COMP 102: Excursions in Computer Science

Lecture 15: Information Theory and Cryptography

Instructor: Joelle Pineau (jpineau@cs.mcgill.ca)

Class web page: www.cs.mcgill.ca/~jpineau/comp102

Today

- What is *information* to a computing device?
- Information theory is about measuring information and its properties.
- Cryptography is about hiding information.

Claude Shannon (1916-2011)

- Working at Bell Labs in the 1940's on basic problems in telecommunications. Known as the father of information theory.
- Key idea:

  The fundamental problem of communication is that of reproducing at one point a message selected at another point.

- Information ≠ knowledge
- Information = reduction in uncertainty

Information in numbers

- Case #1: You're about to observe the outcome of a coin flip.
  - How much information is there in this outcome?
- Case #2: You're about to observe the outcome of a die roll.
  - How much information is there in this outcome?
- Which of Case#1 and Case#2 has more uncertainty?
- Which outcome takes more bit to communicate?
- There is more uncertainty about case #2, therefore there is more information in the outcome of case #2.

Information in language

- Read the following phrases:
  1. Cnsdr ths sntnc, cn y rd t?
  2. Consider this sentence, can you read it?

  Which has more uncertainty? Which contains more information?

- English (and all other languages) contain substantial redundancy. Why?
  - Easier to understand. More robust to noise.

The magic number game

- Let me pick a number x between 1 and N. You are allowed to ask questions of the type : "This number of larger than … ?". What is the smallest number of questions that will always identify x ?

- Try this for $N = 2$, $N=30$, $N=1000$.

Measuring information

- Question: Suppose we have a eight-sided die and want to record the outcomes from a series of rolls. How many bits are required to encode each roll?

- Solution:
  - There are 8 possibilities.
  - With n bits we can encode 2^n possibilities.
  - Therefore we need $\log_2 8 = 3$ bits per roll.

Measuring information (2)

- Let E be an event that occurs with probability $Pr(E)$. If we are told that E has occurred, then we have received $I(E) = \log_2 [\, 1 / P(E) \,]$ bits of information.

- If we want to record a series of coin flips, how many bits are required?
  - For each flip: $I(E=\text{heads}) = \log_2 (1 / 0.5) = 1$ bit per flip
  - For k flips: $I(E) = \log_2 (1 / 0.5^k) = k$ bits

    Information is additive!

Entropy

- Consider an information source, called S, that emits symbols from an alphabet $\{s_1, s_2, \ldots, s_k\}$, each with respective probability $\{p_1, p_2, \ldots, p_k\}$.
- What is the expected amount of information emitted by this source?
- This is called the **Entropy**: $H(S) = \Sigma_i p_i I(s_i) = \Sigma_i p_i \log_2 (1/(p_i))$

Entropy in a language

- Consider the English language:
  - Let's say we have 100,000 words, avg. 5.5 characters per word.
  - Words are written with the alphabet $S = \{a, b, \ldots, z, space\}$.
- If we assume each character in S has the same probability:
  - $H(S) = \log_2 (27) = 4.75$ bit / character
- If we use the true probability of characters (which we can calculate by looking at all words in the dictionary):
  - $H(S) = 4.03$ bit / character

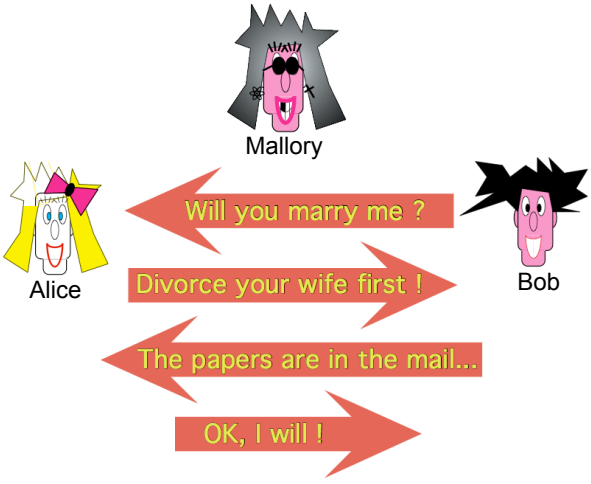
  Note: This assumes independence between characters and words. The true Entropy of English language is much lower than this.
- Playing 20-questions: Remember the magic number game. Now we can do the same with letters, and guess any word!

