

# Computer Science 418

## Introduction, Motivation, and Terminology

Mike Jacobson

Department of Computer Science  
University of Calgary

Week 1

## Outline

- 1 Motivation
- 2 Technicalities
- 3 Overview of Cryptography
- 4 Cryptography and Information Security
- 5 Encryption and Decryption
  - Symmetric Cryptosystems
- 6 Attacks Revisited

## Motivation

### Motivation

*Cryptography* (from the Greek) — ‘hidden writing’

What would you like to see in a secure electronic assignment submission system? Want submission:

- confidential so no one can steal it (confidentiality)
- protected so no one can alter it (data integrity)
- authentic so no one can impersonate creator (entity authentication)
- safe from intrusion on disk (access control)
- safe from denial by instructor or TA (non-repudiation)

This course will work toward solutions for ensuring all of these. Examples of complete systems at end of the course.

## Technicalities

### Resources

Course web page (link from instructor’s home page):

- Course info, assignments, handouts, course schedule, useful links
- Solutions to assignments and certain other documents are only posted on Blackboard
- Blackboard: assignment submission and grade reporting *only*

Resources:

- 3 textbooks recommended: Stinson (3rd ed.), Katz & Lindell, Paar & Pelzl
- other sources on web (see course web pages, in particular the “references” page)

## Evaluation

40%: 4 assignments (3 weeks for each so firm deadlines)

- Some written problems common to CPSC 418 and PMAT 418
- Some for CPSC 418 only (mainly programming problems, can be done by PMAT 418 students for bonus credit)
- Some for PMAT 418 only (mainly mathematical and proof-oriented problems, can be done by CPSC 418 students for bonus credit)
- All work must be done *individually*

20%: midterm exam (Nov. 6, in-class, closed book)

40%: final exam on *all* course material

## Course Content

Rough schedule:

- 7 weeks: Symmetric Cryptography (what is it, what does it do, techniques, attacks)
- 5 weeks: Public Key Cryptography (same)
- 1 week: Cryptography in practice, real-life use examples

Part of two undergraduate concentrations (areas of specialization):

- Concentration in Information Security (part of BSc CPSC)
- Concentration in Cryptography (part of BSc in PMAT)

Both start with

- CPSC 329 Explorations in Information Security
- CPSC/PMAT 418 Introduction to Cryptography

and then go off in different directions. More information: “about 418”

## Basic Terminology

Historically, cryptography is the art of sending messages in secret, or disguised form.

### Definition 1 (encrypt, encipher)

To render a message unintelligible except to the intended recipient.

### Definition 2 (decrypt, decipher)

To transform an encrypted message back into its unencrypted form.

## More Terminology

### Definition 3 (plaintext, cleartext, “in the clear”)

The message or data to be encrypted.

### Definition 4 (ciphertext, cryptogram)

The message after encryption.

### Definition 5 (cipher, cryptosystem)

A particular method of encryption, capable of handling arbitrary messages

## An Old Example

### Example 6 (Caesar Cipher)

Substitute each plaintext letter with the third subsequent letter of the alphabet, wrapping from *Z* to *A*; *i.e.*  $A \rightarrow D$ ,  $B \rightarrow E$ ,  $\dots$ ,  $Z \rightarrow C$  . :

Plaintext: I came, I saw, I conquered.

Ciphertext: L FDPH, L VDZ, L FRQTXHUHG.

Example of a class of ciphers known as *shift ciphers*:

- shift every letter by another letter a fixed position down in the alphabet (with “wrap-around”) at “Z”).

2000 years old: According to Suetonius (“Lives of the Caesars”), Julius Caesar used this cipher during his campaign in Gaul (modern day France) to send encrypted dispatches back to Rome.

## Who Uses Cryptography?

Historic users:

- governments (military, diplomatic service)
- a illicit private uses (secret love letters, conspiracies)

Modern users (since invention of computers):

- everyone! (everyone using a computer, smart phone, credit card, BluRay player, . . .)

Cryptography is ubiquitous! Examples:

- e-commerce, online banking, online purchases, online auctioning (eBay), logging into a computer, using a banking machine, and many more.

