Cryptanalysis

Lecture 1: Computing in the Presence of an Adversary

John Manferdelli
jmanfer@microsoft.com
JohnManferdelli@hotmail.com

jlm20080923

Welcome to Cryptanalysis

Class Mechanics

- Web site is best comprehensive information source.
- Microsoft e-mail is most reliable way to reach me.
- Grading: 25% Final, 75% Homework.
- Sign up for mailing list, Wiki.
- Office: 444 CSE.

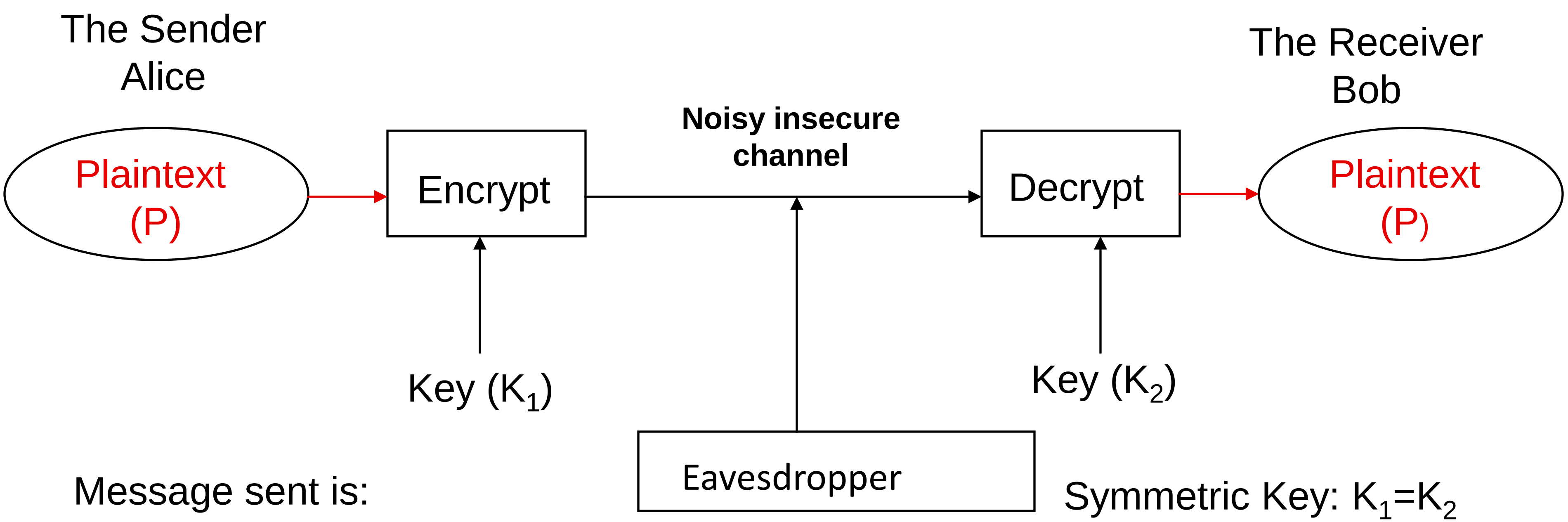
Web Site: http://www.cs.washington.edu/education/courses/599r/08au/

Prerequisites

- Check out description of class and “Short Math Notes.”

Basic Definitions

The wiretap channel: “In the beginning”



Message sent is:

$C= E_{K1}(P)$

Decrypted as:

$P=D_{K2}(C)$

P is called plaintext.

C is called ciphertext.

Symmetric Key: $K_1=K_2$

Public Key: $K_1 \neq K_2$

K_1 is publicly known

K_2 is Bob’s secret

Cryptography and adversaries

- Cryptography is computing in the presence of an adversary.
- An adversary is characterized by:
  - Talent
    - Nation state: assume infinite intelligence.
    - Wealthy, unscrupulous criminal: not much less.
  - Access to information
    - Probable plaintext attacks.
    - Known plaintext/ciphertext attacks.
    - Chosen plaintext attacks.
    - Adaptive interactive chosen plaintext attacks (oracle model).
  - Computational resources
    - Exponential time/memory.
    - Polynomial time/memory .

Computational strength of adversary (edging towards high class version)

- Infinite - Perfect Security
  - Information Theoretic
  - Doesn’t depend on computing resources or time available
- Polynomial
  - Asymptotic measure of computing power
  - Indicative but not dispositive
- Realistic
  - The actual computing resources under known or suspected attacks.
  - This is us, low brow.

Information strength of the adversary (high class version)

- Chosen Plaintext Attack (CPA, offline attack)
  - The adversary can only encrypt messages
- Non-adaptive Chosen Ciphertext Attack (CCA1)
  - The adversary has access to a decryption oracle until, but not after, it is given the target ciphertext
- Adaptive Chosen Ciphertext Attack (CCA2)
  - The adversary has unlimited access to a decryption oracle, *except that the oracle rejects the target ciphertext*
  - The CCA2 model is very general – in practice, adversaries are much weaker than a full-strength CCA2 adversary
  - Yet, many adversaries are too strong to fit into CCA1

Your role

- In real life, you usually protect the user (COMSEC, now IA)
- Here, you're the adversary (COMINT, now SIGINT)
  - Helps you be a smarter for the COMSEC job.
  - You may as well enjoy it, it's fun.
  - Don't go over to the Dark side, Luke.
- In real life, it's important to have ethical people do both jobs

Dramatis persona

Users

- Alice (party A)
- Bob (party B)
- Trent (trusted authority)
- Peggy and Victor (authentication participants)

Users Agents

- Cryptographic designer
- Personnel Security
- Security Guards
- Security Analysts

Adversaries

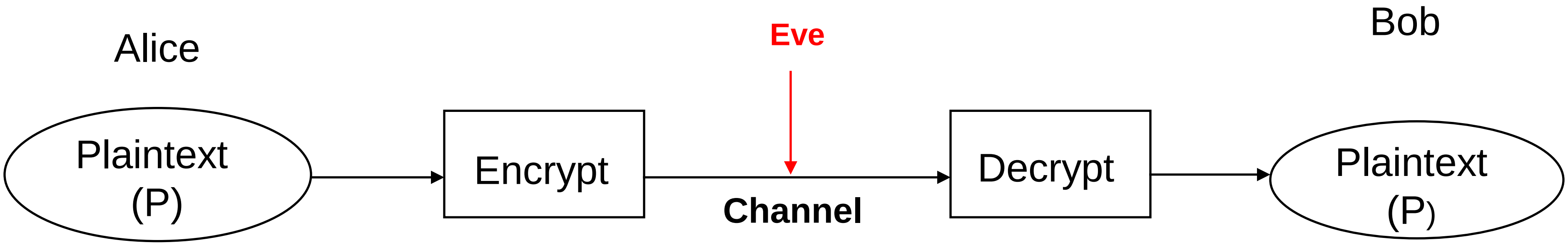
- Eve (passive eavesdropper)
- Mallory (active interceptor)
- Fred (forger)
- Daffy (disruptor)
- Mother Nature
- Users (Yes Brutus, the fault lies in us, not the stars)

Adversaries Agents

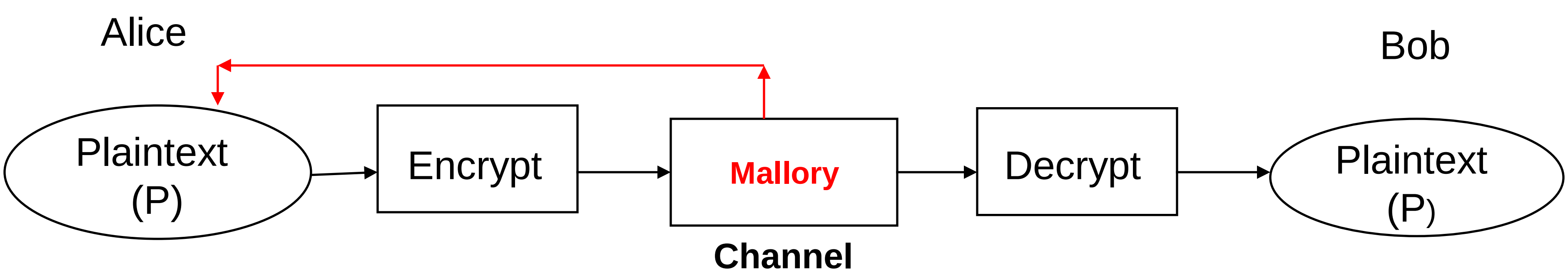
- Dopey (dim attacker)
- Einstein (smart attacker --- you)
- Rockefeller (rich attacker)
- Klaus (inside spy)

Adversaries and their discontents

Wiretap Adversary (Eve)



Man in the Middle Adversary (Mallory)



It’s not just about communications privacy

Users want:

- Privacy/Confidentiality
- Integrity
- Authentication
- Non-repudiation
- Quality of Service

Adversaries want to:

- Read a message
- Get key, read all messages
- Corrupt a message
- Impersonate
- Repudiate
- Deny or inhibit of service

Remember

Who’s the customer? What do they need? What’s the risk?

