TK2100: Informasjonssikkerhet
Lesson 01: Introduction

Dr. Andrea Arcuri
Westerdals Oslo ACT
University of Luxembourg

About Me

Dr. Andrea Arcuri

[simula . research laboratory]
- by thinking constantly about it

Goals

- Learn the *basics* of Information Security
- Cryptography and secure communications
- Operating Systems
- Malwares
- Network Security
- Web Security

Course Info

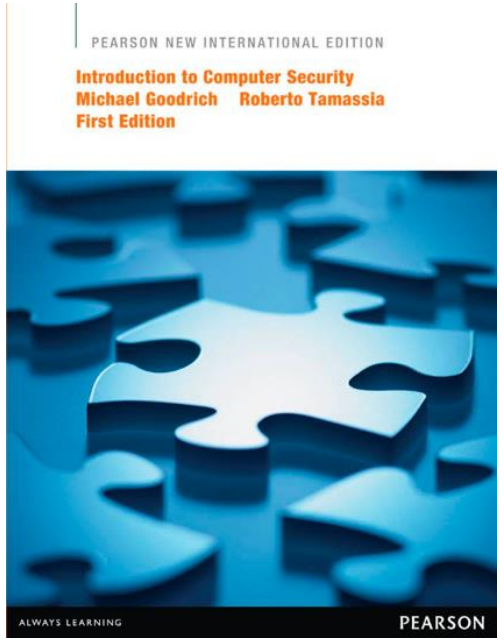
- 12 lessons, once a week
- Check TimeEdit for possible changes of time and rooms
- 4-hour lectures
  - Between 1 and 2 hours of teaching
  - Remaining time is for exercises and questions
    - See TimeEdit for the rooms, which might vary each week
- Exam: 3 hour written exam
  - Note: as teaching this course for first time, and updating it compared to previous years, haven’t decided yet the details of the exam

Theory vs Practice

- This course starts with more focus on the “theoretical” side
  - as you are 1st year student still learning how to code…
  - eg, complex topic, and can’t really ask you to write a malware when you still have problems writing a “for-loop”…
- Practice on using some existing tools, not how to integrate them in existing applications
  - for that, you ll need to wait for Enterprise 1-2 (for Programmering students)
- Last part of the course on Web Security will have more focus on coding compared to the beginning of the course

Course Material

- Its Learning page for TK2100
- Git repository (more on this later in the slides)
  - https://github.com/arcuri82/tk2100
- Book: *Introduction to Computer Security: International Edition*
  - Good book which you will need also in the future
  - **STRONGLY** recommend you to get a copy



Contact

- Do **NOT** send me private emails
  - You are around 200 students…
- If you have questions outside class, post them on the "*Discussion forum*" on the Its Learning page of this course

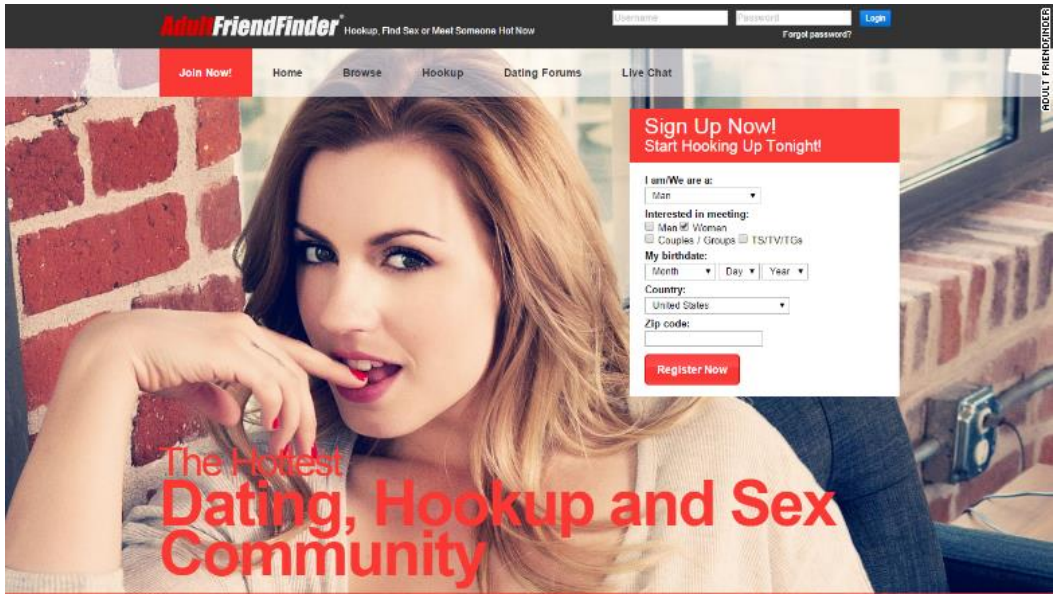
Why studying Information Security???