Cryptography

- Cryptography is the science of hiding information.
- Main problems:
  - Encryption / decryption
  - Authentication
  - Identification

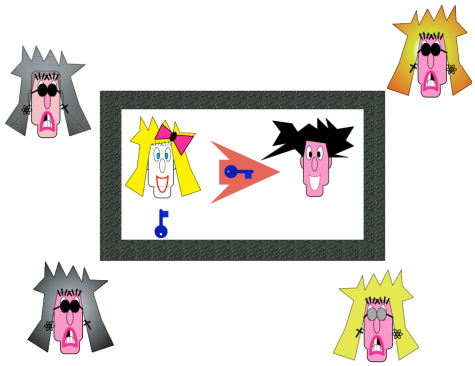
- Main concern is to transmit information in in the presence of an adversary (who is trying to steal your secrets!)

Cryptography



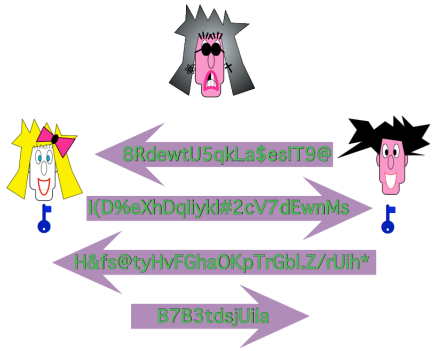
Key distribution

- Objective: to share a secret (or key, or password) in such a way that only the intended recipient receives information about the secret.
- For now, let's assume this can be done securely.



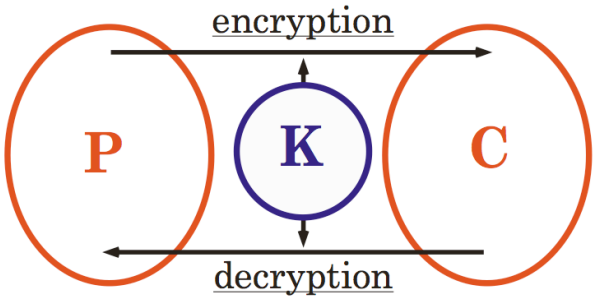
Encryption

- The process of transforming a plaintext message *m* into a ciphertext *c* (the encoded version) and from *c* back to *m*.
- The ciphertext is such that without knowledge of the key *k* required for decryption, no information about *m* is divulged.

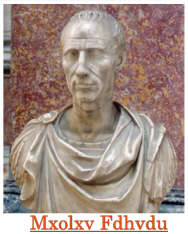


Symmetric encryption

- The same key *k* is used for both encryption and decryption.



Caesar's Cipher



- One of the earliest forms of encryption.
- The key *k* is composed of a number, between 1 and 25.

Mxolxv Fdhvdu

- To encrypt a message, each letter in *m* is shifted forward by *k* letters, so if *k* = 2 then A becomes C, B becomes D, etc.
- To decrypt a message, each letter is shifted backwards by *k* letters.
- Problems?

Problems with Caesar's Cipher

- Known statistics about the plaintext language can be used to easily discover the keys. E.g.
  - A and I are the only one-letter words in the English language.
  - Letters A, E, I, O, and T, are used much more often than other letters.
  - etc.
- This approach to breaking an encryption algorithm is referred to as frequency analysis.