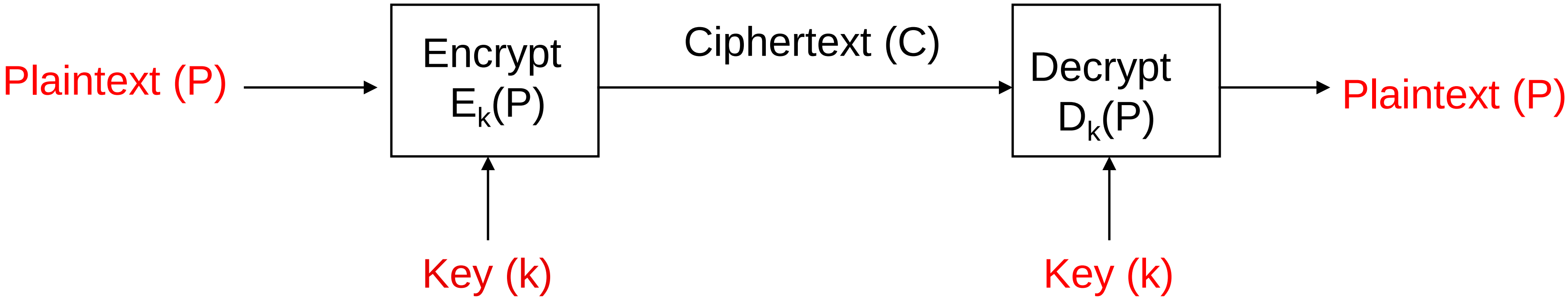
Public policy? Role of standardization and interoperability.

It’s the system, stupid: practices and procedures.

Cryptographic toolchest

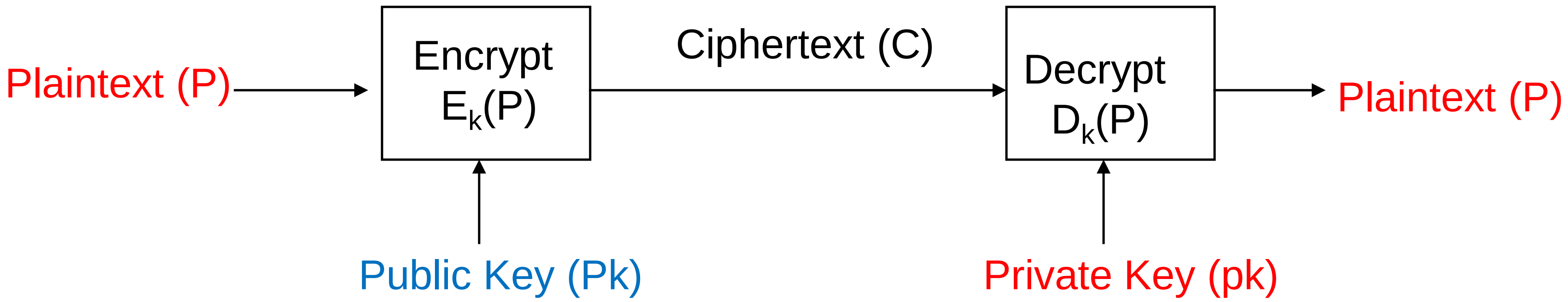
- Symmetric ciphers (includes classical ciphers)
  - Block ciphers
  - Stream ciphers
  - Codes
- Asymmetric ciphers (Public Key)
- Cryptographic Hashes
- Entropy and random numbers
- Protocols and key management

Symmetric ciphers



- Encryption and Decryption use the same key.
  - The transformations are simple and fast enough for practical implementation and use.
  - Two major types: Stream ciphers and block ciphers.
  - Examples: DES, AES, RC4, A5, Enigma, SIGABA, etc.
  - Can’t be used for key distribution or authentication.

Asymmetric (Public Key) ciphers



Encryption and Decryption use different keys.

- Pk is called the public key and pk is the private key. Knowledge of Pk is sufficient to encrypt. Given Pk and C, it is infeasible to compute pk and infeasible to compute P from C.
- Invented in mid 70's –Hellman, Merkle, Rivest, Shamir, Adleman, Ellis, Cocks, Williamson
- Public Key systems used to distribute keys, sign documents. Used in https:. Much slower than symmetric schemes.

Cryptographic hashes, random numbers

- Cryptographic hashes ($h:\{0,1\}^* \longrightarrow \{0,1\}^{bs}$. bs is the output block size in bits--- 160, 256, 512 are common)
  - One way: Given b=h(a), it is hard (infeasible) to find a.
  - Collision Resistant: Given b=h(a), it is hard to find a’ □a such that h(a’)= b.
- Cryptographic random numbers
  - Not predictable even with knowledge of source design
  - Passing standard statistical tests is a necessary but not sufficient condition for cryptographic randomness.
  - Require “high-entropy” source.
  - Huge weakness in real cryptosystems.
- Pseudorandom number generators
  - Stretch random strings into longer strings
  - More next quarter

Algorithm Speed

Algorithm	**Speed**
RSA-1024 Encrypt	.32 ms/op (128B), 384 KB/sec
RSA-1024 Decrypt	10.32 ms/op (128B), 13 KB/sec
AES-128	.53 µs/op (16B), 30MB/sec
RC4	.016 µs/op (1B), 63 MB/sec
DES	.622 µs/op (8B), 12.87 MB/sec
SHA-1	48.46 MB/sec
SHA-256	24.75 MB/sec
SHA-512	8.25 MB/sec

Timings do not include setup. All results typical for a 850MHz x86.

What are Ciphers

A cipher is a tuple $<M, C, K_1, K_2, E(K_1,x), D(K_2,y)>$

- M is message space, x is in M.
- C is cipher space, y is in C.
- K_1 and K_2 are paired keys (sometimes equal).
- E is encryption function and K_1 is the encryption key.
- D is decryption function and K_2 is the decryption key.
- $E(K_1,x)= y$.
- $D(K_2,y)=x$.

Mechanisms for insuring message privacy

- Ciphers
- Codes
- Stegonography
  - Secret Writing (Bacon’s “Cipher”)
  - Watermarking

- We’ll focus on ciphers which are best suited for mechanization, safety and high throughput.

Codes and Code Books

- One Part Code
  - A 2
  - Able 8
- Two Part
  - In first book, two columns. First column contains words/letters in alphabetical order, second column has randomly ordered code groups
  - In second code book, columns are switched and ordered by code groups.
- Sometimes additive key is added (mod 10) to the output stream
- Code book based codes are “manual.” We will focus on ciphers from now on.
- “Codes” also refers to “error correcting” codes which are used to communicate reliably over “noisy” channels. This area is related to cryptography. See, MacWilliams and Sloane or van Lint.

Basic Ciphers

- Monoalphabetic Substitution
  - Shift
  - Mixed alphabet
- Transposition
- Polyalphabetic Substitution
  - Vigenere
- One Time Pad
- Linear Feedback Shift Register

Kerckhoffs' Principle

- The confidentiality required to insure practical communications security must reside solely in the knowledge of the key.
- Communications security cannot rely on secrecy of the algorithms or protocols
  - We must assume that the attacker knows the complete details of the cryptographic algorithm and implementation
- This principle is just as valid now as in the 1800's.

Cipher Requirements

- WW II
  - Universally available (simple, light instrumentation) – interoperability.
  - Compact, rugged: easy for people (soldiers) to use.
  - Security in key only: We assume that the attacker knows the complete details of the cryptographic algorithm and implementation
  - Adversary has access to some corresponding plain and ciphertext
- Now
  - Adversary has access to unlimited ciphertext and lots of chosen text.
  - Implementation in digital devices (power/speed) paramount.
  - Easy for computers to use.
  - Resistant to ridiculous amount of computing power.

Practical attacks

- Exhaustive search of theoretical key space.
- Exhaustive search of actual key space as restricted by poor practice.
- Exploiting bad key management or storage.
- Stealing keys.
- Exploiting encryption errors.
- Spoofing (ATM PIN).
- Leaking due to size, position, language choice, frequency, inter-symbol transitions, timing differences, side channels..

Paper and pencil ciphers --- “In the beginning”

Transposition

- A transposition rearranges the letters in a text.
- Example: Grilles
  - Plain-text: BULLWINKLE IS A DOPE
  - Written into a predefined rectangular array

```
B U L L
W I N K
L E I S   →  BWLAEUINEDLNIOLKSP
A D O P
E
```

$c_i = p_{S(i)}$ **where**

S=(1)(2,5,17,16,12,11,7,6)(3,9,14,4,13,15,8,10)

- Another example: Rail fence cipher.

Breaking filled columnar transposition

Message (from Sinkov)

EOEYE GTRNP SECEH HETYH SNGND DDDET OCRAE RAEMH TECSE USIAR WKDRI RNYAR ABUEY ICNTT CEIET US

Procedure

1. Determine rectangle dimensions (l,w) by noting that message length=m = l x w. Here m=77, so l=7, w=11 or l=11, w=7
2. Anagram to obtain relative column positions

Note a transposition is easy to spot since letter frequency is the same as regular English.