2013-2014: YAHOO!

- 3 **B**illions accounts compromised
  - yes, it is **B**s, not a **M**illions
- Stealing information is bad, especially if *credit card* numbers
- But what is another major issue? People *re-use same password* for different sites, eg Gmail, Facebook, etc.
- Yahoo had easy to crack hashed passwords (eg, MD5)
- *How many of you use a different password for each different service???*

2016: Adult Friend Finder

- Web site for “hook-ups”
- 412 million accounts were compromised
- Users got blackmailed
  - eg, people cheating on spouses
  - some even committed suicide…



2014: eBay

- 145 million accounts compromised
- But fortunately credit card numbers were stored in different databases

2017: Equifax

- 143 million consumer info breached
- Personal information
  - including Social Security Numbers, birth dates, addresses, and in some cases drivers' license numbers
- 209,000 consumers also had their credit cards exposed

2011: Sony's PlayStation Network

- 77 million accounts hacked
- 12 million unencrypted credit card numbers got stolen
- Site down for a month
- Around $200 million in losses

And the list goes on…

- Could go on all days with examples of compromised web sites where user info was stolen…
- But stealing of info (eg credit cards and passwords) is not the only thing you should be concerned about security…

Web Defacement

- Hack into a website and change its content
- Religious, governments and corporations are typical targets
- Usually for political reasons, or bored “kids”…

Malware

- **Mal**icious Soft**ware**
- Code that execute on your machine could do anything
  - Allow remote control
  - Log keystrokes to steal passwords
  - Etc.
- Usually received via emails or downloads from shady websites…
- Virus: can replicate themselves, and spread to other users, eg via email

Ransom-ware

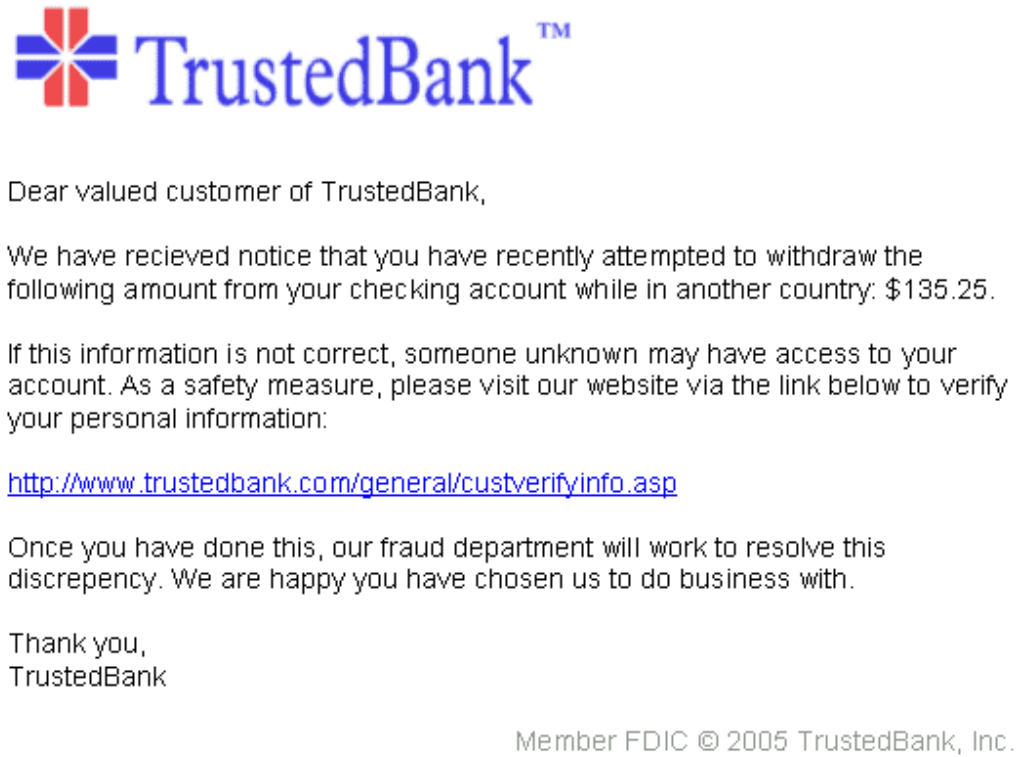
- A special kind of malware
- It encrypts your hard drive, and ask *ransom* to decrypt it
  - usually asked payment in BitCoins
- 2017: 16 hospitals hit in UK
  - hospitals are typical targets, because have critical data of patients on life-or-death conditions