Modern cryptography does MUCH more than just hiding messages.

## Information Security

### Definition 7 (information security)

Measures to protect information and information systems from unauthorized access, use, disclosure, disruption, modification or destruction.

Cryptography provides *some* such measures

- important part of complete security systems
- does *not* do it all!

## Security Objectives

Services provided by modern cryptography:

- Data confidentiality (data only readable to legitimate parties)
- Data integrity (data has not been modified)
- Non-repudiation (protection against denial by one of the parties in a communication)
- Authentication (communicating entity is the one claimed)
- Access Control

## Security Mechanisms

Encryption is just one of many *security mechanisms* that achieve one or more of the above security objective.

Cryptographic security mechanisms discussed in this course include:

- Encryption systems — for confidentiality and limited data integrity
- Digital signatures — for data integrity and non-repudiation
- Hash functions, Message Authentication Codes (MACs) — for data integrity and authentication
- Authentication exchange/protocol — for authentication and access control

Cryptography provides many security mechanisms, but not all

- Necessary, but not sufficient for information security (more later!)
- See Anderson “Why cryptosystems fail” (see “external links”).

## Security Attacks

Security mechanisms are designed to detect, prevent, or recover from a *security attack*, i.e. an action that compromises the security of information owned by an organization.

We distinguish between

- *passive* attacks – listening, eavesdropping on information
- *active* attacks – modifying information (for impersonation, replaying messages, changing contents, or denial of service)

Successful cryptographic protocols typically combine several mechanisms to guard against as many different attacks as possible (especially active ones).

## Modern Terminology

### Definition 8

*Cryptography* – the study of mathematical techniques for providing information security services

*Cryptanalysis* – the study of mathematical techniques for attempting to defeat cryptographic security mechanisms

*Cryptology* – combined fields of cryptography and cryptanalysis

*Cryptographic primitive* – tool that represents a cryptographic security mechanism

*Cryptographic protocol* – an algorithm (sequence of steps) to be undertaken by two or more entities to achieve a specific security objective

Will cover primitives/protocols for all security mechanisms listed above.

Great reference: *Handbook of Applied Cryptography* (see “external links”)

## Terminology

### Definition 9

*Message space*  $\mathcal{M}$  – set of all possible plaintext messages

*Ciphertext space*  $\mathcal{C}$  – set of all possible encrypted messages

*Key space*  $\mathcal{K}$  – the finite set of possible keys

*Encryption transformation* – a left invertible map  $E_K : \mathcal{M} \rightarrow \mathcal{C}$ , indexed by some key  $k \in \mathcal{K}$

*Decryption transformations* – the left inverse map  $D_K$  of  $E_K$ , so  $D_K(E_K(M)) = M$  for all plaintexts  $M \in \mathcal{M}$ .

**Note:**  $D_K(E_K(M)) = M$  implies that  $D_K \circ E_K = I$  is the *identity transformation* on  $\mathcal{M}$ .

**Note:** The fact that  $E_K$  is left-invertible is equivalent to  $E_K$  is an *injective* (i.e. one-to-one) map.

## The Idea of Encryption and Decryption

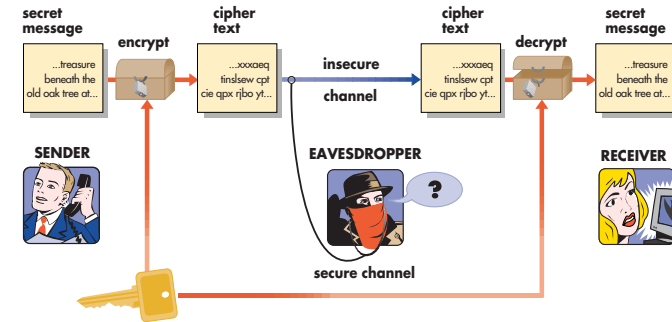
Gilles Brassard, a professor at the Université de Montréal and the inventor of quantum cryptography, created the protagonists *Alice* and *Bob*.