Anagramming

- Look for words, digraphs, etc.
- Note: Everything is very easy in corresponding lain/ciphertext attack

1	**2**	**3**	**4**	**5**	**6**	**7**
E	E	G	A	E	R	C
O	C	N	E	U	N	N
E	E	D	R	S	Y	T
Y	H	D	A	I	A	T
E	H	D	E	A	R	C
G	E	D	M	R	A	E
T	T	E	H	W	N	I
R	Y	T	T	K	U	E
N	H	O	E	D	E	T
P	S	C	C	R	Y	U
S	N	R	S	I	I	S

→

3	**6**	**1**	**5**	**7**	**2**	**4**
G	R	E	E	C	E	A
N	N	O	U	N	C	E
D	Y	E	S	T	E	R
D	A	Y	I	T	H	A
D	R	E	A	C	H	E
D	A	G	R	E	E	M
E	N	T	W	I	T	H
T	U	R	K	E	Y	T
O	E	N	D	T	H	E
C	Y	P	R	U	S	C
R	I	S	I	S	N	S

Alphabetic substitution

- A *mono-alphabetic* cipher maps each occurrence of a plaintext character to a cipher-text character (the same one every time).
- A *poly-alphabetic* cipher maps each occurrence of a plaintext character to more than one cipher-text character.
- A *poly-graphic* cipher maps more than one plain-text character at a time
  - Groups of plaintext characters are replaced by assigned groups of cipher-text characters

Et Tu Brute?: Substitutions

- Caeser Cipher (Shift)

  `Message: B U L L W I N K L E I S A D O P E`

  `Cipher:  D W N N Y K P M N G K U C F Q S G`

  c= pC^k, C= (ABCDEFGHIJKLMNOPQRSTUVWXYZ), k= 2 here

  k=3 for classical Caeser

- More generally, any permutation of alphabet

Attacks on substitution

- Letter Frequency

A	.0651738	B	.0124248	C	.0217339	D	.0349835
E	.1041442	F	.0197881	G	.0158610	H	.0492888
I	.0558094	J	.0009033	K	.0050529	L	.0331490
M	.0202124	N	.0564513	O	.0596302	P	.0137645
Q	.0008606	R	.0497563	S	.0515760	T	.0729357
U	.0225134	V	.0082903	W	.0171272	X	.0013692
Y	.0145984	Z	.0007836	sp	.1918182		

- Probable word.
- Corresponding plain/cipher text makes this trivial.

Inter symbol information

- Bigraphs

EN	RE	ER	NT	TH
ON	IN	TE	AN	OR
ST	ED	NE	VE	ES
ND	TO	SE	AT	TI

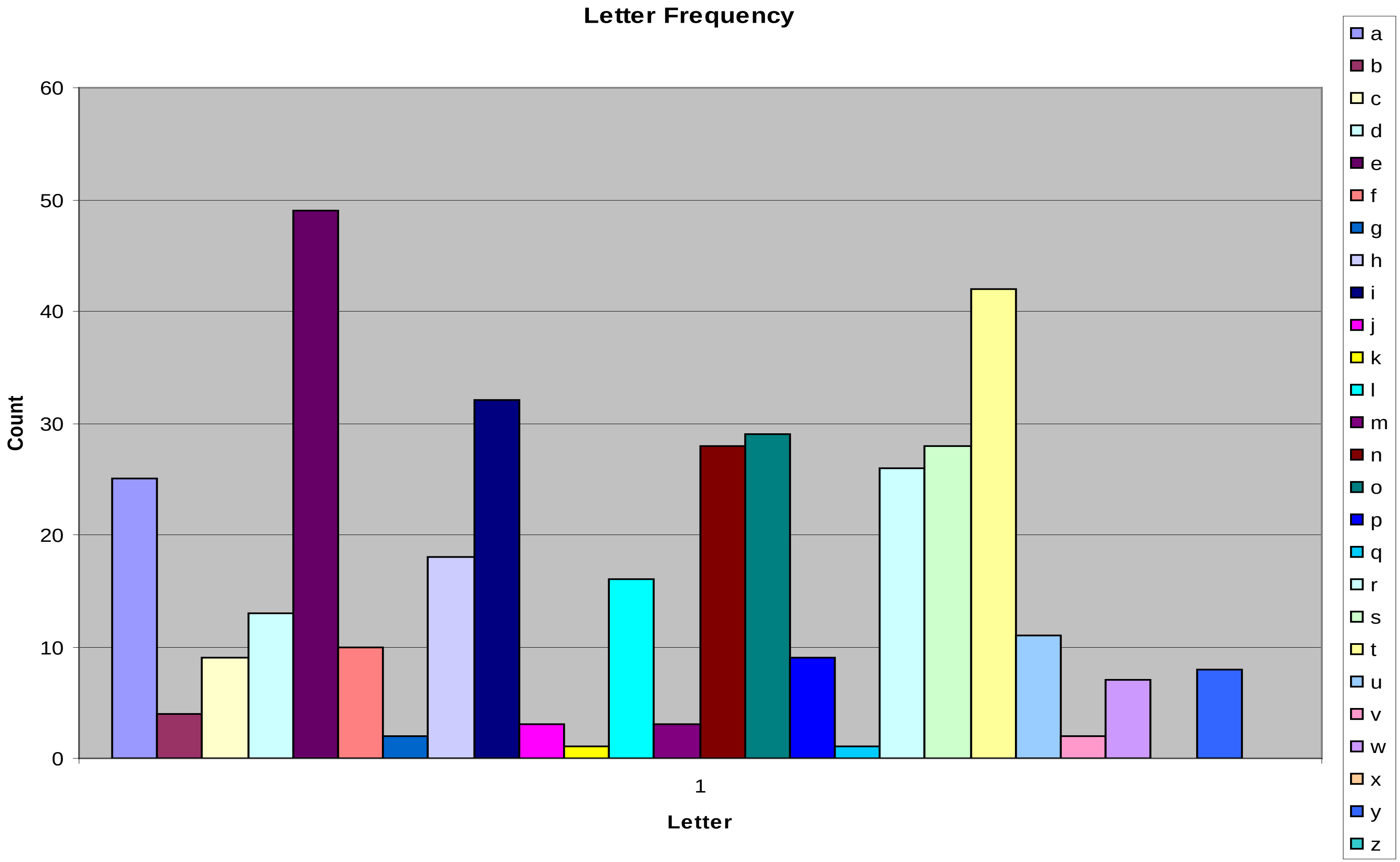
- Trigraphs

ENT	ION	AND	ING	IVE
TIO	FOR	OUR	THI	ONE

- Words

THE	OF	AND	TO	A
IN	THAT	IS	I	IT
FOR	AS	WITH	WAS	HIS
HE	BE	NOT	BY	BUT
HAVE	YOU	WHICH	ARE	ON

Letter frequency far graph



Breaking a mono-alphabet substitution

LB HOMVY QBF TFIL EOON LWO HFLLBY SDJVYM FNADPZI

Ch	#	Freq	Ch	#	Freq	Ch	#	Freq	Ch	#	Freq
L	5	0.125	F	4	0.100	O	4	0.100	B	3	0.075
Y	3	0.075	D	2	0.050	M	2	0.050	N	2	0.050
H	2	0.050	V	2	0.050	I	2	0.050	E	1	0.025
P	1	0.025	Q	1	0.025	S	1	0.025	T	1	0.025
A	1	0.025	W	1	0.025	J	1	0.025	Z	1	0.025

40 characters, index of coincidence: 0.044.

```
LB HOMVY QBF TFIL EOON LWO HFLLBY SDJVYM FNADPZI
to begin you must keep the button facing upwards
```

Breaking a mono-alphabet substitution

FMGWG OWG O XQJYGW UI YOEE YGOWLXPH LXHLRG FMG LHLH FMOF KOX YG MGOWR

Ch	#	Freq	Ch	#	Freq	Ch	#	Freq	Ch	#	Freq
G	9	0.161	O	7	0.125	L	5	0.089	W	5	0.089
M	4	0.071	H	4	0.071	F	4	0.071	X	4	0.071
Y	4	0.071	R	2	0.036	E	2	0.036	Q	1	0.018
I	1	0.018	U	1	0.018	J	1	0.018	K	1	0.018
P	1	0.018									

56 characters, index of coincidence: 0.071.

```
FMGWG OWG O XQJYGW UI YOEE YGOWLXPH LXHLRG FMG
there are a number of ball bearings inside the

LHLH FMOF KOX YG MGOWR
isis that can be heard
```

Using probable words

- From Eli Biham's notes (127 characters)

```
UCZCS NYEST MVKBO RTOVK VRVKC ZOSJM UCJMO MBRJM
VESZB SMOSJ OBKYE MJTRV VEMPY JMOMJ AMVEM HKOVJ
KTRVK CZCQV EMNMV VMJOS ZHVER OVEMP BSZTM MSOKN
PTJCI MZ
```

C-letter	# Occur	Pletter	ExpOcc
M	19	e	15
V	15	t	12
O	11	a	10
J	10	o	10
S	9	n	9
E	8	i	9
K	8	s	8
Z	7	r	8
C	7	h	7
R	6	l	5
T	6	d	5
B	5	c	4
N	3	U	4

C-letter	# Occur	Pletter	ExpOcc
Y	3	u	4
P	3	p	3
H	2	f	3
U	2	m	3
A	1	y	2
I	1	b	2
Q	1	g	2
D	0	v	1
F	0	k	1
W	0	q	0
L	0	x	0
G	0	j	0
X	0	z	0

Breaking mono-alphabet with probable word

- From Eli Biham's notes (127 characters)

```
UCZCS NYEST MVKBO RTOVK VRVKC ZOSJM UCJMO MBRJM
VESZB SMOSJ OBKYE MJTRV VEMPY JMOMJ AMVEM HKOVJ
KTRVK CZCQV EMNMV VMJOS ZHVER OVEMP BSZTM MSOKN
PTJCI MZ
```