DDOS

- **D**istributed **D**enial **o**f **S**ervice
- Overload servers with “time-consuming” requests
- So many requests that web site goes “down”, as not enough resources to establish new connections for regular users
- Malware: infected machines can work as “bots” and take part in these DDOS, ie **d**istributed

Phishing

- Obtain sensitive data by by disguising as a trustworthy entity
- Typically email with link to a known site (eg your bank), but such link go to a malicious one looking same as original



TrustedBank™

Dear valued customer of TrustedBank,

We have recieved notice that you have recently attempted to withdraw the following amount from your checking account while in another country: $135.25.

If this information is not correct, someone unknown may have access to your account. As a safety measure, please visit our website via the link below to verify your personal information:

http://www.trustedbank.com/general/custverifyinfo.asp

Once you have done this, our fraud department will work to resolve this discrepency. We are happy you have chosen us to do business with.

Thank you,
TrustedBank

Member FDIC © 2005 TrustedBank, Inc.

HTML Injection

- If accessing a web site is un-encrypted (ie, HTTP instead of HTTP**S**), any entity between you and the server can alter the data
- Eg, accessing internet via WiFi in an hotel or restaurant
- Why altering a HTML page?
  - eg, add advertisement or crypto-miners

The Enemy

- Not only typical organized crime
- Foreign government organizations
  - NSA (USA), and equivalents in Russia, China, UK, etc
  - Being “allied” nations does not prevent their **criminal** activities

Security Foundations

What is Information Security?

- A system, application, or protocol is only safe in relation to:
  - Predefined, desired properties
  - An opponent with specified abilities and characteristics
- Example: Access to a Windows machine is not secure for anyone who has physical access to it and can boot it.

C.I.A. Concepts

CIA: Confidentiality

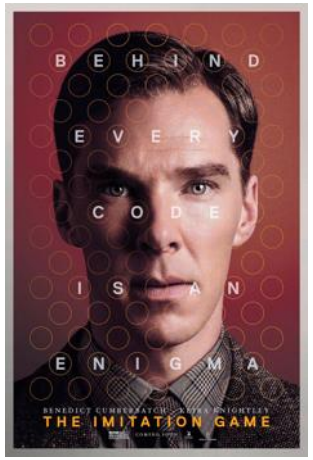
- Avoidance of unauthorized disclosure of information
- Authorized users should be able to read the data…
- … whereas unauthorized should not
- Typically trying to keep data “secret”

Cyphers during Roman Empire

- Communications to generals in the army should be kept secret
- Even if messages get caught by the enemy
- Substitution cyphers, mapping for each letter, eg D -> A, E -> B, etc.
- Trivial to break nowadays
- But shows security was important already 2000 years ago…

WW2: Enigma

- Used by Nazis to make their messages secret
- The “breaking” of this code was a major development for the victory of the allied forces
- Good movie: “*The Imitation Game*”

