existence of a message.

Stream Cipher A *stream cipher* uses a sequence (or stream) of different keys in order to encipher successive pieces of plaintext.

Substitution Cipher A *substitution cipher* changes individual characters or groups of characters, but does not change their position.

Transposition Cipher A *transposition cipher* rearranges the positions of characters, but does not change the characters themselves.

Index

Appendix G

GNU Free Documentation License

0. PREAMBLE The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties — for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with… Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

This book was authored in MathBook XML.