- By frequency and contact VEM is likely to be the and thus P is likely y or m.
- Playing around with other high frequency letters UCZCA could be "monoa" which suggests "monoalphabet" which is a fine probable word. The rest is easy.
- Word structure (repeated letters) can also quickly isolate text like "beginning" or "committee"

Breaking mono-alphabet with probable word

```
UCZCS NYEST MVKBO RTOVK VRVKC ZOSJM UCJMO MBRJM
monoa lphab etics ubsti tutio nsare mores ecure
VESZB SMOSJ OBKYE MJTRV VEMPY JMOMJ AMVEM HKOVJ
thanc aesar scsph erbut theyp reser vethe distr
KTRVK CZCQV EMNMV VMJOS ZHVER OVEMP BSZTM MSOKN
ibuti onoft helet tersa ndthu sthey canbe easil
PTJCI MZ
ybrok en
```

Word breaks make it easier

Vigenere polyalphabetic cipher

6 Alphabet Direct Standard Example (Keyword: SYMBOL)

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ      PLAIN:  GET OUT NOW
--------------------------        KEY:  SYM BOL SYM
STUVWXYZABCDEFGHIJKLMNOPQR     CIPHER:  YCF PIE FMI
YZABCDEFGHIJKLMNOPQRSTUVWX
MNOPQRSTUVWXYZABCDEFGHIJKL
BCDEFGHIJKLMNOPQRSTUVWXYZA
OPQRSTUVWXYZABCDEFGHIJKLMN
LMNOPQRSTUVWXYZABCDEFGHIJK
```

Initial Mathematical Techniques

Matching distributions

- Consider the Caeser cipher, $E_a(x)= (x+a) \pmod{26}$
- Let $p_i= P(X=i)$ be the distribution of English letters
- Given the text $\mathbf{y}=(y_0,\ldots,y_{n-1})$ with frequency distribution, q_i, where **y** are the observations of n ciphertext letters, we can find a by maximizing $f(t)= \sum_{i=0}^{25} p_{i+t}\, q_i$.
- t=a, thus maximizes f(t).

Correct alignments

- Here we show that $\sum p_i q_i$ is largest when the ciphertext and plaintext are 'aligned' to the right values.
  - Proof: Repeatedly apply the following: If $a_1 \geq a_2 \geq 0$ and $b_1 \geq b_2 \geq 0$ then $a_1b_1+ a_2b_2 \geq a_1b_2+ a_2b_1$. This is simple: $a_1(b_1-b_2) \geq a_2(b_1-b_2)$ follows from $a_1 \geq a_2$ after multiplying both sides by $(b_1-b_2) \geq 0$.
- A similar theorem holds for the function $\sum p_i \lg(p_i)$ which we'll come across later; namely, $\sum p_i \lg(p_i) \geq \sum q_i \lg(p_i)$.
  - Proof: Since $\sum p_i = 1$ and $\sum q_i =1$, by the weighted arithmetic-geometric mean inequality, $\sum p_i a_i \geq \prod a_i^{p[i]}$. Put $a_i= q_i/p_i$. $1= \sum p_i a_i \geq \prod (q_i/p_i)^{p[i]}$. Taking lg of both sides gives $0 \geq \sum p_i \lg(q_i) - p_i \lg(p_i)$ or $\sum p_i \lg(p_i) \geq \sum p_i \lg(q_i)$.

Statistical tests for alphabet identification

- Index of coincidence (Friedman) for letter frequency
  - Measure of roughness of frequency distribution.
  - Can choose same letters f_i choose 2 ways

    $IC = \sum_i f_i(f_i-1)/(n(n-1))$, so $IC \approx \sum_i p_i^2$
  - For English Text IC $\approx$ .07, for Random Text IC= 1/26=.038.
  - IC is useful for determining number of alphabets (key length) and aligning alphabets.
  - For n letters enciphered with m alphabets: $IC(n,m \approx 1/m\ (n-m)/(n-1)\ (.07) + (m-1)/m\ n/(n-1)\ (.038)$.

- Other Statistics
  - Vowel Consonant pairing.
  - Digraph, trigraph frequency.

Statistical estimation and mono-alphabetic shifts

- Solving for the “shift’’ using the frequency matching techniques is usually dispositive.
- For general substitutions, while frequency matching maximization is very helpful, it is scarcely adequate because of variation from the “ideal” distribution.
- Inter-symbol dependency becomes more important so we must use probable words or look for popular words. For example, in English, “the” almost always helps a lot.
- Markov modelling (next topic) can be dispositive for general substitutions. We introduce it here not because you need it but the mono-alphabet setting is a good way to understand it first time around.
- In more complex situations, it can be critical.

Group Theory in Cryptography

- Groups are sets of elements that have a binary operation with the following properties:
  1. If $x,y,z \in G$, $xy \in G$ and $(xy)z=x(yz)$. It is not always true $xy=yx$.
  2. There is an identity element $1 \in G$ and $1x=x1=x$ for all x in G
  3. For all, x in G there is an element $x^{-1} \in G$ and $x\,x^{-1} = 1 = x^{-1}\,x$

- One very important group is the group of all bijective maps from a set of n elements to itself denoted S_n or Σ_n.

- The "binary operation" is the composition of mappings. The identity element leaves every element alone.

- The inverse of a mapping, x, "undoes" what x does.

Operations in the symmetric group

- If □ □□S_n and the image of x is y we can write this two ways:
  - From the left, y= □ □(x). This is the usual functional notation your used to where mappings are applied “from the left”. When mappings are applied from the the left and □ □and □□are elements of S_n □ □ denotes the mapping obtained by applying □ first and then □ - i.e. y= □ □□(x)).
  - From the right, y=(x) □ □□For them, □ □ denotes the mapping obtained by applying □ first and then □ - i.e. y= ((x)□ □□.

Element order and cycle notation

- The smallest k such that □^{k}=1 is called the *order* of □.

- G is finite if it has a finite number of elements (denoted |G|).
  - In a finite group, all elements have finite order
  - *Lagrange's Theorem:* The order of each element divides |G|.

- Example. Let G= S_4.
  - □= 1→2, 2→3, 3→4, 4→1, □= 1→3, 2→4, 3→1, 4→2.

    □$^{□□}$= 1→4, 2→1, 3→2, 4→3
  - Applying mappings "from the left", □□= 1→4, 2→1,3→2,4→3.
  - Sometimes □ is written like this:

    □= 1 2 3 4

    2 3 4 1
  - Sometimes permutations are written as products of cycles:

    □=(1234)and □□□(13)(24).

William Freidman

Vigenere -polyalphabetic cipher

6 Alphabet Direct Standard Example (Keyword: SYMBOL)

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ       PLAIN:  GET OUT NOW
--------------------------         KEY:  SYM BOL SYM
STUVWXYZABCDEFGHIJKLMNOPQR      CIPHER:  YCF PIE FMI
YZABCDEFGHIJKLMNOPQRSTUVWX
MNOPQRSTUVWXYZABCDEFGHIJKL
BCDEFGHIJKLMNOPQRSTUVWXYZA
OPQRSTUVWXYZABCDEFGHIJKLMN
LMNOPQRSTUVWXYZABCDEFGHIJK
```

Constructing Vig Alphabets

Direct Standard:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Reverse Standard:

```
ZYXWVUTSRQPONMLKJIHGFEDCBA
```

Keyword Direct (Keyword: NEW YORK CITY):

```
NEWYORKCITABDFGHJLMPQRSUVZ
```

Keyword Transposed (Keyword: CHICAGO):

```
CHIAGO
BDEFJK
LMNPQR
STUVWX
YZ
```

```
CBLSYHDMTZIENUAFPVGJQWOKRX
```

Mathematical description of Vigenere

- Suppose we have a sequence letters (a message), s_0, s_1, …, s_n.
- The transposition cipher, □ □ S_m, works on blocks of m letters as follows. Let j= um+v, v<m, $C(s_j)= s_{um+□(v)}$ where the underlying set of elements, S_m, operates on is {0, 1, 2, …, m-1}.
- If the first cipher alphabet of a Vigenere substitution is □ □ S_{26} where the underlying set of elements, S_m, operates on is {a, b, …, z} then $C(s_j)=$ □ $P^{(i \bmod k)}(s_j)$ where P is the cyclic permutation (a,b,c,…,z). Sometimes k=26 or could be the size of the codeword.
- Mixing many of these will obviously lead to complicated equations that are hard to solve.

Solving Vigenere

1. Determine Number of Alphabets
   - Repeated runs yield interval differences. Number of alphabets is the gcd of these. (Kasiski)
   - Statistics: Index of coincidence

2. Determine Plaintext Alphabet

3. Determine Ciphertext Alphabets

Example of Vigenere

- Encrypt the following message using a Vigeniere cipher with direct standard alphabets. Key: JOSH.

```
All persons born or naturalized in the United States, and
subject to the jurisdiction thereof, are citizens of the
United States and of the state wherein they reside. No
state shall make or enforce any law which shall abridge
the privileges or immunities of citizens of the United
States; nor shall any state deprive any person of life,
liberty, or property, without due process of law; nor deny
to any person within its jurisdiction the equal protection
of the laws.
```

- We'll calculate the index of coincidence of the plaintext and ciphertext.
- Then break the ciphertext into 4 columns and calculate the index of coincidence of the columns (which should be mono-alphabets).