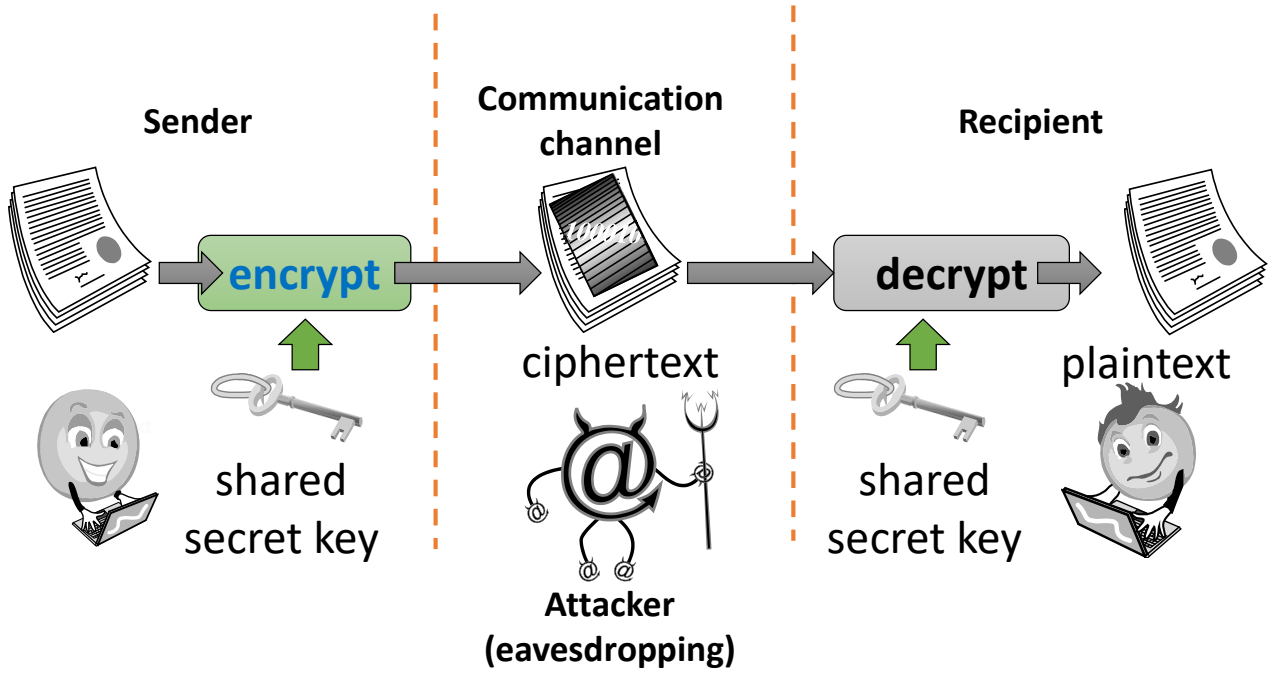


Confidentiality Tools

1. Encryption
2. Access Control
3. Authentication
4. Authorization
5. Physical Security

Confidentiality: Tool 1

- **Encryption**: The conversion of information by secret (encryption key) so that the encrypted message (ciphertext) can only be read / seen with the help of another secret (decryption key)
  - Encryption and decryption keys may be the same or different



Confidentiality: Tool 2

- **Access Control (AC)**
- Rules and policies that restrict access to confidential information
- Applies to both individuals and computer systems
- Can be controlled by identity (name), MAC address, or role (CEO, employee, employee PC, system administrator, ...)

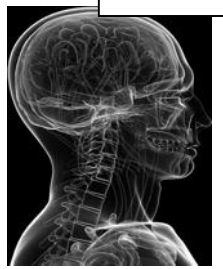
Confidentiality: Tool 3

- **Authentication**
- Determine the identity or role of someone
- Can be done in several ways but is usually based on a combination of:
  - Something one person has: smart card, cell phone, HW key ...
  - Something a person knows: password
  - Something a person is: fingerprints, irises.

