Section 1.2: Cryptanalysis

Math 495, Fall 2008

Hope College

September, 2008

Breaking a Cryptosystem

- What does it mean to *break* a cryptosystem?
- What methods of attack are there?
- What does it mean for a cryptosystem to be *secure*?
- What assumptions should we make when trying to determine whether or not our cryoptosystem is secure?

Kerckhoff's principle

- The system must be substantially, if not mathematically, undecipherable;
- *The system must not require secrecy and can be stolen by the enemy without causing trouble;*
- It must be easy to communicate and remember the keys without requiring written notes, it must also be easy to change or modify the keys with different participants;
- The system ought to be compatible with telegraph communication;
- The system must be portable, and its use must not require more than one person;
- Finally, regarding the circumstances in which such system is applied, it must be easy to use and must neither require stress of mind nor the knowledge of a long series of rules.

from http://www.petitcolas.net/fabien/kerckhoffs/

Security through Obscurity?

- Can we secure data by hiding it?
- Steganography is a great example of this.
- Kerkhoff's principle, which most modern cryptographers would agree with, basically says that security through obscurity is no security at all.
- Kerhkoff's principle is used in another context– open-source software.
- The bottom line
  - Assuming Oscar cannot find a message $=$ bad
  - Assuming Oscar does not know the cryptosystem $=$ bad

Cryptanalysis

- One goal might be to decrypt a single message
- Another goal might be to decrypt a key so that we can decrypt all future messages from Alice.
- A more lofty goal might be to break the system–so that no matter what the key is, you are able to decrypt any message.
- We shall focus on the second one–that of finding the key used.

Common Attack Models

- Ciphertext only
- Known plaintext
- Chosen plaintext
- Chosen ciphertext

Frequency Analysis

- Sometimes properties of the language used can be used to assist in cryptanalysis.
- For instance, we can use the frequency of letters
  See http://en.wikipedia.org/wiki/Letter_frequencies, for instance (Hey, if Wikipedia says it, it has to be true...)
- Digrams and trigrams can also be very useful
  See http://www-math.cudenver.edu/~wcherowi/courses/m5410/engstat.html

Cryptanalysis: Shift Cipher

Example

- Ciphertext is **spsuxogsdsgyevnxdlokcusxq**
- We'll try all 26 possible keys until one makes sense:

```
spsuxogsdsgyevnxdlokcusxq
rortwnfrcrfxdumwcknjbtrwp
qnqsvmeqbqewctlvbjmiasqvo
pmpruldpapdvbskuailhzrpun
oloqtkcozocuarjtzhkgyqotm
nknpsjbnynbtzqisygjfxpnsl
mjmoriamxmasyphrxfiewomrk
lilnqhzlwlzrxogqwehdvnlqj
khkmpgykvkyqwnfpvdgcumkpi
jgjlofxjujxpvmeoucfbtljoh
ifiknewitiwouldntbeasking
hehjmdvhshvntkcmsadzrjhmf
gdgilcugrgumsjblrzcyqigle
fcfhkbtfqftlriakqybxphfkd
ebegjasepeskqhzjpxawogejc
dadfizrdodrjpgyiowzvnfdib
...
```

- Can you see the correct plaintext message?

Cryptanalysis: Affine Cipher

- We only need to know the mapping of any two characters to determine the key.
- A frequency analysis will reveal which ciphertext characters are most common.
- A good strategy is to assume **E** maps to the most common character, and one of **T**, **A**, **O**, **I**, **N**, **S**, **H**, or **R** maps to the second most common.
- Given two mappings, we can reconstruct the key and determine whether or not it makes sense.

Cryptanalysis: Affine Cipher

Example

Assume we have the following ciphertext:

```
FMQVINKVFILSDHAFSXNJSMHZLJSZPVPLKKZAVSENKFZZQLBVELZNHZZLSDZJBVJMQ
UVQJALNZLSBJIVEVAQUZVVLSDVGFBAKVAIJDIFBZFSEPILQLSDSVPAIJDIFBZLMXJ
HFIVSJQEJLSDQUVFZZLDSVEIVFELSDZFSEVGVINLZVZXJHPLKKSJQDVQSVFIKXFZB
HNUJHQJMQUVNJHIZVFZAJZZLWKVFSELQLZKLTVKXXJHIDIFEVPLKKIVMKVNQQUFQ
```

A frequency analysis reveals

- V is most common
- Z, L, S, F, I are pretty frequent
- ZZ appears several times.