Message as “five” group and IC

```
ALLPE RSONS BORNO RNATU RALIZ EDINT HEUNI TEDST ATESA NDSUB JECTT
OTHEJ URISD ICTIO NTHER EOFAR ECITI ZENSO FTHEU NITED STATE SANDO
FTHES TATEW HEREI NTHEY RESID ENOST ATESH ALLMA KEORE NFORC EANYL
AWWHI CHSHA LLABR IDGET HEPRI VILEG ESORI MMUNI TIESO FCITI ZENSO
FTHEU NITED STATE SNORS HALLA NYSTA TEDEP RIVEA NYPER SONOF LIFEL
IBERT YORPR OPERT YWITH OUTDU EPROC ESSOF LAWNO RDENY TOANY PERSO
NWITH INITS JURIS DICTI ONTHE EQUAL PROTE CTION OFTHE LAWS
```

Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq
E	49	0.129	T	42	0.111	I	32	0.084	O	29	0.077
S	28	0.074	N	28	0.074	R	26	0.069	A	25	0.066
H	18	0.047	L	16	0.042	D	13	0.034	U	11	0.029
F	10	0.026	C	9	0.024	P	9	0.024	Y	8	0.021
W	7	0.018	B	4	0.011	M	3	0.008	J	3	0.008
Z	3	0.008	V	2	0.005	G	2	0.005	K	1	0.003
Q	1	0.003	X	0	0.000						

379 characters, index of coincidence: 0.069, IC (square approx): 0.071.

Ciphertext and IC for ciphertext

```
JZDWN FKVWG TVABG YWOLB AODPI SVPWH ZLDBA ANRKA JHWZJ BVZDP BLLHL
VCVWQ DFAZM WUARC FAQSJ LXTSY NQAAR NWUBC XAQSM URHWK BHSAN GSUMC
XAQSK AJHWD QSJLR BLONM JLBWV LWCKA JHWZQ ODSVO CLXFW UOCJJ NOFFU
OODQW UOBVS SUOTY RRYLC VWWAW NPUSY LBCJP VAMUR HALBC XJRHA GNBKV
OHZLD BAANR KAJHW ZWCJZ QODSJ BQZCO LLMSH YRJWH WMHLA GGUXT DPOSD
PKSJA HCJWA CHLAH QDRHZ VDHVB NDJVL SKZXT DHFBG YMSFF CCSUH DWYBC
FDRHZ PWWLZ SIJPB RAJCW GUCVW LZISS YFGAN QLPXB GMCVW SJKK
```

Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq
W	29	0.077	A	28	0.074	S	23	0.061	L	23	0.061
J	22	0.058	H	22	0.058	C	20	0.053	B	20	0.053
D	18	0.047	V	17	0.045	O	15	0.040	Z	15	0.040
R	14	0.037	U	13	0.034	N	12	0.032	Q	12	0.032
F	11	0.029	K	11	0.029	P	10	0.026	G	10	0.026
Y	9	0.024	M	9	0.024	X	8	0.021	T	5	0.013
I	3	0.008	E	0	0.000		0	0.000			

379 characters, index of coincidence: 0.045, IC (square approx): 0.048

Ciphertext broken into 4 columns with IC

```
JNWAW AIWDN JJDLC DMRQX NRBQR BNMQJ QRNBW JQVXO
NUQBU RCAUB VRBRN ODNJW QJCMR WAXOK HAARD NLXFM
CHBRW SBCCZ YNXCJ
```

Column 1: 95 characters, index of coincidence: 0.058, IC (square approx): 0.068.

```
ZFGBO OSHBR HBPHV FWCST QNCSH HGCSH SBMWC HOOFC
OOWVO RVWSC AHCHB HBRHC OBOSJ MGTSS CCHHH DSTBS
CDCHW IRWVI FQBVK
```

Column 2: 95 characters, index of coincidence: 0.077, IC (square approx): 0.087.

```
DKTGL DVZAK WVBLW AUFJS AWXMW SSXKW JLJVK WDCWJ
FOUST YWNYJ MAXAK ZAKWJ DQLHW HGDDJ JHQZV JKDGF
SWFZL JAGWS GLGWK
```

Column 3: 95 characters, index of coincidence: 0.060, IC (square approx): 0.070.

```
WVVYB PPLAA ZZLVQ ZAALY AUAUK AUAAD LOLLA ZSLUJ
FDOSY LWPLP ULJGV LAAZZ SZLYH LUPPA WLDVB VZHYF
UYDPZ PJULS APMS
```

Column4: 94 characters, index of coincidence: 0.081, IC (square approx): 0.090.

Breaking a Vigenere

- Break the Vigeniere based ciphertext below. Plaintext and ciphertext alphabets are direct standard. What is the key length? What is the key?

```
IGDLK MJSGC FMGEP PLYRC IGDLA TYBMR KDYVY XJGMR TDSVK ZCCWG ZRRIP
UERXY EEYHE UTOWS ERYWC QRRIP UERXJ QREWQ FPSZC ALDSD ULSWF FFOAM
DIGIY DCSRR AZSRB GNDLC ZYDMM ZQGSS ZBCXM OYBID APRMK IFYWF MJVLY
HCLSP ZCDLC NYDXJ QYXHD APRMQ IGNSU MLNLG EMBTF MLDSB AYVPU TGMLK
MWKGF UCFIY ZBMLC DGCLY VSCXY ZBVEQ FGXKN QYMIY YMXKM GPCIJ HCCEL
PUSXF MJVRY FGYRQ
```

Look for repeats

```
   1     2     3     4     5     6     7     8     9     10    11
IGDLK MJSGC FMGEP PLYRC IGDLA TYBMR KDYVY XJGMR TDSVK ZCCWG ZRRIP   1
UERXY EEYHE UTOWS ERYWC QRRIP UERXJ QREWQ FPSZC ALDSD ULSWF FFOAM   2
DIGIY DCSRR AZSRB GNDLC ZYDMM ZQGSS ZBCXM OYBID APRMK IFYWF MJVLY   3
HCLSP ZCDLC NYDXJ QYXHD APRMQ IGNSU MLNLG EMBTF MLDSB AYVPU TGMLK   4
MWKGF UCFIY ZBMLC DGCLY VSCXY ZBVEQ FGXKN QYMIY YMXKM GPCIJ HCCEL   5
                        PUSXF MJVRY FGYRQ
```

First Repetition: 20, Second: 25. Third: 35. (20,25,35)=5

ALDSD	FFOAM	IFYWF	NYDXJ	UCFIY	ZBCXM
APRMK	FGXKN	IGDLA	OYBID	UERXJ	ZBMLC
APRMQ	FGYRQ	IGDLK	PLYRC	UERXY	ZBVEQ
AZSRB	FMGEP	IGNSU	PUSXF	ULSWF	ZCCWG
DCSRR	FPSZC	KDYVY	QREWQ	UTOWS	ZCDLC
DGCLY	GNDLC	MJSGC	QRRIP	VSCXY	ZQGSS
DIGIY	GPCIJ	MJVLY	QYMIY	XJGMR	ZRRIP
EEYHE	HCCEL	MJVRY	QYXHD	YMXKM	ZYDMM
EMBTF	HCLSP	MLDSB	TDSVK	YVPU	
ERYWC		MLNLG	TGMLK		
		MWKGF	TYBMR		

IC study of 5 alphabet hypothesis

Full Cipher

Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq
Y	23	0.079	M	21	0.072	C	19	0.066	R	18	0.062
G	17	0.059	L	16	0.055	D	16	0.055	S	15	0.052
F	13	0.045	I	12	0.041	P	11	0.038	E	11	0.038
X	10	0.034	Z	10	0.034	Q	9	0.031	B	8	0.028
K	8	0.028	U	8	0.028	W	7	0.024	A	7	0.024
J	7	0.024	V	7	0.024	N	5	0.017	T	5	0.017
H	4	0.014	O	3	0.010		0	0.000			

290 characters, index of coincidence: 0.044, IC (square approx): 0.047.

Column 1 of 5

Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq
Z	8	0.138	M	6	0.103	A	5	0.086	U	5	0.086
F	5	0.086	I	4	0.069	Q	4	0.069	T	3	0.052
D	3	0.052	E	3	0.052	H	2	0.034	P	2	0.034
G	2	0.034	O	1	0.017	K	1	0.017	V	1	0.017
X	1	0.017	Y	1	0.017	N	1	0.017	S	0	0.000
B	0	0.000	C	0	0.000	J	0	0.000	W	0	0.000
L	0	0.000	R	0	0.000		0	0.000			

58 characters, index of coincidence: 0.059, IC (square approx): 0.075.

IC of columns

Column 2 of 5

Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq
G	7	0.121	Y	7	0.121	C	6	0.103	L	5	0.086
P	4	0.069	R	4	0.069	J	4	0.069	E	3	0.052
B	3	0.052	M	3	0.052	F	2	0.034	D	2	0.034
Q	1	0.017	N	1	0.017	S	1	0.017	T	1	0.017
U	1	0.017	W	1	0.017	I	1	0.017	Z	1	0.017
O	0	0.000	K	0	0.000	V	0	0.000	H	0	0.000
X	0	0.000	A	0	0.000		0	0.000			

58 characters, index of coincidence: 0.058, IC(square approx): 0.074.