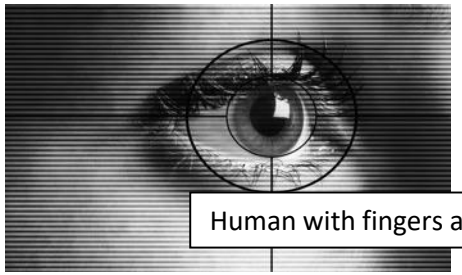
Radio token with secret key

Something you have

passord=uclb()w1V
mor=Godhjerta
årstall=1984



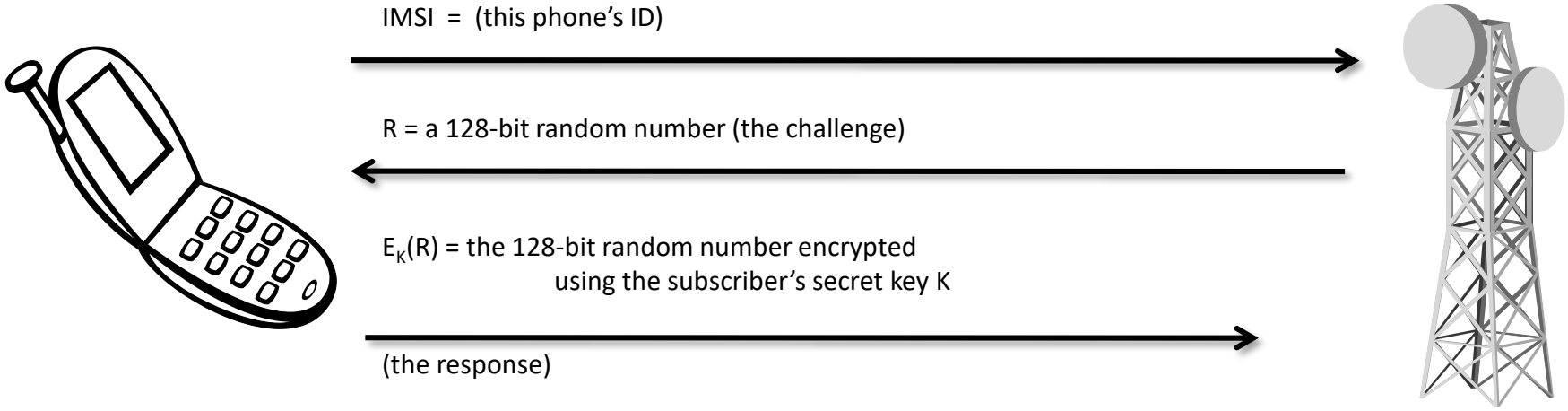
Something you know



Human with fingers and eyes

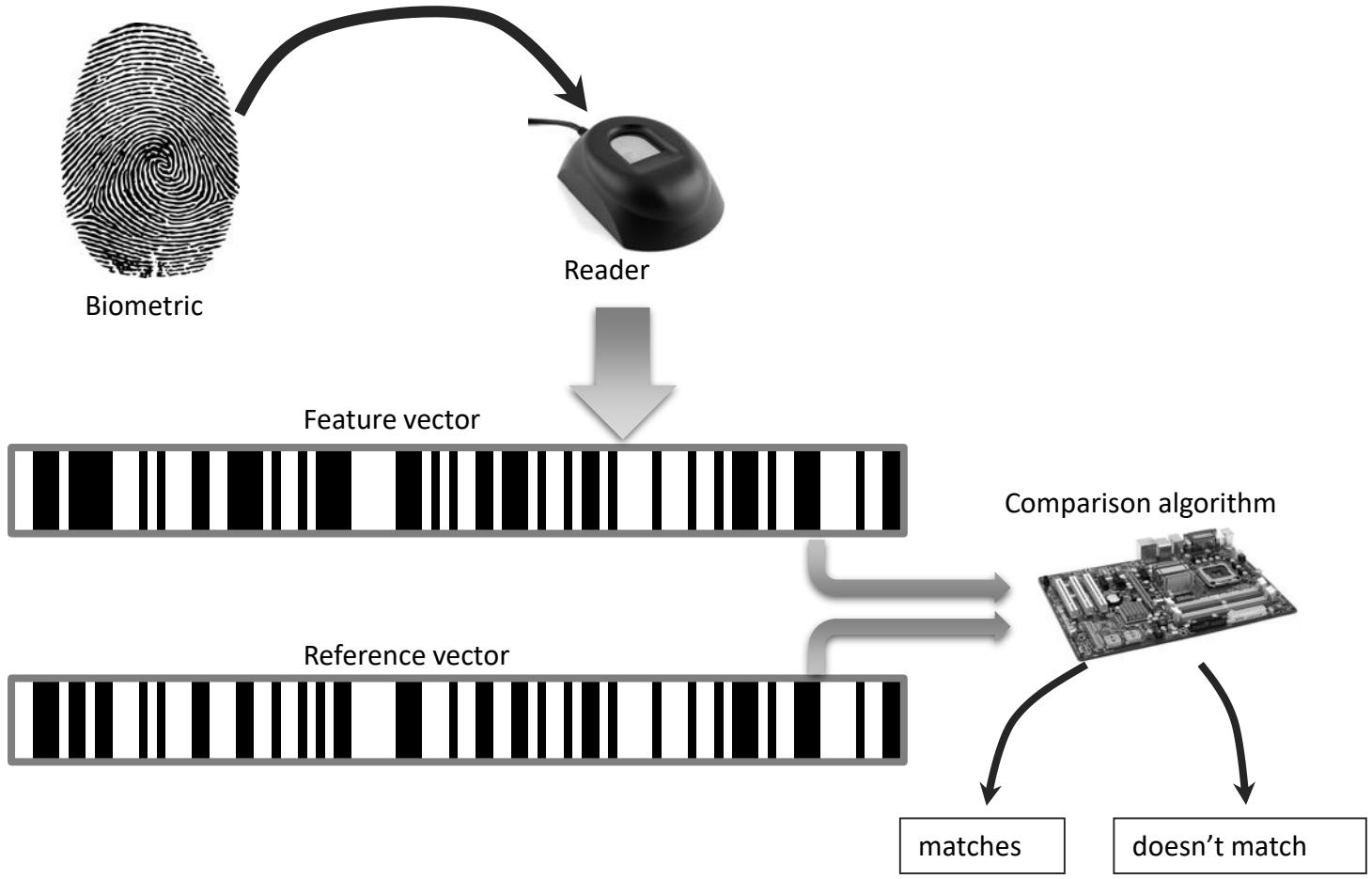
Something you are

Example: 2-phase authentication with Mobiles



- Eg, based on the phone's hardware id, IMSI (International Mobile Subscriber Identity)