A	8	J	18	S	19
B	7	K	14	T	1
C	0	L	22	U	6
D	11	M	6	V	30
E	10	N	9	W	1
F	18	O	0	X	6
G	2	P	6	Y	0
H	9	Q	16	Z	24
I	16	R	0		

- We might guess that $e_K(e) = V$ and $e_K(t) = Z$.
- This is equivalent to $e_K(4) = 21$ and $e_K(19) = 25$.

Cryptanalysis: Affine Cipher

Example Continued

- We think that $e_K(4) = 21$ and $e_K(19) = 25$.
- We also know that $e_K(x) = ax + b$.
- Thus, we think that
  $4a + b = 21$
  $19a = b = 25$
- Solving (see chalk board), we get $a = 2$ and $b = 13$.
- $a = 2$ is not relatively prime with 26, so that is not correct.

- Next we try $e_K(e) = V$ and $e_K(s) = Z$.
- This is equivalent to
  $e_K(4) = 21$ and
  $e_K(18) = 25$.

A	8	J	18	S	19
B	7	K	14	T	1
C	0	L	22	U	6
D	11	M	6	V	30
E	10	N	9	W	1
F	18	O	0	X	6
G	2	P	6	Y	0
H	9	Q	16	Z	24
I	16	R	0		

Cryptanalysis: Affine Cipher

Example Continued

- We think that $e_K(4) = 21$ and $e_K(18) = 25$.
- Thus, we think that

$$\begin{array}{rcrcll} 4a & + & b & = & 21 & (1) \\ 18a & + & b & = & 25 & (2) \end{array}$$

- (2)-(1) yields $14a = 4$, or $14a - 4 = 0$.
- (Why can't we divide both sides by 2 to obtain $7a = 2$?)
- This is equivalent to $2(7a - 2) = 0$.
- Since we are working mod 26, either $7a - 2 = 0$ or $7a - 2 = 13$.
- If $7a - 2 = 0$, $a = 2 * 7^{-1} = 2 * 15 = 4$. But a can't be even.
- If $7a - 2 = 13$, $a = 15 * 7^{-1} = 15 * 15 = 17$.
- Plugging back into $4a + b = 21$, we can obtain $b = 21 - 4a = 21 - 4 * 17 = 5$.

Cryptanalysis: Affine Cipher

Example Continued

- Thus, $e_K(x) = 17x + 5$.
- Then $d_K(y) = 17^{-1}(y - 5) = 23y - 23 * 5 = 23y + 15$
- Applying this to the ciphertext (and adding spaces), we obtain:

```
after clearing up any confusions we will spend class time
discussing some of the topics in more depth seeing example
programs and writing new programs if you are not doing the
assigned readings and exercises you will not get nearly as
much out of the course as possible and it is likely your
grade will reflect that
```

Cryptanalysis: Substitution Cipher

Example

Assume we have the following ciphertext:

```
CUDVFBLUDTUCNUTQGJQKQRNYRTYHVQKBGFYDMBDVFYRYRSTUIZJBBTNBTGLUDTQV
MBRBVDYVBGBKBRVUTQKQTQRNGUIJYRTGMQDQVPUQRVGYRTXYDDULQVPITNBGVFBV
FUINFVGYRTYVVQVITBGUCVFBFBYDVRUVFQRNQRYJJHDBYVQURQGFQTTBRCDUXNUT
GGQNFVBKBDSVFQRNQGIRHUKBDBTYRTJYQTZYDBZBCUDBVFBBSBGUCFQXVULFUXLB
XIGVNQKBYHHUIRV
```

- **B** is the most common letter.
- Following this are **V**, **Q**, **U**, **R**, and **T**.
- We will turn to a simple piece of software to finish analyzing this one.

