Cosc 412: Cryptography and security
Lecture 2 (15/7/2020)
Symmetric cryptosystems and discrete probability

Michael Albert
michael.albert@otago.ac.nz

This week

- Abstracting the problem
- Symmetric cryptosystems
- Attacks and examples
- Discrete probability

The basic problem

Hello Bob

Alice wishes to send **Bob** a confidential message whose contents may be of interest to a third party, **Eve**.

What resources can Eve devote to the discovery of the contents?

Objective

It should be at least as difficult for Eve to reconstruct the message having intercepted it, as it would be to suborn the process in some other way.

That is, the *message security* should be at least as good as the *general security*.

Messages and keys

Message space

The *message space*, $\mathcal{M}$, is the set of all possible messages. These can be thought of as strings, or just sequences of bits, bytes, or words.

Keys and key space

A *key* is a piece of genuinely private information held by Alice and Bob (but not Eve!) The *key space*, $\mathcal{K}$, is the set of all possible keys.

Symmetric cryptosystems

A *symmetric cryptosystem* (or *symmetric cryptographic protocol*) is a pair of functions:

$$\begin{aligned} E: &\quad \mathcal{K} \times \mathcal{M} \to \mathcal{M} \\ D: &\quad \mathcal{K} \times \mathcal{M} \to \mathcal{M} \end{aligned}$$

such that for all $m \in \mathcal{M}$ and all $k \in \mathcal{K}$,

$$D(k, E(k, m)) = m.$$

That is, if, from a message m, Alice produces a *ciphertext*, $c = E(k, m)$, and sends it to Bob then he can recover it by computing $m = D(k, c)$.

Attack types

We'll consider four broad type of attack:

- *Ciphertext only* Eve has access only to the encrypted message c (or possibly some sequence of encrypted messages).
- *Known plaintext* Eve has access to some pairs (m, c) of previous messages and ciphertexts.
- *Chosen plaintext* Eve can choose certain messages and gain access to their encrypted form.
- *Brute force* What it sounds like.

Caesar cipher

- Take $\mathcal{M}$ to be the space of strings over $\mathcal{A}$, the set of upper case letters, A through Z.
- Think of these as $A = 0$ through $Z = 25$.
- Take $\mathcal{K}$ to be the set of upper case letters, and let k be a particular key.
- E just "adds k" to each letter of the message (wrapping around, i.e., taking a remainder modulo 26).
- D just "subtracts k".

Substitution cipher

- Take $\mathcal{M}$ to be the space of strings of upper case letters, *A* through *Z*.
- Take $\mathcal{K}$ to be the set of permutations of $\mathcal{A}$, and let κ be a particular key.
- *E* just applies κ to each letter of the message.
- *D* just applies the inverse of κ

Vigenère cipher (sixteenth century)

- Take $\mathcal{M}$ to be the space of strings of upper case letters, A through Z.
- Take $\mathcal{K}$ to be the set of strings from $\mathcal{A}$ of some fixed length, n, and let $\mathbf{k} = k_0 k_1 k_2 \ldots k_{n-1}$ be a particular key.
- E is just application of the Caesar ciphers corresponding to the characters of $\mathbf{k}$ sequentially to m, wrapping around back to the beginning of $\mathbf{k}$ when necessary.

Vigenère revisited

- To break the Vigenère cipher it's pretty much sufficient to be able to work out the key length.
- Friedman test:
  - break up the text according to an assumed key length,
  - if correct each block will either represent a sample of letters according to the standard frequency distribution (rotated),
  - if incorrect each block will represent a mixture of two or more such samples (with different rotations) so will be "smoother",
  - try to quantify that smoothness.
- Additional information can be obtained from Kasiski examination which looks for repeated trigrams and uses the fact that gaps between them are likely to be multiples of the key length.

The key insights

- Attacks on classical cryptosystems are based on discovering patterns in the ciphertext that correspond to the structure of the plaintext.
- As computing resources increase these attacks grow stronger and stronger.
- Any cryptosystem which creates such patterns must be deemed to be (potentially) insecure.
- Random text contains no patterns.

Question

How can we create cryptosystems in which the ciphertext is, or appears to be, random and yet still contains the information we desire to transmit?

Discrete probability

To try to understand randomness we need an understanding of *probability*. We're working with discrete data (strings etc.) so *discrete probability* is all we need to know about.

That's good because, in most instances, discrete probability is just about counting!

See notes and examples from lecture.