Cryptology

Background

Goals:

- ***Confidentiality*:** It should cost Eve more to recover m than m is worth.
- ***Authentication*:** Bob should be able to verify that it was Alice that sent m.
- ***Integrity*:** Bob should be able to verify that m was not tampered with.
- ***Nonrepudiation*:** Alice should not be able to deny sending m.

The best cryptosystems assume that Eve and Mallory know E, D, and c. Most cryptosystems do **not** rely on their algorithms being kept secret, because

- If your secret algorithm is compromised (someone could leave your group), you have to change it, and that is way harder than just changing a key!
- Public algorithms can be subjected to thousands of experts looking for flaws, so you have some degree of confidence in those that withstand scrutiny.

The study of cryptology includes the design of various ciphers, cryptanalysis methods (attacks), key exchange, key authentication, cryptographic hashing, digital signing, and social issues (legal, political, etc.).

Definitions

Cryptography
: The art and science of **making** ciphers.

Cryptanalysis
: The art and science of **breaking** ciphers. In other words, the extraction of m given c, E, D, and possibly k_e.

Cryptology
: The study of cryptography and cryptanalysis.
Exercise: Find out about steganography. How is it different from cryptography?

Cryptosystem
: A particular suite of algorithms and protocols for encryption, decryption, and key generation. Examples: Cramer-Shoup cryptosystem, Rabin cryptosystem, Benaloh cryptosystem, RSA cryptosystem.

Cryptographic System
: Any system that uses cryptography.

Cipher
: An algorithm used in a cryptosystem.
Exercise: How is a "code" different from a "cipher". Are codes more secure than ciphers? Why aren't they used as often?

Confusion
: The property of having the relationship between the plaintext, ciphertext, and key so complicated as to be useless to the cryptanalyst.

Diffusion

The property of having statistical patterns in the plaintext spread widely throughout the ciphertext.

Kinds of Ciphers

Here are some useful categories of ciphers. Note that a particular cipher may belong to more than one of these categories.

- Classical (a.k.a. manual): A cipher easy enough to be performed by hand, usually character-based
- Modern: Pretty much any cipher that isn't "classical"
- Substitution: Each character of the plaintext is replaced with one or more characters to make the ciphertext
- Transposition: Characters in the plaintext are scrambled to form the ciphertext
- Monoalphabetic: A substitution cipher in which a character of the plaintext is always replaced by the same character
- Polyalphabetic: A substitution cipher that essentially uses multiple monoalphabetic substitution mappings
- Homophonic: A substitution in which one character can map to one of a set of characters
- Polygraphic: A substitution of blocks of characters for blocks of characters
- Periodic: A polyalphabetic cipher in which the replacement scheme "repeats"
- Non-periodic: (Self-explanatory if you understand periodic)
- Block: Encryption takes place not per character but per "blocks" of characters.
- Stream: A cipher operating on a data stream of unknown length, usually incorporating feedback.
- Secret Key: A cipher in which k_e and k_d are the same or trivially derivable from one another; requires the parties to meet in secret to exchange the keys they'll be using. Also called *symmetric*.
- Public Key: A scheme in which everyone's encryption key is publicly known but their decryption key is kept secret. Also called *asymmetric*.

NOTE: In the character-based examples below, we'll assume (without any loss of generality) a 26 symbol alphabet ('A'..'Z').

Secret Key Cryptography

Secret key (a.k.a. symmetric key) ciphers are much faster than public key ciphers, but key management can be a huge problem.

- If n people in a group need to communicate, they need $n(n-1)/2$ keys.
- Keys must be distributed securely (in secret).
- Keys must be kept safe.
- Keys should be changed frequently, which feeds back into the distribution headache.

Caesar Cipher

A completely pathetic and insecure cipher by modern standards. The encryption key k_e is a small integer and $k_d = k_e$. To encrypt, add k_e to each plaintext character; to decrypt, subtract.

Example: For k=5, "ATTACKATDAWN" becomes "FYYFHPFYIFBS"

Trivial to crack: just guess k_e.