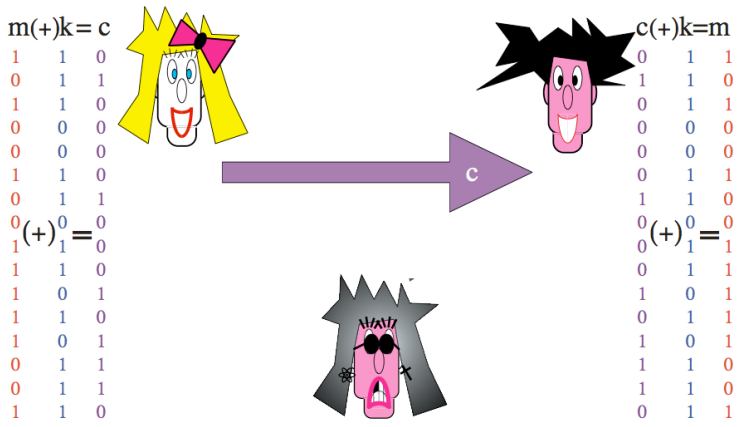
Entropy in cryptography

- Encoded messages that have low entropy are vulnerable to this type of analysis. Much better to produce high entropy encodings.
- What probability distribution over characters produces messages with the highest possible entropy?

  E.g. Consider a generic coin: *Pr(heads) = p ; Pr(tail) = 1-p*

  For what value of *p* are we most uncertain about the outcome?
- How do we choose a better key?

Vernam's one time pad (OTP)

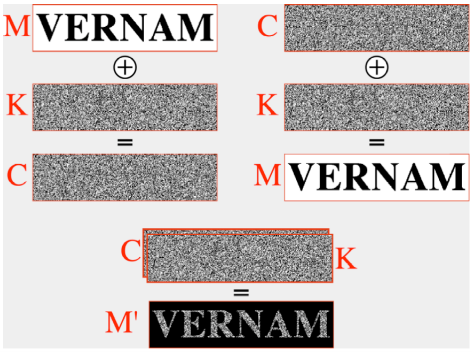


Vernam's one time pad (OTP)

- Basic steps of OTP:
  - Assume the message *m* is encoded as a binary sequence.
  - The key *k* is the same length as the message, and contains a set of random bits.
  - Encryption is performed by XORing each bit of *m* with its corresponding bit in *k* to produce the cipher *c*.
  - To decode, every bit is XORed with the corresponding bit in *k*, and this recovers *m*.
- This is the only scheme to give (provably) perfect security!
  - Why? If *k* is unknown, then given a bit of *c*, the corresponding bit in *m* could be 0 or 1 with equal probability.
- Problems?

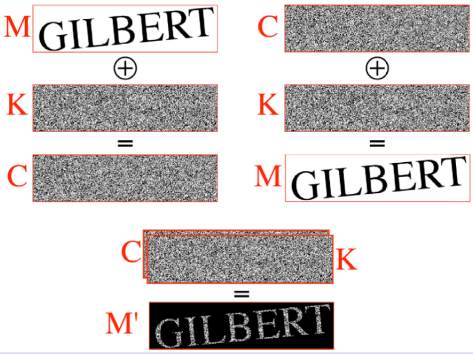
Problem with OTP

- If a key *k* is used to encrypt two messages, *m* and *m'*, and someone sees both *c* and *c'*, then they they are able to recover something about *m* and *m'*.



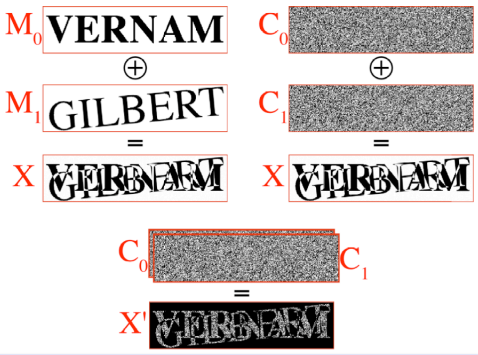
Problem with OTP

- If a key *k* is used to encrypt two messages, *m* and *m'*, and someone sees both *c* and *c'*, then they they are able to recover something about *m* and *m'*.