- Since then, many more characters have joined the crypto game; most notably *Eve*.

Idea:

- A transmitter (Bob) generates a plaintext  $M \in \mathcal{M}$ , to be communicated to a legitimate receiver (Alice) over an insecure channel.
- To prevent an eavesdropper (Eve) from learning the contents of  $M$ , Bob chooses a key  $K \in \mathcal{K}$  and encrypts  $M$  with  $E_K$  to produce the ciphertext  $C = E_K(M)$ .
- $C$  is sent along the insecure channel. When Alice obtains  $C$ , she deciphers it by applying  $D_K$  to  $C$  to obtain  $M = D_K(C)$ .

## Conventional Cryptosystem



## Issues

Encryption functions are our first example of a cryptographic primitive

- could easily formalize the above description to create a cryptographic protocol.

Note that Bob must somehow communicate the secret key to Alice without Eve obtaining it, *i.e.* over a secure channel (more on that later).

The assumption is that the workings of  $E_K$  and  $D_K$  are not secret, but  $K$  is secret. So only Bob can decrypt, but no one else can.

## Example: Shift Cipher

Description:

- $\mathcal{M} = \mathcal{C} = \{A, B, \dots, Z\}$ .
- Keys represent shifts by a position between 0 and 25.
- Encryption is a forward circular shift of a plaintext letter by  $K$
- Decryption is the corresponding backward circular shift of a ciphertext letter by  $K$ .

## Example, cont.

More formally, first assign each letter a numerical equivalent as follows.

0	1	2	3	...	25
a	b	c	d	...	z

With that, we have  $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$  (the integers modulo 26).

Encryption:  $E_K(M) \equiv M + K \pmod{26}$  (remainder between 0 and 25).

Decryption:  $D_K(C) \equiv C - K \pmod{26}$  (remainder between 0 and 25).

For the Caesar cipher,  $K = 3$ .

## Issues with the Shift Cipher

Main problem: very small key space ( $|\mathcal{K}| = 26$ )

- Easily falls to a “brute force attack” by simply trying each key in turn. (assumes that you know that a shift cipher is used)

**Note:** How small is “small?”

- With modern technology, one tenth of a billion billion billion =  $10^{17} \approx 2^{56}$  is small (DES has  $|\mathcal{K}| = 2^{56}$ ).
- Clearly,  $26 < 2^{56}$ . ( $2^{80}$  questionable!)

## Symmetric Cryptosystems

We are now in a position to formally a cryptosystem.

## Definition 10 (Symmetric Cryptosystem)

A single-parameter family  $\{E_K\}_{K \in \mathcal{K}}$  of injective transformations

$$E_K : \mathcal{M} \rightarrow \mathcal{C}$$

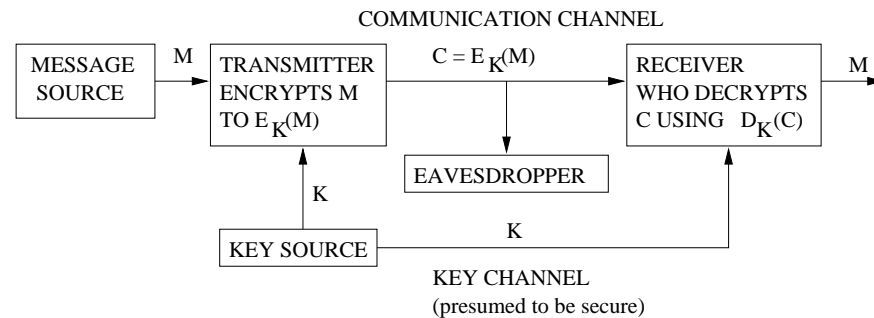
$$M \mapsto E_K(M) = C \quad (M \in \mathcal{M}, C \in \mathcal{C}),$$

where  $E_K$  acts on a *message-space*  $\mathcal{M}$  and injects it into a *cipher-space*  $\mathcal{C}$ .

- The parameter or key  $K$  is selected from the *key space*  $\mathcal{K}$ .
- For any  $K \in \mathcal{K}$ , the left inverse of  $E_K$  is denoted  $D_K$ .