Column 3 of 5

Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq
D	8	0.138	S	7	0.121	R	6	0.103	C	6	0.103
Y	6	0.103	V	4	0.069	G	4	0.069	B	3	0.052
X	3	0.052	M	3	0.052	O	2	0.034	N	2	0.034
F	1	0.017	E	1	0.017	K	1	0.017	L	1	0.017
P	0	0.000	Q	0	0.000	A	0	0.000	T	0	0.000
U	0	0.000	H	0	0.000	W	0	0.000	I	0	0.000
J	0	0.000	Z	0	0.000		0	0.000			

58 characters, index of coincidence: 0.071, IC (square approx): 0.087.

IC of columns continued

Column 4 of 5

Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq
L	9	0.155	I	7	0.121	W	6	0.103	X	6	0.103
S	5	0.086	M	5	0.086	R	5	0.086	E	3	0.052
H	2	0.034	V	2	0.034	G	2	0.034	K	2	0.034
A	1	0.017	P	1	0.017	T	1	0.017	Z	1	0.017
C	0	0.000	Q	0	0.000	D	0	0.000	J	0	0.000
U	0	0.000	F	0	0.000	B	0	0.000	N	0	0.000
Y	0	0.000	O	0	0.000		0	0.000			

58 characters, index of coincidence: 0.075, IC (square approx): 0.091.

Column 5 of 5

Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq	Ch	Count	Freq
Y	9	0.155	C	7	0.121	F	5	0.086	M	4	0.069
P	4	0.069	Q	4	0.069	K	4	0.069	J	3	0.052
R	3	0.052	D	3	0.052	G	2	0.034	S	2	0.034
U	2	0.034	B	2	0.034	A	1	0.017	N	1	0.017
E	1	0.017	L	1	0.017	H	0	0.000	O	0	0.000
T	0	0.000	I	0	0.000	V	0	0.000	W	0	0.000
X	0	0.000	Z	0	0.000		0	0.000			

58 characters, index of coincidence: 0.063, IC (square approx): 0.079.

Since the alphabets are standard study most likely slides

Side normal alphabet against input alphabet and check distance:

$D_i = \sum_{i=0}^{25}(d_i - d'_{((i+s)(\mathrm{mod}\ 26))})^2$. d_i is the cipher alphabet frequency, d_i' is the normal alphabet frequency.

Alphabet 1 Slide	Alphabet 1 Distance	Alphabet 1 Slide	Alphabet 1 Distance	Alphabet 2 Slide	Alphabet 2 Distance	Alphabet 2 Slide	Alphabet 2 Distance
00 (A)	0.0656	13 (N)	0.0707	00 (A)	0.0724	13 (N)	0.0494
01 (B)	0.0556	14 (O)	0.0791	01 (B)	0.0733	14 (O)	0.0724
02 (C)	0.0703	15 (P)	0.0723	02 (C)	0.0540	15 (P)	0.0636
03 (D)	0.0753	16 (Q)	0.0603	03 (D)	0.0795	16 (Q)	0.0689
04 (E)	0.0704	17 (R)	0.0621	04 (E)	0.0712	17 (R)	0.0691
05 (F)	0.0775	18 (S)	0.0736	05 (F)	0.0649	18 (S)	0.0693
06 (G)	0.0616	19 (T)	0.0700	06 (G)	0.0730	19 (T)	0.0702
07 (H)	0.0619	20 (U)	0.0693	07 (H)	0.0645	20 (U)	0.0446
08 (I)	0.0401	21 (V)	0.0440	08 (I)	0.0785	21 (V)	0.0752
09 (J)	0.0896	22 (W)	0.0679	09 (J)	0.0625	22 (W)	0.0777
10 (K)	0.0899	23 (X)	0.0704	10 (K)	0.0701	23 (X)	0.0732
11 (L)	0.0666	24 (Y)	0.0816	11 (L)	0.0404	**24 (Y)**	**0.0135**
12 (M)	**0.0163**	25 (Z)	0.0553	12 (M)	0.0784	25 (Z)	0.0754

Slides continued

Side normal alphabet against input alphabet and check distance:

$D_i = \sum_{i=0}^{25}(d_i - d'_{((i+s)(\mathrm{mod}\ 26))})^2$. d_i is the cipher alphabet frequency, d_i' is the normal alphabet frequency.

Alphabet 3 Slide	Distance	Alphabet 3 Slide	Distance	Alphabet 4 Slide	Distance	Alphabet 4 Slide	Distance
00 (A)	0.0764	13 (N)	0.0647	00 (A)	0.0711	13 (N)	0.0929
01 (B)	0.0901	14 (O)	0.0599	01 (B)	0.1091	14 (O)	0.0839
02 (C)	0.0841	15 (P)	0.0763	02 (C)	0.1079	15 (P)	0.0734
03 (D)	0.0836	16 (Q)	0.0838	03 (D)	0.0672	16 (Q)	0.1000
04 (E)	0.0744	17 (R)	0.0799	**04 (E)**	**0.0231**	17 (R)	0.0759
05 (F)	0.0823	18 (S)	0.0907	05 (F)	0.0829	18 (S)	0.0577
06 (G)	0.0849	19 (T)	0.0871	06 (G)	0.0878	19 (T)	0.0508
07 (H)	0.0960	20 (U)	0.0741	07 (H)	0.0751	20 (U)	0.0782
08 (I)	0.0966	21 (V)	0.0752	08 (I)	0.0675	21 (V)	0.0949
09 (J)	0.0718	22 (W)	0.1086	09 (J)	0.0893	22 (W)	0.0971
10 (K)	**0.0338**	23 (X)	0.0919	10 (K)	0.0924	23 (X)	0.0860
11 (L)	0.0755	24 (Y)	0.0494	11 (L)	0.0896	24 (Y)	0.0832
12 (M)	0.0917	25 (Z)	0.0426	12 (M)	0.1074	25 (Z)	0.0876

Slides concluded

Side normal alphabet against input alphabet and check distance:

$D_i = \sum_{i=0}^{25} (d_i - d'_{((i+s)(\mathrm{mod}\ 26))})^2$. d_i is the cipher alphabet frequency, d_i' is the normal alphabet frequency.

Alphabet 5 Slide	Distance	Alphabet 5 Slide	Distance
00 (A)	0.0900	13 (N)	0.0684
01 (B)	0.0696	14 (O)	0.0759
02 (C)	0.0624	15 (P)	0.0846
03 (D)	0.0871	16 (Q)	0.0613
04 (E)	0.0888	17 (R)	0.0724
05 (F)	0.0598	18 (S)	0.0806
06 (G)	0.0763	19 (T)	0.0889
07 (H)	0.0732	20 (U)	0.0466
08 (I)	0.0833	21 (V)	0.0833
09 (J)	0.0663	22 (W)	0.0781
10 (K)	0.0593	23 (X)	0.0661
11 (L)	0.0539	**24 (Y)**	**0.0215**
12 (M)	0.0599	25 (Z)	0.0699

Vigenere Table

Vig Tableau

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
--------------------------
MNOPQRSTUVWXYZABCDEFGHIJKL
YZABCDEFGHIJKLMNOPQRSTUVWX
KLMNOPQRSTUVWXYZABCDEFGHIJ
EFGHIJKLMNOPQRSTUVWXYZABCD
YZABCDEFGHIJKLMNOPQRSTUVWX
```

The answer is…

```
WITHM ALICE TOWAR DNONE WITHC HARIT YFORA LLWIT
HFIRM NESSI NTHER IGHTA SGODG IVESU STOSE ETHER
IGHTL ETUSS TRIVE ONTOF INISH THEWO RKWEA REINT
OBIND UPTHE NATIO NSWOU NDSTO CAREF ORHIM WHOSH
ALLHA VEBOR NETHE BATTL EANDF ORHIS WIDOW ANDHI
SORPH ANTOD OALLW HICHM AYACH IEVEA NDCHE RISHA
JUSTA NDLAS TINGP EACEA MONGO URSEL VESAN DWITH
ALLNA TIONS

Key Length: 5
Key: MYKEY
```

- Cipher only< 25k [assuming 25 letters are required to identify one letter with high certainty, a pretty conservative assumption. You could argue it was as small as about 8k.].

Probable Word Method

$c_i = p_i S C^{i-1}$, $S = (AJDNCHEMBOGF)(IRQPKL)(Z)(Y)(W)(V)(U)(T)(S)$

- Placing a probable word gets several letters.
- Equivalent letters (in the different cipher alphabets) can be obtained be applying C or C^{-1}.

Differencing

Sliding Components

L J T Z G X V Y V T Q G K S Y X S	Cipher Text
B U L L W I N K L E I S A D O P E	Probable Text
J O H N J O H N J O H N J O H N J	Difference

Vigenere Cipher Solutions

- If the alphabets are direct standard, after determining number, just match frequency shapes.

- MIC(x, y)= $\sum f_i f_i'/(n n')$ is used to find matching alphabets

- For both plain and cipher mixed, first determine if any alphabets are the same (using matching alphabets test: **IC=** $\sum(f_i + f'_i)^2$. The only term that matters is $\sum(f_i f'_i)$).)

- Use equivalent alphabets or decimation symmetry of position to transform all alphabets into same alphabet, then use monoalphabetic techniques.

Equivalent alphabets

- Suppose a message is sent with a mixed plaintext alphabet (permuted by □) but a direct standard cipher text alphabet.
- Each position of the message represents the same plaintext letter.
- The Vigenere table looks like this:

```
□(A) □(B) □(C) □(D) □(E) □(F) □(G) □(H) …
------------------------------------------
  A    B    C    D    E    F    G    H   …
  B    C    D    E    F    G    H    I   …
  C    D    E    F    G    H    I    J   …
  D    E    F    G    H    I    J    K   …
  …    …    …    …    …    …    …    …
```

Equivalent alphabets - continued

- If the message bits are m_1, m_2, m_3, … and there are k alphabets used, the message is enciphered as $\square^{-1}(m_1)$, $\square^{-1}(m_2)+1$, $\square^{-1}(m_3)+2$,… or in general $(\square^{-1}(m_i)+(i-1)(\text{mod } k))$ (mod 26)).

- Note that the “columns” retain the correct order of the k enciphering alphabets.

- By substituting the letters (B for A in the second cipher alphabet, etc.), the cipher-text becomes a mono-alphabet which can be solved the usual way.

Mixed plaintext and cipher-text alphabets

- In general, this is harder but may still be solvable with a shortcut. Suppose, for example, we encrypt the same message two different ways (say with k_1 and k_2 mixed plain/cipher alphabets).

- Example from Sinkov. The same message with two different keys.