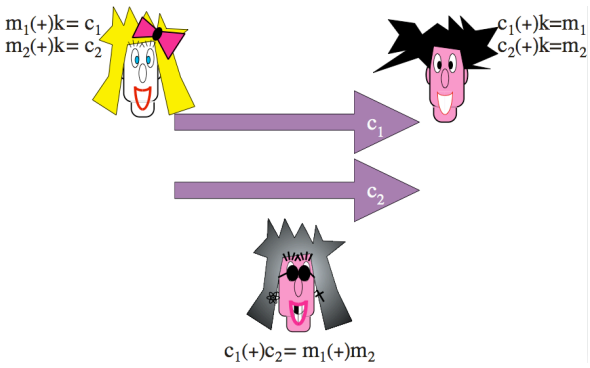
Problem with OTP

- If a key *k* is used to encrypt two messages, *m* and *m'*, and someone sees both *c* and *c'*, then they they are able to recover something about *m* and *m'*.



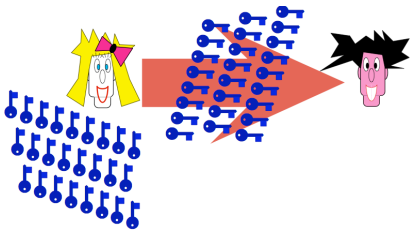
Problem with OTP

- If a key *k* is used to encrypt two messages, *m* and *m'*, and someone sees both *c* and *c'*, then they they are able to recover something about *m* and *m'*.



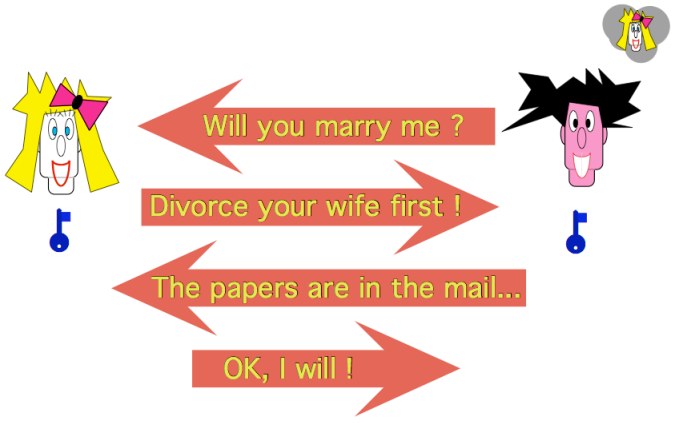
OTP in practice

- Now the big problem becomes how to distribute enough keys securely to facilitate communication!



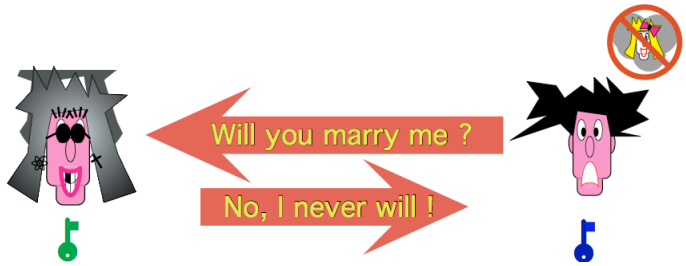
- Modern encryption algorithms are designed so that a key can be reused many times without compromising security.
  - Idea: Turn a short password into an arbitrarily long, random-looking, key that can be XORed with a message.

