Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 2 Part 1

Vigenère Cipher

The Vigenère cipher

- **Key is a string**, not a character
- Encrypt: shift each character in the plaintext by the amount dictated by the corresponding character of the key
- Wrap around in the key as needed
- Decryption just reverses the process

```
tellhimaboutme
cafecafecafeca
veqpjiredozxoe
```

The Vigenère cipher

- Size of key space?
- Let key be **14**-character English string
- $\implies$ key space has size $\mathbf{26^{14} \approx 2^{66}}$
- Brute-force search infeasible
- Is the Vigenère cipher secure?
- (Believed secure for many years...)

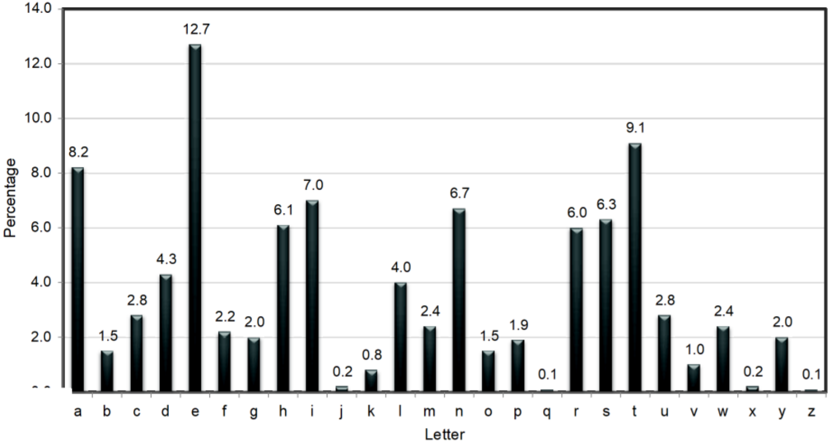
Attacking the Vigenère cipher

Observation

- Every **14**-th character is ”encrypted” using the same shift
- Looking at every **14**-th character is (almost) like looking at ciphertext encrypted with the Shift Cipher
- (Direct brute-force attack still doesn’t work)

```
[v]eqpjiredozxoe[u]alpcmsdjquiqn
[d]nossoscdcusoa[k]jqmxpqrhyycjq
[o]qqodhjcciowie[i]i
```

Using plaintext letter frequencies



Attacking the Vigenère cipher

- Look at every **14**-th character of the ciphertext, starting with the first – call this a "stream"
- Let $\boldsymbol{\alpha}$ be the most common character appearing in this stream
- Most likely $\boldsymbol{\alpha}$ corresponds to the most common plaintext character i.e. $\boldsymbol{e}$
- $\implies$ guess that the first character of the key is $\boldsymbol{\alpha} - \boldsymbol{e}$
- Repeat for all other positions
- Require long ciphertext; prone to errors; can do better...

A better attack 1/2

- Let $p_i : 0 \leq i \leq 25$ denote the frequency of the i-th English letter in normal English plaintext
- Compute $\sum_i {p_i}^2 = 0.065$: constant for English text
- Let q_i denote the **observed** frequency of the i-th English letter within a given **ciphertext stream**
- (q_i is the number of times letter i appears in the ciphertext stream divided by the stream length)
- i of q_i was obtained from letter $i - j$ for key j
- Therefore $q_i \approx p_{i-j}$ or equivalently $q_{i+j} \approx p_i$

A better attack 2/2

- So if the key for the stream is $\boldsymbol{j}$, expect $\boldsymbol{q_{i+j} \approx p_i, \forall i}$
- So expect $\sum_i \boldsymbol{p_i q_{i+j} \approx 0.065}$ for the **right key** $\boldsymbol{j}$
- Test for every value of $\boldsymbol{j}$ to find the right one
- This recovers **the first key character**
- Repeat for the second stream to recover **the second key character**
- Repeat for all streams to recover **the whole key**
- **Recall:** # streams = # key characters

Finding the key length

- The previous attack assumes we know the key length
- What if we don't?
- Of course, can always try the previous attack for all possible key lengths as long as: # **key lengths** $\ll$ # **keys**
- We can do better!

Finding the key length

Observation: correct key length

- For the **correct key length**, the ciphertext frequencies $\{q_i\}$ of a stream will be shifted versions of the $\{p_i\}$
- Recall that $q_i \approx p_{i-j}$ (equivalently $q_{i+j} \approx p_i$), where j is the key (the shift)
- In other words $\{q_i\}$ is a permutation of $\{p_i\}$
- It follows that:

$$\sum_i {q_i}^2 \approx \sum_i {p_i}^2 = 0.065$$

Finding the key length

Observation: incorrect key length

- When using an **incorrect key length**, expect (heuristically) that ciphertext letters are uniform
- For uniform distribution:

$$\sum_i {q_i}^2 = \sum_i \left(\frac{1}{26}\right)^2 = 26\left(\frac{1}{26}\right)^2 = \frac{1}{26} = 0.038$$

Finding the key length

Key length recovery

- For a cadidate key length, the attacker needs to distinguish between $\sum_i q_i^2 = 0.065$ and $\sum_i q_i^2 = 0.038$
- In fact, good enough to find the key length N that maximizes $\sum_i q_i^2$
- (Can verify by looking at other streams)

Attack time?

Time for determining the key length

- Let the key length be at most $\boldsymbol{L}$ i.e. $\mathbf{1} \leq \boldsymbol{N} \leq \boldsymbol{L}$
- Execute at most $\boldsymbol{L}$ trials for the correct key length
  - In each trial compute **26** frequencies $\boldsymbol{q_i} : \mathbf{0} \leq \boldsymbol{i} < \mathbf{26}$
- Total time: $\approx \mathbf{26}\ \boldsymbol{L}$

Attack time?

Time for determining the key

- To deterime the $\boldsymbol{i}$-th character of the key:
  - Execute $\mathbf{26}$ decryptions of the $\boldsymbol{i}$-th stream for each candidate value $\boldsymbol{B}$
    - In each decryption compute $\mathbf{26}$ frequencies $\boldsymbol{q'_i} : \mathbf{0} \leq \boldsymbol{i} \leq \mathbf{25}$
- Total time to recover the $\boldsymbol{i}$-th character: $\approx \mathbf{26^2}$
- Total time to recover all key bytes: $\leq \mathbf{26^2}\boldsymbol{L}$

Time for Brute-force

$$\mathbf{26}^{\boldsymbol{L}}$$

Total attack time vs. brute-force

$$26L + 26^2L \approx 26^2L \ll 26^L$$

Note

- The attack is more reliable as the ciphertext length grows larger
- A Similar attack can be performed on byte-wise Vigenere

Lessons learned

Crypto Design Lesson One (recall)

- The key space must be large enough to make brute-force attacks impractical (cf. Shift Cipher)

Crypto Design Lesson Two

- Large key space is a necessary, but not sufficient condition for a secure encryption scheme (cf. Vigenère Cipher)

But what does *secure* actually mean? (next lecture!)

End

Reference: Section 1.3 of the book