```
WCOAK TJYVT VXBQC ZIVBL AUJNY BBTMT JGOEV GUGAT KDPKV GDXHE WGSFD
XLTMI NKNLF XMGOG SZRUA LAQNV IXDXW EJTKI TAOSH NTLCI VQMJQ FYYPB
CZOPZ VOGWZ KQZAY DNTSF WGOVI IKGXE GTRXL YOIP
```

```
TXHHV JXVNO MXHSC EEYFG EEYAQ DYHRK EHHIN OPKRO ZDVFV TQSIC SIMJK
ZIHRL CQIBK EZKFL OZDPA OJHMF LVHRL UKHNL OVHTE HBNHG MQBXQ ZIAGS
UXEYR XQJYC AIYHL ZVMQV QGUKI QDMAC QQBRB SQNI
```

Mixed plain and cipher alphabets

- The Vigenere table looks like this:

```
□(A)  □(B) □(C) □(D) □(E) □(F) □(G) □(H) …
------------------------------------------
□(A)  □(B)  □(C) □(D) □(E)  □(F) □(G) □(H)  …
□(B)  □(C)  □(D) □(E) □(F)  □(G) □(H) □(I)  …
□(C)  □(D)  □(E) □(F) □(G)  □(H) □(I) □(J)  …
□(D)  □(E)  □(F) □(G) □(H)  □(I) □(J) □(K)  …
  …     …     …    …    …     …    …    …
```

- If the message bits are m_1, m_2, m_3, … and there are k alphabets used, the message is enciphered as $\square(\square^{-1}(m_1))$, $\square(\square^{-1}(m_2)+1)$, $\square(\square^{-1}(m_3)+2)$,… or in general $\square(((\square^{-1}(m_i)+(i-1)(\text{mod } k))\ (\text{mod } 26))$.

Mixed plain and cipher example

- Plain

  NEWYORKCITABDFGHJKLMPQSUVZ

- Cipher

  CHIAGO

  BDEFJK

  LMNPQR

  STUVWX

  YZ → CBLSYHDMTZIENUAFPVGJQWOKRX

NEWYORKCITABDFGHJKLMPQSUVZ

CBLSYHDMTZIENUAFPVGJQWOKRX

Alphabet rewritten

```
NEWYORKCITABDFGHJLMPQRSUVZ    ABCDEFGHIJKLMNOPQRSTUVWXYZ
--------------------------    --------------------------
CBLSYHDMTZIENUAFPVGJQWOKRX    IENUAFPVGJQWOKRXCBLSYHDMTZ
BLSYHDMTZIENUAFPVGJQWOKRXC    ENUAFPVGJQWOKRXCBLSYHDMTZI
LSYHDMTZIENUAFPVGJQWOKRXCB    NUAFPVGJQWOKRXCBLSYHDMTZIE
SYHDMTZIENUAFPVGJQWOKRXCBL    UAFPVGJQWOKRXCBLSYHDMTZIEN
YHDMTZIENUAFPVGJQWOKRXCBLS    AFPVGJQWOKRXCBLSYHDMTZIENU
HDMTZIENUAFPVGJQWOKRXCBLSY    FPVGJQWOKRXCBLSYHDMTZIENUA
DMTZIENUAFPVGJQWOKRXCBLSYH    PVGJQWOKRXCBLSYHDMTZIENUAF
MTZIENUAFPVGJQWOKRXCBLSYHD    VGJQWOKRXCBLSYHDMTZIENUAFP
TZIENUAFPVGJQWOKRXCBLSYHDM    GJQWOKRXCBLSYHDMTZIENUAFPV
ZIENUAFPVGJQWOKRXCBLSYHDMT    JQWOKRXCBLSYHDMTZIENUAFPVG
IENUAFPVGJQWOKRXCBLSYHDMTZ    QWOKRXCBLSYHDMTZIENUAFPVGJ
ENUAFPVGJQWOKRXCBLSYHDMTZI    WOKRXCBLSYHDMTZIENUAFPVGJQ
NUAFPVGJQWOKRXCBLSYHDMTZIE    OKRXCBLSYHDMTZIENUAFPVGJQW
```

Alphabet rewritten