Authentication

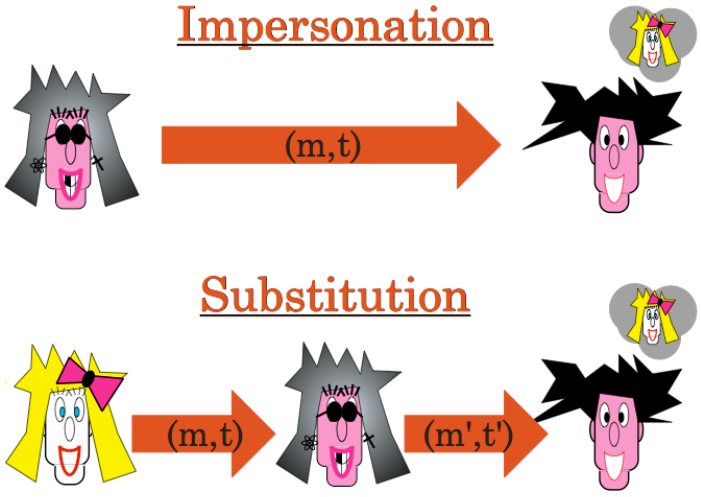


Authentication

- Objective: For a person to be able to establish that a claim about an object, typically that it is authentic, is true.
  - Serves a similar purpose as a *signature*.

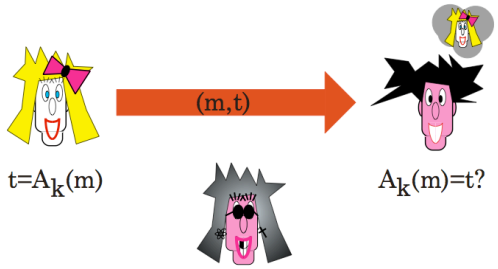


What we want to avoid

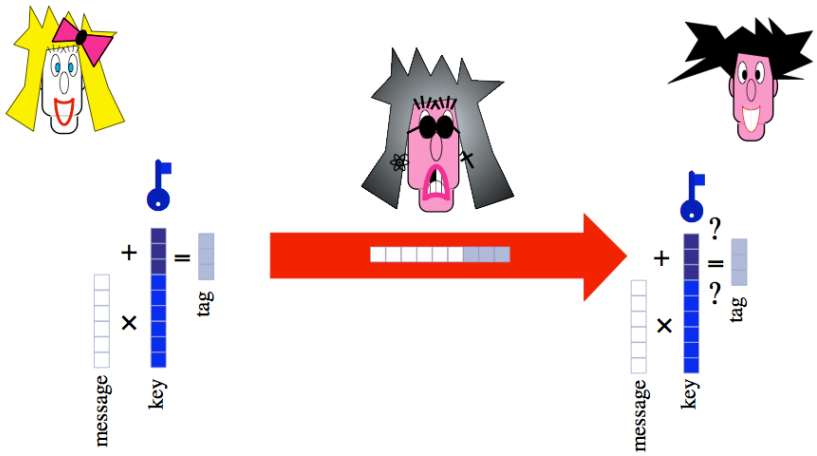


Symmetric authentication

- Using a key *k*, a tag *t* is generated for a message *m* by calling a procedure $A_k(m)$. Anyone who knows *k* can run $A_k(m)$, and assume that the claim is verified if *t* is generated.



One-time authentication



Wegman-Carter authentication

$$t = A_k(m) = am+b \pmod p$$

$$|m| = n,\ |t| = s,\ |k| = n + s$$

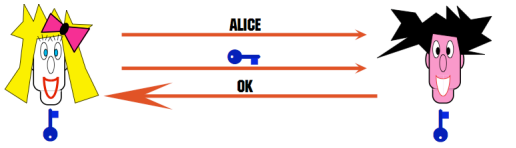
$$\forall m \in M, \forall t \in T$$
$$\Pr(A_k(m)=t) = 1/|T| = 1/2^s$$

$$\forall m \neq m' \in M, \forall t, t' \in T$$
$$\Pr(\ A_k(m')=t' \mid A_k(m)=t\ ) = 1/|T| = 1/2^s$$

Identification

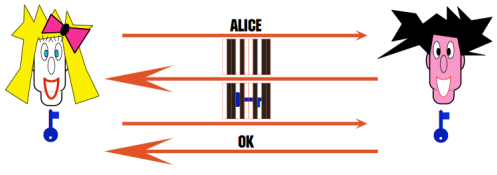
Objective is to prove one's identity. Similar to authentication.