A	0	J	6	S	3
B	30	K	7	T	20
C	6	L	5	U	22
D	16	M	3	V	26
E	0	N	11	W	0
F	15	O	0	X	5
G	16	P	2	Y	18
H	5	Q	24	Z	3
I	8	R	20		

Cryptanalysis: Substitution Cipher

Example Continued

- If I did everything correctly, you should already have seen that the plaintext in this case is:

  forthewordofgodislivingandactivesharperthananydoubleedgedswordit

  penetrateseventodividingsoulandspiritjointsandmarrowitjudgesthe

  thoughtsandattitudesoftheheartnothinginallcreationishiddenfrom

  godssighteverythingisuncoveredandlaidbarebeforetheeyesofhimto

  whomwemustgiveaccount

Cryptanalysis: Hill Cipher

- Since you all did the reading assignment, you already know that the Hill Cipher does not succumb easily to a ciphertext only attack.
- However, a known plaintext attack works beautifully.
- Recall that encryption is done by computing xK where x is a vector of size m, and $K$is the $m \times m$ encryption matrix.
- Notice that we can encrypt m sets of m characters by placing the m sets of characters in the rows of an $m \times m$ matrix and computing $Y = XK$, and the result is an $m \times m$ matrix of ciphertext.
- Computing K is as easy as computing $K = X^{-1}Y$.
- How do we deal with the following problems?
  - We do not necessarily know m beforehand.
  - The matrix X we form may not be invertible.

Cryptanalysis: Hill Cipher

Example

- I happen to know that the string **pickforit** was encrypted with the Hill Cipher to ciphertext **HMCAFAJSD**.
- Numerically, the plaintext is 15, 8, 2, 10, 5, 17, 8, 19, and the ciphertext is 7, 12, 2, 0, 5, 0, 9, 18 3.
- I live on the edge, so I'll skip $m = 2$ and assume $m = 3$.
- Then the plaintext was encrypted 3 characters at a time.
- The plaintext and ciphertext can be placed in matrices as follows:

$$X = \begin{bmatrix} 15 & 8 & 2 \\ 10 & 5 & 14 \\ 17 & 8 & 19 \end{bmatrix} \text{ and } Y = \begin{bmatrix} 7 & 12 & 2 \\ 0 & 5 & 0 \\ 9 & 18 & 3 \end{bmatrix}$$

- It is not hard to verify that $Y = XK$, since this is identical to encrypting each of the 3 sections independently, and then placing them in rows of a matrix.

Example Continued

- Since $Y = XK$, we also know that $K = X^{-1}Y$, assuming X is invertible.
- Through much effort, we determine that
$$X^{-1} = \begin{bmatrix} 11 & 10 & 12 \\ 24 & 15 & 22 \\ 17 & 8 & 17 \end{bmatrix}$$
- Then we know that
$$K = X^{-1}Y = \begin{bmatrix} 11 & 10 & 12 \\ 24 & 15 & 22 \\ 17 & 8 & 17 \end{bmatrix} \begin{bmatrix} 7 & 12 & 2 \\ 0 & 5 & 0 \\ 9 & 18 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 8 & 6 \\ 2 & 5 & 10 \\ 12 & 4 & 7 \end{bmatrix}$$
- How do we verify that K is the correct matrix?

Cryptanalysis: Index of Coincidence

Definition: *index of coincidence*

Given a string x of n alphabetic characters, the *index of coincidence*, denoted $I_c(x)$, is the probability that two elements chosen at random from x are the same.

- Let x be a string of length n.
- Then $I_c(x)$ is number of ways of choosing 2 identical characters from x divided by the number of ways of selecting any 2 characters from x.
- Let $f_0, f_1, \ldots, f_{25}$ denote the frequencies of $A, B, \ldots, Z$ in x.
- The former number is $\sum_{i=0}^{25} \binom{f_i}{2}$ and the latter is just $\binom{n}{2}$.
- Thus, $I_c(x) = \dfrac{\sum_{i=0}^{25} \binom{f_i}{2}}{\binom{n}{2}}$
- How does $I_c(x)$ change if x is encrypted with a monoalphabetic cipher?

Cryptanalysis: Index of Coincidence

- Let x be a string of English text (of sufficient length).
- Let $p_0, p_1, \ldots, p_{25}$ denote the probabilities that $A, B, \ldots, Z$ occur in English text (From Table 1.1 of your text, for instance).
- Then the probability that 2 characters chosen from x at random are both A is p_0^2, are B is p_1^2, etc.
- Thus, we might then expect that $I_c(x) \approx \sum_{i=0}^{25} p_i^2 = 0.065$.
- If x is random, we would expect that $I_c(x) \approx 26\left(\frac{1}{26}\right)^2 = \frac{1}{26} \approx 0.038$.
- Thus, the *index of coincidence* can actually be used as evidence that a monoalphabetic cipher was used to encrypt English text.

