Cryptography summary

Intro (monoalphabetic cypher)

Ceasar cypher:

```
Map a plain alphabet to a single shifted cipher alphabet:
    * Plain  : abcdefghijklmnopqrstuvwxyz
    * Cypher : defghijklmnopqrstuvwxyzabc

CRACKED by using rotation.
```

Ceasar variation:

```
A variation of this is the instead of a simple shift, map the normal alphabet with
a customized alphabet

CRACKED by using frequency analysis (mapping frequency of letters to the alphabet)
```

Classic cyphers

General approach to transmission:

```
1. Compression
2. Encryption
3. Add error detection + recovery
```

stenography vs cryptography

```
a) conceals the existence of message
b) accessible cypher but the meaning is concealed
```

Basic Cryptanalysis

```
1. frequency
2. crib (common combination of letters)
3. informed guessing (context, length)
```

Tips

```
1. A E I O (high frequency) | U (moderate) | Y (low frequency)
2. Letters next to low frequency letters are usually vowels
3. Letters with a large variety of combinations are vowels
4. In repeated diagrams (pair of letters) 1 is a vowel
5. In reversed diagrams (VX, XV) 1 is a vowel
```

```
6. Doule consonants are bordered by vowels
7. No more than 5 consonants in a row
8. Rare combinations of different vowels
```

Polyalphabetic ciphers

```
Instead of a single replacement alphabet, we have multiple ones, and we alternate
between them:

    Plain : abcdefghijklmnopqrstuvwxyz
    C1    : fzbvkixaymeplsdhjorgnqcutw
    C2    : goxbfwthqilapzjdesvycrkuhn

    This counters frequency analysis because same letters will be encrypted into
different letters (a -> f or g)
```

Vigenère Cipher:

```
polyalphabetic cipher with 26 alphabets that are decided by a key.


  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

1. have the Vigenere Table
2. choose a key (ex: love)
3. Encrypt the plaintext by encrypting each letter with the rotations of the love
alphabets.
key will repeat itself along the plaintext: lovelovelovelove (indicates which
alphabet to use on the top
and replace the left letter with the one in the column)
4. Decryption works in reverse

CRACKING:
    a) Find key length
    b) Divide message into sub-ciphers
    c) Use frequency analysis to resolve the sub-ciphers

    How to find the key ?
        a) kasisky test
        b) Index of coincidence


        Kasisky test:
            Some word might be encrypted the same way, the space between these
repetitions should be a multiple of the key length

            1) Search for pairs of identical segments (min 3 letters)
            2) Record the distances between segments (8, 12, 16)
            3) key length should divide the greatest common divider (2.2.2, 2.2.3,
2.2.2.2 -> gcd is 2.2)

        Index of coincidence:
            Looks at the probability the randomly selected letters are identical:
                IC = somme_a->Z (f_i(f_i -1)) / N(N-1) (english=0.066 |
random=0.038p)

                How to use this index:
                    1. Assume a text y_1,y_2,y_3,y_4,y_5,y_6,y_7,y_8,..., y_n
                    2. Make a guess of the key length m
                    3. Arrange the text into matrix of m rows*
                    y_1,y_m+1...y_n-m+1
                    y_2,y_m+2...y_n-m+2
                    ...
                    y_m,y_2m...y_n


                    4. We calculate IC for each row
                    5. If we have guessed right then the IC for each row should
average the IC of the language expected.
                    6. Repeat until you find the right length
```

Transposition ciphers

```
The idea is to change the order of basic units without changing their value.(rail
and columnar transposition)
```

Rail Fence Cipher

```
1. We decide of a number of rails
2. Read the message by going down the rails and then up in triangle fashion
3. Obtain the cipher by reading the rows.
```

```
CRACKING:
    1. Guess the number of rails
    2. Position the spots needed for reading of the cipher in reverse.
    3. Read in the reverse order the plan text.
```

Colmunar Transposition Cipher

```
1. Decide of a length(with a key)
2. Read the message by going to the line at the end of the length + fill the last
line with random letters if smaller then the key length.

Alphabetical Order  1256347
Key                 DESTINY
Plain               attackp
                    ostpone
                    duntilt
                    wofgxnu

3. Reorder the columns in alphabetical order

Alphabetical Order  1234567
Key                 DEINSTY
Plain               atcktap
                    osontpe
                    duilntt
                    woxnfgu

4. You get the cipher by reading by columns and no spaces to hide the nb of rows
    AODWTSUOCOIXKNLNTTNFAPTGPETU

CRACKING:
    1. count the cipher length -> key must be a multiple
    2. find anagrams in the columns obtained
    3. patterns to find the order
    4. If nothing found try with the next possible key length
    5. Repeat process
```

Fitness metrics

```
Value used to determine when doing cryptanalysis if the technique used found the
plaintext or not:
    single letter frequencies
    diagrams statistics
    trigrams statistics
    quadgrams statistics

Text similar to english will get a high fitness score, random or cipher text will
get low scores.
```

CRACKING TRANSPOSITION CIPHERS

```
- Brute-Force attacks: assume short key words
- Dictionary attacks:
    1. compile wordlist
    2. text file of possible keys
    3. Decrypt the cypher text with possible words and record the plaintext thanks to the high fitness scores
- Hill climbing:
    1. Assume a key length N + a random keyword with that length (key parent)
    2. Apply parent key and assess fitness score
    3. Generate child key by swaping letters in parent key
    4. Apply child key -> a) if kid fitness > parent fitness => parent == key
                          b) if kid fitness < parent fitness => keep the parent for the swaping process
    5. Rank different encryption keys by fitness score until one gets something that looks good
    6. If after 100 swaps no good scores => assume a new random word
    7. If still no results after several hundred times, increase N by 1
```

MODERN CIPHERS:

```
- combinations of transposition and substitutions (2 subst = harder subst, 2 trans = harder trans, subs + trans = way harder cipher)
```

VISUAL STENOGRAPHY:

```
 24-bit image => 3x8 RGB, hide message in LSB for each plane.
 cf links for advanced information
```