First share a key:



One-time identification:

- Alice identifies herself.
- Bob sends back random message.
- Alice sends back random message with key mixed in.
- Bob confirms this is correct.



Enigma machine

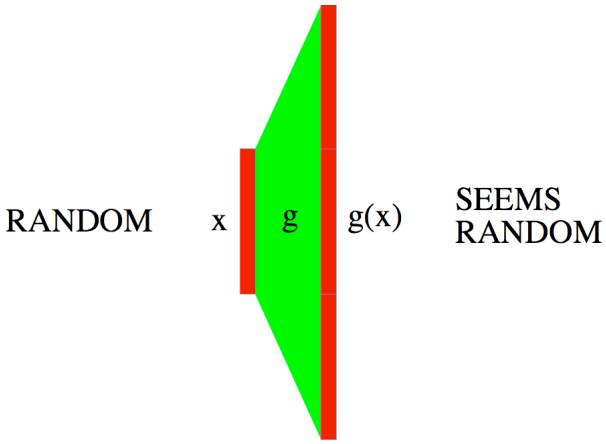
- Electro-mechanical machine used in WWII by the Germans for secret communication.
- Uses a series of rotating wheels to scramble the plain text message.
- A list of daily key settings was printed in a codebook.
- Each key defined the machine's wheel order, wheel start position, etc.

GERMAN ARMY MILITARY ENIGMA. THIS MODEL WAS THE MOST WIDELY USED VERSION OF THE GERMAN WAR-TIME ENIGMAS.

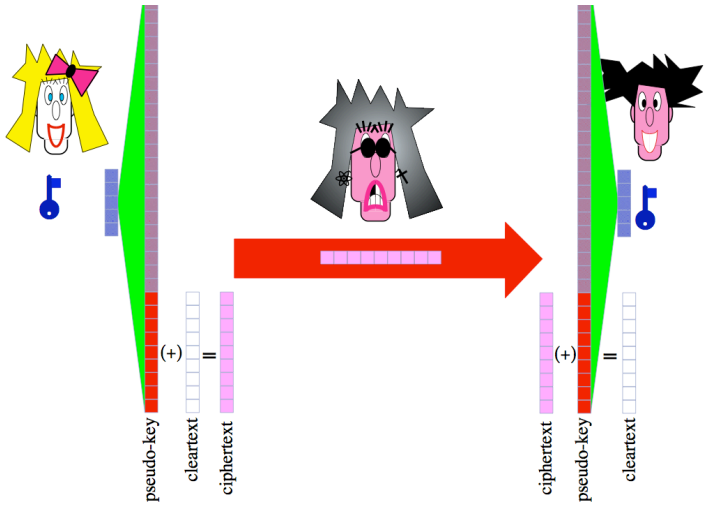
Generating random numbers

- Generating keys requires generating random numbers.
- Where do we get random numbers?
  - Nature: Measure some physical process that is expected to be random. E.g. Variations in the amplitude of atmospheric noise.
  - Computers: Run an algorithm that generates a sequence of apparently random results. E.g. $X_{n+1} = (aX_n + b) \bmod m$
    - These are not truly random. Can only generate m different numbers. Fully depends on the initial seed, X_0.

Pseudo-random bit generator



Encryption with pseudo-random bits



Take-home message

- Basic concepts of information theory: information, entropy.
- Understand the basic components of cryptography: key distribution, encryption, authentication, identification.
- Vernam's one-time pad: how it works, its properties.
- Role of random numbers in cryptography.