Cryptanalysis: Vigenère Cipher

- Assume a plaintext was encrypted using a key of length m.
- Then we can partition the ciphertext into m segments by taking every m-th character to be in each segment.

Example

- Assume $m = 3$, and the ciphertext is `AIDOLFHSNNTEU`
- Then we can partition the ciphertext into 3 segments:
  `AOHNU`
  `ILST`
  `DFNE`

- Each of these segments was encrypted with a shift cipher.
- Thus, if we compute the *index of coincidence* for each of these segments, it should be close to .065.
- Two methods of guessing m
  - Kasiski test
  - Trial and error

Cryptanalysis: Vigenère Cipher

Example

- Assume the following text was encrypted using the Vigenère Cipher:

```
GJABWAIBTLUCRGAFBQKTIWWDMHLVEKOGGNIDASFGTFULIACLRDSLXDNETAO
ZECSBDEYPRVAHHODYWIKTEDQESWTAUGIRLIOTDAHACTPCICGNZTNXEVBKTO
KEKEMRSZEJTQMSESSCQGEEIFNGBMIRGFBTMFLWSBALGEVOZXEMNLHQVCEPZ
TAISIRAGTIELENEDITGEKADTCIPJEETYXEVBKRTVCDEESTEMGNPHEVISNSA
PWAFDDTNXAFGXTDXHWNGBMIRKRQECISWNFISINMMNTCSFHENQWISGNQPNLV
WRFTIXHWLUCPWBWTITPRTZEHTCXIUEEPCITZEQSRISHENQWISUAZQPQONEP
UCSMSVQGEIXLOMCJZEJTQMELALIERZRNWCFTOFYSNQSRIBMTFWPVEASMRLX
CZ
```

- The first step is to determine the key length.

Example Continued

- We can computer the index of coincidence for various values of m for each segment:

m	$I_c(x)$
1	0.0461
2	0.0454 0.0466
3	0.0391 0.0458 0.048
4	0.0428 0.0442 0.0489 0.0431
5	0.0447 0.0532 0.0394 0.0417 0.0414
6	0.0435 0.0468 0.0451 0.0398 0.0414 0.0443
7	0.0582 0.0621 0.0746 0.0554 0.0701 0.0599 0.114
8	0.0475 0.0437 0.0430 0.0430 0.0483 0.0528 0.0520 0.0475
9	0.0416 0.0324 0.0666 0.0398 0.0509 0.0481 0.0342 0.0416 0.0324

- It appears $m = 7$ is the most likely value

- What if, for instance, $m = 6$ and $m = 3$ both had high values?

Cryptanalysis: Vigenère Cipher

- We know how to determine the key length
- How do we determine the actual key?
- Since each segment of text was encrypted using the shift cipher, it turns out that there is a technique that works well.
- Let p_i and f_i be as defined earlier, and let $n' = n/m$ (the length of each segment).
- For $0 \leq g \leq 25$, define $M_g = \sum_{i=0}^{25} \frac{p_i f_{i+g}}{n'}$.
- If g corresponds to the key used for this segment, then we would expect that $\frac{f_{i+g}}{n'} \approx p_i$, so $M_g \approx \sum_{i=0}^{25} p_i^2 = 0.065$.
- If g does not correspond to the key used for this segment, M_g will generally be much smaller than 0.065.
- So for each of the m segments, we compute all 26 values of M_g, and pick the g for which this is the maximum. This should give us the key.

Example Continued

- We compute the values of M_g for our previous example:

1	**0.063** 0.04 0.029 0.03 0.05 0.031 0.029 0.033 0.037 0.036 0.038 0.045 0.038 0.046 0.041 0.044 0.039 0.036 0.031 0.039 0.035 0.028 0.043 0.039 0.036 0.04
2	0.033 0.033 0.038 0.048 0.037 0.044 0.039 0.048 0.04 0.036 0.029 0.035 0.032 0.033 0.047 0.035 0.029 0.04 **0.069** 0.041 0.032 0.035 0.047 0.028 0.033 0.04
3	**0.07** 0.04 0.034 0.026 0.04 0.032 0.036 0.031 0.033 0.039 0.038 0.047 0.036 0.046 0.034 0.043 0.035 0.034 0.028 0.04 0.036 0.033 0.041 0.038 0.031 0.041
4	0.037 0.042 0.04 0.034 0.031 0.036 0.035 0.028 0.044 0.036 0.034 0.037 **0.064** 0.041 0.032 0.029 0.042 0.03 0.034 0.037 0.034 0.034 0.038 0.05 0.038 0.045
5	0.05 0.041 0.047 0.033 0.04 0.037 0.037 0.028 0.041 0.03 0.03 0.044 0.043 0.037 0.04 **0.065** 0.039 0.032 0.028 0.044 0.026 0.03 0.03 0.033 0.039 0.04
6	0.052 0.033 0.039 0.026 0.041 0.034 0.032 0.033 0.035 0.044 0.041 **0.058** 0.033 0.037 0.036 0.046 0.03 0.035 0.031 0.031 0.038 0.036 0.039 0.032 0.052 0.041
7	0.047 0.04 0.031 0.035 **0.078** 0.036 0.033 0.026 0.048 0.025 0.033 0.028 0.033 0.034 0.034 0.05 0.041 0.053 0.036 0.042 0.039 0.04 0.028 0.043 0.025 0.026

- It is evident that the key is likely **ASAMPLE**.
- Applying this to the ciphertext (by subtracting, of course), we obtain (after adding spaces):

graph pebbling can be viewed as a resource distribution and allocation problem A graph consists of a set of vertices which are connected by edges Each vertex has a certain number of pebbles placed on it In the graph to the right the vertices are represented by circles diamonds triangles and rectangles The numbers represent the number of pebbles on each vertex The lines between the vertices are the edges Pebbles can be moved from a vertex to any vertex that is connected by an edge but there is a catch: moving one pebble costs an additional pebble So moving a pebble from a vertex with two pebbles to an adjacent vertex with zero pebbles would result in the first vertex having zero pebbles and the second vertex having one

Cryptanalysis: LFSR Stream Cipher

- Recall that the keystream is produced from an initial m-tuple $(z_1, \ldots z_m)$ using the recurrence

$$z_{m+i} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2$$

  where the values $c_0, \ldots c_{m-1}$ are fixed.
- We need to determine the values $z_1, \ldots z_m$ and $c_0, \ldots c_{m-1}$.
- As a first step, we can easily reconstruct the keystream given a plaintext/ciphertext pair. (what type of attack?)

Example

plaintext XOR ciphertext = keystream

plaintext	`1001111101011000`
ciphertext	`0010101001101011`
keystream	`1011010100101001`

Cryptanalysis: LFSR Stream Cipher

- The recurrence $z_{m+i} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2$ leads to:

$$(z_{m+1}, z_{m+2}, \ldots, z_{2m}) = (c_0, c_1, \ldots c_{m-1}) \begin{pmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \vdots & \vdots & & \vdots \\ z_m & z_{m+1} & \cdots & z_{2m-1} \end{pmatrix}$$

- If we have at least $n \geq 2m$ plaintext/ciphertext pairs, we can determine the values $z_1, \ldots z_{2m}$ as in the previous example.
- Thus, we only need to determine the values of $c_0, \ldots c_{m-1}$
- We can solve the above for $(c_0, c_1, \ldots c_{m-1})$ to obtain

$$(c_0, c_1, \ldots c_{m-1}) = (z_{m+1}, z_{m+2}, \ldots, z_{2m}) \begin{pmatrix} z_1 & z_2 & \cdots & z_m \\ z_2 & z_3 & \cdots & z_{m+1} \\ \vdots & \vdots & & \vdots \\ z_m & z_{m+1} & \cdots & z_{2m-1} \end{pmatrix}^{-1}$$

Cryptanalysis: LFSR Stream Cipher

Example (continued)

- Given the plaintext and corresponding ciphertext, we already determined the keystream bits:

  plaintext `1001111101011100`
  ciphertext `0010101001110101 1`
  keystream `1011010100101001 1`

- The first m bits are $(z_1, \ldots z_m)$.
- If we assume $k = 4$, the equation we need to solve is

$$(c_0, c_1, \ldots c_{m-1}) = (0, 1, 0, 1) \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}^{-1}$$

- If we felt like it, we could compute the inverse, and easily determine $(c_0, c_1, \ldots c_{m-1})$