## Schematic of a Symmetric Cryptosystem

AKA *conventional* or *private key* cryptosystems.



## Key Channel

In order for the encryption to be secure, *key channels* must be absolutely secure, as must the channel from the source to the transmitter.

In the real world, this usually means expensive.

For example, the keys to the Moscow-Washington hotline are transmitted by means of highly paid couriers, who fly there and back every week.

## Goals of an Attacker

We can now refine our notions of attacks on cryptosystems

Goals of an attacker:

- Deduce the key or portions thereof
- Deduce a one or more plaintexts or portions thereof
- Modify a message
- Replay a message
- Impersonate (*i.e.* masquerade as) another entity

The first two are passive attacks, the last three active attacks.

## But If We Already Have a Secure Channel...?

It would be nice to dispense with the key channel. Why bother encrypting when we have a secure channel already?

- *Time-shifting, convenience* – you have access to a secure channel now, but would like to use it later, when the channel may not be available.
- *Speed, bandwidth* – the secure channel may be slow or of a limited bit rate.
- *Cost* – the secure channel may be expensive; *e.g.* hand-delivered by courier.
- *Feasibility* – the secure channel may be impractical; *e.g.* Alice and Bob meet in person before securely communicating.

## Types of Attacks on Cryptosystems

Depends on what adversary has available and what he/she can do.

- *Ciphertext Only Attack* (COA) – adversary has only ciphertext, but no plaintext.
- *Known Plaintext Attack* (KPA) – adversary has some plaintext and corresponding ciphertext.
- *Chosen Plaintext Attack* (CPA) – adversary has some plaintext of his choosing and the corresponding ciphertext.

## Types of Attacks, cont.

- *Adaptive CPA* – adversary's choice of plaintext may depend on ciphertexts received from previous requests.
- *Chosen Ciphertext Attack (CCA)* – adversary chooses some ciphertext and is then given the corresponding plaintext. He is not allowed to choose the ciphertext he wishes to decrypt.
- *Adaptive CCA (CCA2)* – adversary's choice of ciphertext may depend on plaintexts received from previous requests

COA and the known text attacks are passive; the chosen text attacks and their adaptive versions are active.

## More on Attacks

**Note:** A good/secure cryptosystem should be secure against adaptive CCA's (as strong as possible)

Some attacks that cryptography cannot protect against:

- *Side Channel Attacks* – adversary exploits some physical aspect of the cryptosystem implementation to extract the key (power/timing/radiation analysis)
- *Clandestine Attacks (AKA Rubber Hose Cryptography)* – adversary bribes, blackmails, threatens, steals, or beats the key out of the recipient

## Notions of Security

### Definition 11 (Kerckhoff's Principle)

The security of a cryptosystem should depend entirely upon knowledge of the key, not of the method.

- From "La Cryptographie Militaire" (1883), one of the 1st scientific treatments of cryptography.
- This implies in particular that a cipher should be completely published and still be secure (against its own designer and everyone else).

So what constitutes a *secure* cryptosystem? We saw that a good system should be secure against adaptive CCA's. What does "secure" mean? There are different notions of security.

## Measures of Security

Listed from strongest to weakest:

- *Unconditional Security* – can an adversary with unlimited computing power defeat the system?
- *Provable Security* – breaking the system can be reduced (mathematically) to another, supposedly difficult problem; e.g. integer factorization.
- *Computational Security* – does the perceived amount of computing power necessary to break the system (using the best known method) exceed (by a comfortable margin) the available computing power of the attacker?
- *Ad-hoc Security* – security is "proved" via a series of convincing arguments that every successful attack is impractical.



## Remarks

Computational security often used in conjunction with provable security

- Eg. a typical security claim might read something like “a cryptosystem is provably secure against an adaptive CCA assuming integer factorization is hard”

Provable security does *not* mean that a cryptosystem is *proved* secure!

- Proofs typically only reduce to another problem (which could eventually be solved)
- Proofs assume specific adversarial capabilities and attacks (eg. adaptive CCA)