```
NEWYORKCITABDFGHJLMPQRSUVZ    ABCDEFGHIJKLMNOPQRSTUVWXYZ
--------------------------    --------------------------
UAFPVGJQWOKRXCBLSYHDMTZIEN    KRXCBLSYHDMTZIENUAFPVGJQWO
AFPVGJQWOKRXCBLSYHDMTZIENU    RXCBLSYHDMTZIENUAFPVGJQWOK
FPVGJQWOKRXCBLSYHDMTZIENUA    XCBLSYHDMTZIENUAFPVGJQWOKR
PVGJQWOKRXCBLSYHDMTZIENUAF    CBLSYHDMTZIENUAFPVGJQWOKRX
VGJQWOKRXCBLSYHDMTZIENUAFP    BLSYHDMTZIENUAFPVGJQWOKRXC
GJQWOKRXCBLSYHDMTZIENUAFPV    LSYHDMTZIENUAFPVGJQWOKRXCB
JQWOKRXCBLSYHDMTZIENUAFPVG    SYHDMTZIENUAFPVGJQWOKRXCBL
QWOKRXCBLSYHDMTZIENUAFPVGJ    YHDMTZIENUAFPVGJQWOKRXCBLS
WOKRXCBLSYHDMTZIENUAFPVGJQ    HDMTZIENUAFPVGJQWOKRXCBLSY
OKRXCBLSYHDMTZIENUAFPVGJQW    DMTZIENUAFPVGJQWOKRXCBLSYH
KRXCBLSYHDMTZIENUAFPVGJQWO    MTZIENUAFPVGJQWOKRXCBLSYHD
RXCBLSYHDMTZIENUAFPVGJQWOK    TZIENUAFPVGJQWOKRXCBLSYHDM
XCBLSYHDMTZIENUAFPVGJQWOKR    ZIENUAFPVGJQWOKRXCBLSYHDMT
```

Letter identification and alphabet chaining

- Using IC, we determine first uses 6 alphabets, the second, 5. Same letters at the following positions:

X	**C**	**D**	**V**	**Z**	**A**	**Q**	**Q**	**G**	**I**
12	**15**	**42**	**45**	**72**	**75**	**102**	**105**	**132**	**135**

- Msg1, alphabet 5 = Msg2, alphabet 2. Msg1, alphabet 3 = Msg2, alphabet 5. Can confirm with IC test.

- If we have two rows separated by k (3, in our example):

Plain:	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
Cipher 1:	I	E	M	N	B	U	A	F	T	P	D	V	G	C	Y	J	Q	H	W	Z	O	K	L	R	S	X
Cipher 2:	U	A	I	F	Y	P	V	G	E	J	Z	O	W	S	M	O	K	T	R	N	X	C	H	B	D	L

Alphabet Chaining

Plain:	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
Cipher 1:	I E M N B U A F T P D V G C Y J Q H W Z O K L R S X
Cipher 4:	U A I F Y P V G E J Z Q W S M O K T R N X C H B D L

The decimated interval is:

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

Rearranging by decimation:

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

Rearranging we get the original sequence.

Review of attacks on poly-alphabet

- Letter Frequency, multi-gram frequencies, transition probabilities
- Index of coincidence
- Alphabet chaining
- Sliding probable text
- Limited keyspace search
- Long repeated sequences in ciphertext
- Markoff like contact processes
- Decimation of sequences
- Direct and indirect symmetries

More sophisticated mathematical technique

Estimation-Maximization

- Find the MLE for the parameters $\lambda=(\pi,P,q)$ that maximizes the likelihood of an observed sequence produced by a Markov chain, where **O** consists of T length output sequence (in m symbols) of an HMM with n states.
- Let $S: \lambda \rightarrow \lambda'$ be defined by the maximization formulas on the next slides and $Q(\lambda,\lambda')= \sum_{s\in S} P_{\lambda}(\mathbf{O},s) \lg(P_{\lambda'}(\mathbf{O},s))$.
- Baum showed that if $Q(\lambda,\lambda')>Q(\lambda,\lambda)$ then $P_{\lambda'}(\mathbf{O},s)> P_{\lambda}(\mathbf{O},s)$ and that the sequence of re-estimations converge to a global maximum.
- This re-estimation can be accomplished with $O(n^2(T+1))$ operations using the forward backwards recursion (rather than $O(2(T+1)n^{T+1})$ as the naïve computation might suggest.
- Baum made a lot of money on the stock market using similar techniques; so did James Simons; so did Elwyn Berlekamp.

Hidden Markov Models (HMM)

- Uses more sophisticated source model – fairly general
- Think of cipher as state machine.
- Each state transition depends only on previous state, P(j|i).
- Map from state to output is also given by probability distribution q(o|i). There are m output symbols.
- Output is observed. We have T observations O_0 ,…, O_{T-1}.
- Input (state) is the hidden variable. There are n states.
- Baum offered very efficient procedure to find optimal estimators for this situation

Calculating likelihood for HMMs

1. $\pi(i)$, $\sum_{i=1}^{n-1} \pi(i)=1$ --- Initial Probability
2. $P(j|i)$, $\sum_{j=1}^{n-1} P(j|i)=1$ --- Next State $(n-1 \geq j \geq 0)$
3. $q(j|i)$, $\sum_{j=1}^{n-1} q(j|i)=1$ --- Output symbol $(m-1 \geq j \geq 0)$
4. **$\mathbf{O}$**$= (O_0, \ldots, O_{T-1})$ --- Output observations

$S= \{0,\ldots, n-1\}$, $OS= \{0,\ldots, m-1\}$

- Let $\lambda=(\pi, P, q)$ be the distribution regarded as parameters, then the 'likelihood' of the observation **y** is $P(O=\mathbf{O}|\lambda)= \sum_{\mathbf{x} \in \mathbf{S}^{\mathbf{T}}} P(\mathbf{O}, \mathbf{x})= \sum_{\mathbf{x}} \pi(x_0) \prod_{s=1}^{n} P(x_s|x_{s-1})q(O_s|x_s)$.

Forward-Backwards recursion for HMM

Recall

- $P(O=\mathbf{O}) = \square_{\mathbf{x}}\ P(\mathbf{O,x}) = \square_{\mathbf{x}}\ \square(x_0)\ \square_{s=1}^{n}\ P(x_s|x_{s-1})P(O_s|x_s)$

Define

- $\square_t(i) =$ $\square(i)\ q(O_0)$, if $t=0$;
  $\square_{k=0}^{n-1} P(k|i)\ q(O_t|i)\ \square_{t-1}(k)$, otherwise
- $\square_t(i) =$ 1, if $t=n$
  $\square_{k=0}^{n-1} P(k|i)\ q(O_t|i)\ \square_{t-1}(k)$, otherwise

Then

- $P(O=\mathbf{O}) = \square_t(i) \times \square_t(i)$

Maximization equations

- If $D_X(F)$ denotes the partial derivative of F with respect to X, Lagrange's equations to maximize Y subject to the three stochastic constraints give:

1. $D_{\square(i)} \left(P(O=\mathbf{O}) - \square_1 \square_{k=0}^{n-1} (\square(k)-1)\right) = 0$
2. $D_{P(j|i)} \left(P(O=\mathbf{O}) - \square_2 \square_{k=0}^{n-1} (P(k|i)-1)\right) = 0$
3. $D_{q(j|i)} \left(P(O=\mathbf{O}) - \square_3 \square_{i=0}^{n-1} (q(k|i)-1)\right) = 0$

- The solution (that defined the re-estimated $\square$') is:

$$\square(i) = \square_0(i) = (\square_0(i)\square_0(i)) \; [\square_{k=0}^{n-1} \square_0(k)\square_0(k))]^{-1}, \; j=0,\ldots,n-1$$

$$P(j|i) = [\square_{t=0}^{n-1}(\square_t(i)\, q(y_{k+1}|j)\, P(j|i)\, \square_t(j))][\square_{t=0}^{n-1} \square_t(i)\square_t(i))]^{-1}, \; j=0, \ldots, n-1$$

$$q(j|i) = [\square_{t=0,y(t)=j}^{n-1}(\square_t(i)\, \square_t(i))][\square_{t=0}^{n-1} \square_t(i)\square_t(i))]^{-1}, \; j=0, \ldots, m-1$$

Scaling

- Multiplying a lot of floating point numbers whose absolute value is <1 (as we do in EM) leads to underflow. The renormalization technique to avoid this problem is called *scaling*.
- Put a_{ij}= P(j|i), $b_i(O_t)$= q(i|O_t).
- Set $\alpha_t'(i)= \sum_{j=0}^{(n-1)} \alpha_{t-1}(j)a_{ji}b_i(O_t)$, $\alpha_0'(i)=\alpha_0(i)$, i=1,2,…,n-1.
- $c_0=1/(\sum_{j=0}^{(n-1)} \alpha_0'(j))$, $\alpha_0''(i)=c_0\alpha_0'(i)$.
- For t= 1,2,…,T-1
  - $\alpha_t'(i)= \sum_{j=0}^{(n-1)} \alpha_{t-1}''(j)a_{ji}b_i(O_t)$, $\alpha_t''(i)=c_t \alpha_t'(i)$.
  - $\alpha_{t+1}''(i)=c_{t+1} \alpha_{t+1}'(i)= c_0 c_1 \ldots c_t \alpha_t(i)$ and $\alpha_t''(i)= \alpha_t(i)/(\sum_{j=0}^{(n-1)} \alpha_t(j))$
  - $P(\mathbf{O}|\lambda)= (\sum_{j=0}^{(T-1)} c_j)^{-1}$, $\ln(P(\mathbf{O}|\lambda))= -(\sum_{j=0}^{(T-1)} \ln(c_j))$.
  - Use same scale factor for $\beta_t(i)$, compute $\gamma_t(i)$ as before with $\alpha_t''(i)$, $\beta_t''(i)$ in place of $\alpha_t(i)$, $\beta_t(i)$.

Breaking a mono-alphabet with EM

- m=4, T=48 observations

p: 0.25, 0.25, 0.25, 0.25

P:				
	.2	.2	.5	.1
	.333	.333	.167	.167
	.2	.4	.1	.3
	.5	0	.25	.25

i:	0	1	2	3
q(i|0):	1	0	0	0
q(i|1):	0	0	1	0
q(i|2):	0	1	0	0
q(i|3):	0	0	0	1

50^{th} re-estimation settles on:

i j →	0	1	2	3
0	1.00000	0	0	0
1	.000004	.000001	.906980	.093015
2	.000023	.998303	.001667	0
3	.000023	0	0	.999977

Example from Konheim

Other paper and pencil systems

Poly-graphic Substitution

- PlayFair Digraphic Substitution
  - Write alphabet in square.
  - For two consecutive letter use other two letters in rectangle
  - If letters are horizontal or vertical, use letters to right or below.

```
OHNMA
FERDL
IBCGK          TH→ QM
PQSTU
VWXYZ
```

- Hill's multi-graphic substitution
  - Convert letters into numbers (0→25).
  - Multiply 2-tuples by encrypting 2x2 matrix.
  - Better have inverse in multiplicative group mod 26.

Identifying Playfair

- Rare consonants j, k, q, x, and z will appear in higher frequencies than plaintext and digraphs containing these consonants will appear more frequently
- There are an even number of letters in the ciphertext
- When the ciphertext is broken up into digrams, doubled letters such as SS, EE, MM, . . . will not appear.

Hill Cipher

- Each character is assigned a numerical value
  - a = 0, b = 1, . . ., z = 25
- for ***m*** = 3 the transformation of $p_1p_2p_3$ to $c_1c_2c_3$ is given by 3 equations:

KEY

$$c_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$$

$$c_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$$

$$c_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$$

Slide by Richard Spillman

Hill Matrix

- The Hill cipher is really a matrix multiplication system
  - The enciphering key is an n x n matrix, M
  - The deciphering key is M^{-1}
- For example, if n = 3 one possible key is:

$$M = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \qquad M^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

Encrypt 'n o w'
13 14 22

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 13 \\ 14 \\ 22 \end{pmatrix} = \begin{pmatrix} 23 \\ 20 \\ 4 \end{pmatrix} \bmod 26$$

x u e

Slide by Richard Spillman

Breaking Hill

- The Hill cipher is resistant to a cipher-text only attack with reasonable message size.
  - In fact, the larger the matrix, the more resistant the cipher becomes.
- It is easy to break using a known plaintext attack.
  - The process is much like the method used to break an affine cipher in that the known plaintext/ciphertext group is used to set up a system of equations which when solved will reveal the key.

Hill Cipher

- The Hill cipher is a block cipher with block size is 2 over the “normal” alphabet.
- Assign each letter a number between 0 and 25 (inclusive)
  - For example, a = 0, b = 1, . . ., z = 25 (z is used as space)
- Let $p_1 p_2$ be two successive plaintext letters. $c_1 c_2$ are the cipher-text output where

$$\mathbf{c_1 = k_{11}p_1 + k_{12}p_2 \ (mod\ 26)}$$
$$\mathbf{c_2 = k_{21}p_1 + k_{22}p_2 \ (mod\ 26)}$$

- Apply the inverse of the “key matrix” [k_{11} k_{12} | k_{21} k_{22}] to transform ciphertext into plaintext
- Works better if we add space ($27=3^3$ letters) or throw out a letter ($25=5^2$) so there is an underlying finite field

Breaking Hill

- The Hill cipher is resistant to a cipher-text only attack with limited cipher-text.
  - Increasing the block size increases the resistance.

- It is trivial to break using a known plaintext attack.
  - The process is much like the method used to break an affine cipher. Corresponding plaintext/ciphertext are used to set up a system of equations whose solutions are the key bits.

End