Monoalphabetic Substitution

Instead of simply adding a fixed offset to each character, you can precompute a "substitution table" by generating a random permutation of your alphabet. For example:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
MQHPSVJYCURFTBILAKWNGZDOEX
```

Now "ATTACKATDAWN" is now "MNNMHRMNPMDB".

You don't crack this by guessing the key (there are n! possible keys), but frequency analysis can crack any monoalphabetic substitution cipher, provided the message is long enough.

For techniques whose "key" is a permutation, one way to make the key easier to remember is to pick a phrase, lay out its unique letters, then fill in missing letters in order. For example, "PREMATURE OPTIMIZATION IS THE ROOT OF ALL EVIL" yields this substitution mapping:

```
PREMATUOIZNSHFLVBCDGJKQWXY
```

Homophonic Substitution

Each plaintext letter maps to one or more symbols in the ciphertext. The number of targets should be proportional to its frequency (to defeat frequency analysis). Example:

```
A  12 15 36 50 56 70 81 95
B  51 84
C  16 44 65
D  04 06 48 82
E  01 17 19 34 47 49 58 60 67 77 85 90
F  13 27
G  09 28
H  26 42 53 59 68 71
I  35 73 76 86 91 96
J  18
K  07
L  29 40 54 87
```

M 25 30
N 21 61 62 69 74 94
O 02 03 08 10 57 75 93
P 41 98
Q 97
R 32 38 43 45 80 83
S 14 22 39 79 89 99
T 00 20 23 33 46 52 72 78 88
U 11 64 66
V 37
W 63 92
X 31
Y 24 55
Z 05

To encrypt, choose *randomly* among possibilities. Example, one possible encryption of "ATTACKATDAWN" is

56 78 20 95 65 07 12 72 06 50 92 61

Polyalphabetic Substitution

Polyalphabetic substitution ciphers use multiple substitution alphabets. There are quite a few ways to do this.

Simple Vigenère

The cipher known as the simple shift Vigenère cipher was not invented by Vigenère at all... it seems to have been first described by Giovan Battista Bellaso. The key is a string that you add to the plaintext with modular addition, like in this example (A=0, B=1, C=2, ..., Z=25):

Plaintext:
TAKEACOPYOFYOURPOLICYTONORMAWILCOXONTHETHIRDFLOOR
Key:
QUARKQUARKQUARKQUARKQUARKQUARKQUARKQUARKQUARKQUA
R
Cipher text:
JUKVKSIPPYVSOLBFILZMONOEYHGANSBWOOYDNHVDXCRUPBIOI

To generate cipher text by hand you can use a code wheel or tabula recta.

This scheme isn't secure since the key repeats. If the key length can be determined, the cryptanalyst can do multiple frequency analyses (one for each shift value in the key). Methods for determining key length are the Kaisiski Method and the Friedman test.

For "binary data" (i.e., a sequence of bits) modular addition base-2 is just a simple xor . Example:

Plaintext: 01100001010100001111010010101010010000001111101
Key: 00000111000001110000011100000111000001110000001110
Ciphertext: 01100110010101111111001110101101100101110011100 11

Auto-Key Vigenère

Vigenère actually created an auto key cipher which is stronger because the key never repeats. Instead the "key" is made up of the key phrase followed by the plaintext, like this:

Plaintext:
TAKEACOPYOFYOURPOLICYTONORMAWILCOXONTHETHIRDFLOOR
Key:
QUARKTAKEACOPYOFYOURPOLICYTONORMAWILCOXONTHETHIRD
Cipher text:
JUKVKVOZCOHMDSFUMZCTNHZVQPFOJWCOOTWYVVBHUBYHYSWFU

That one used the plaintext as part of the key. You could also use the cipher text. See how?

Modern Auto-Key Ciphers

You can still crack autokey Vigenère ciphers by linguistic analysis, because the key contains text and is thus likely to have high-frequency letters. Modern auto-key ciphers generate the shift values with a random number generator. The key seeds the generator.

Exercise: Implement an autokey cipher in Java.

One Time Pad

If the key

- is as long as or longer than the message being encoded
- is truly randomly generated
- is used once and only once

Then you have a **provably secure** cipher called the *one time pad*. Your actual algorithm can use polyalphabetic substitution or even simple xoring the message with the key, as long as you meet the three criteria above.

The one-time pad can never be cracked. It is a perfect encryption scheme, from a mathematical perspective, anyway.

Exercise: Why aren't one time pads commonly used, then, given that they are the most secure ciphers possible?

Playfair

This is an example of a *poly graphic substitution cipher*. It replaces pairs of characters. The key is a permutation of {A..I,K..Z}, for example:

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

To encrypt, write out the plaintext (without spaces or punctuation), sticking in an X between double letters and at the end if necessary to make the text have even length. Then for each pair of letters:

- Let (a,b) be the row and column of the first character and (c,d) be the row and column of the second.
- If a ≠ c and b ≠ d then return (a,d)(b,c).
- If a = c then return (a,(b+1)%5)(c,(d+1)%5).
- If b = d then return ((a+1)%5,b)((c+1)%5,d).

Example: "THEN ATTACK FROM THE EAST" □ "TH EN AT XT AC KF RO MT HE XE AS TX" □ "UT HW GO FO DB TV YK ZK NH NA DX OF".

Decryption runs the rules in reverse. The Play fair cipher is pretty insecure.

Four-square

Encrypts digraphs like play fair, but slightly stronger because it allows for double letters and doesn't yield reversed cipher text digraphs for reversed plaintext digraphs. Example

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

P R E M A a b c d e
T U O I Z f g h i k
N S H F L l m n o p
V B C D G q r s t u
K Q W X Y vwx y z

For which "THEN ATTACK FROM THE EAST" □ "TH EN AT TA CK FR OM TH EE AS TX" □ "NI VL EV FM MO BV DF NI MA VV NX".

Okay, so slightly stronger than Playfair but so what! Computers can crack these things in seconds, or perhaps minutes (given enough ciphertext).

Simple Block Transposition

The simplest transposition cipher breaks up the message into blocks of size n, then scrambles each block according to a permutation of (1..n).

Example: For the key (4,1,6,3,2,5) the message "GETTHATHEALTHINSPECTOR" becomes "TGATEHATTEHLSHENIPRCOT".

Columnar Transposition

Write out the message row by row in a grid, then read it out in columns. Totally insecure. The key is just the number of rows. Guess it.

Rail Fence

The rail fence is no better than the last one, just funkier. The key is the number of "rails" on which you write the plaintext in an up and down fashion, generating the ciphertext by reading one rail at a time.

Example: To encode "fill out and file a WS2475 form" on 4 rails:

```
f   t   l   4   m
 i u a  i e  2 7  r
  l o  n f  a s  5 o
   l   d   w   f
```

you then read out the ciphertext "ftl4miuaie27rlonfas5oldwf". This is trivial to crack. Just guess k.

Combining Substitution and Transposition

Transposition alone is very weak; substitution is weak; combining them is better. You can mix a lot of the classic substitution ciphers with various transpositions, or use some special combination ciphers like bifid. Also, most of the famous rotor machines and modern ciphers use this combination; in fact they apply these transformations many times.

Bifid

This one substitutes letters with their coordinates in a grid and does a columnar transposition on the coordinates. Example:

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

Write the (row, column) coordinates under each letter of the plaintext (e.g., "A" is at row 1, column 2; "T" is at row 2, column 0, etc.):

ATTACKATDAWN
122102121143
200213201244

Then read out in rows, group by twos and look up the ciphertext letters

122102121143200213201244
A U B A D R T B Q T A W

Trifid

Like Bifid, but on a cube. Example:

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

To encrypt, first write the coordinates

ATTACKATDAWN
000001000022
122102121110
200210201221
000001000022122102121110200210201221
Z C Z O S F H Q V I N .

Enigma

The Enigma was the famous German rotor machine from World War II (actually a family of machines). Most versions implemented a polyalphabetic substitution cipher with a period of 16900 plus a plugboard for scrambling (transposition). The "key" consisted of the order of the rotors, the starting position of each roter, the ring settings, and the plugboard settings (about 1.6×10^{20} possibilities). There was a new key each day (more or less) prepublished in codebooks.

The Allies were able to crack it thanks to some weaknesses in its design...

- No letter would encrypt to itself
- Self-reciprocity meant that there were fewer scrambler setup possibilities

...but more importantly, many weakness in the way it was used...

- It was really easy to find cribs. Most messages began with a weather report.
- Early on, message keys appeared twice in succession.

...and by obtaining codebooks from captured vessels.

You can read about how the Enigma was broken from the NSA, and from Wikipedia.

Modern Cryptographic Methods

Now that we have Shannon's information theory, very powerful computers, and centuries of theory and practice behind us, most modern techniques

- operate on bit strings, not character strings
- are careful to completely mask patterns and redundancies in the plaintext
- use **random** keys (that can be reused, too)
- ensure that very slight changes in the plaintext affect a large portion of the ciphertext (and vice versa). This is called the **Avalanche Effect**.

In addition, it's nice if the cipher is

- efficient
- fault-tolerant

Most ciphers are either block ciphers or stream ciphers. Block ciphers require padding and can operate in different *modes* (See Schnier's book or the Wikipedia article:

- ECB — Electronic Codebook
- CBC — Cipher Block Chaining
- CFB — Cipher Feedback
- OFB — Output Feedback
- CTR — Counter
- BC — Block Chaining
- PCBC — Propagating Cipher Block Chaining
- CBCC — Cipher Block Chaining with Checksum