Example: Fingerprints



- Eg, used to activate mobile without pin code

Confidentiality: Tool 4

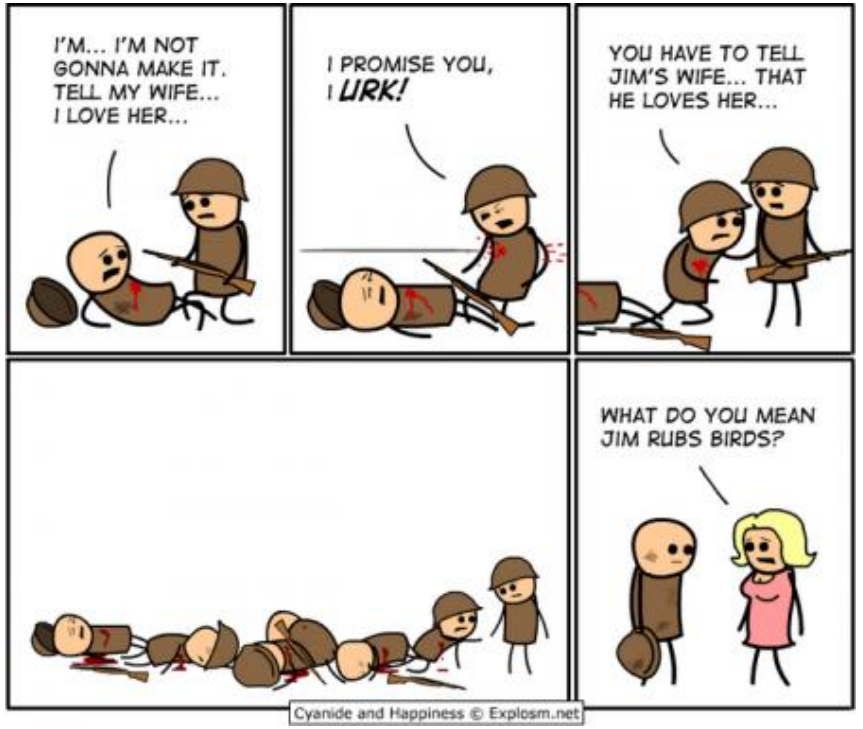
- **Authorization**
- Once *authenticated*, determine which resources a person / system will have access to, based on the *access policies*
- Should prevent the attacker from tricking the system to allow him access to protected resources.

Confidentiality: Tool 5

- **Physical Security**
- Establish physical barriers that restrict access to protected data resources.
- eg, locks on doors
- eg, place servers in rooms without windows eg, soundproofing materials
- eg, Faraday cages (radio wave isolation)
- etc.

C.I.A.: Integrity

- Information not *altered* in unauthorized ways
- Alterations in data are not always malicious, eg *Telephone Game*



Integrity Tools

- **Confidentiality**: authentication/authorization to avoid malicious actors altering the data
- **Backups**: periodic archiving of data. If data compromised, can be restored
- **Checksums**: mathematical mapping from whole data to a single number. If data is altered, mapped number might change, and so detect the alteration
- **Data correcting codes**: redundancy systems in which small changes can be automatically detected and corrected

C.I.**A.**: Availability

- Information should be accessible in timely fashion by those authorized
- Most secure computer is one that it is *not connected* to internet, *switched off* and *unplugged*…
- … but data in it would not really be so *accessible*…

A.A.A. Concepts

Authenticity

Anonymity

Assurance

Assurance

- Assurance is about how *trust* is established and managed in computer systems
  - very hard to quantify
- Trust management involves:
  - *Policies*: specifies expected behavior from people and systems
    - eg, iTunes specifies how users access and can copy / share intellectual property
    - eg, Facebook's rules on what is acceptable behavior and what can be published
  - *Permissions*: describes what behaviors are permissible / allowed for those who use the system
    - eg, iTunes permissions/limitations on copying movies / songs.
  - *Protections*: describes which mechanisms are used to enforce the policies and permissions
    - eg, DRM for copy protection

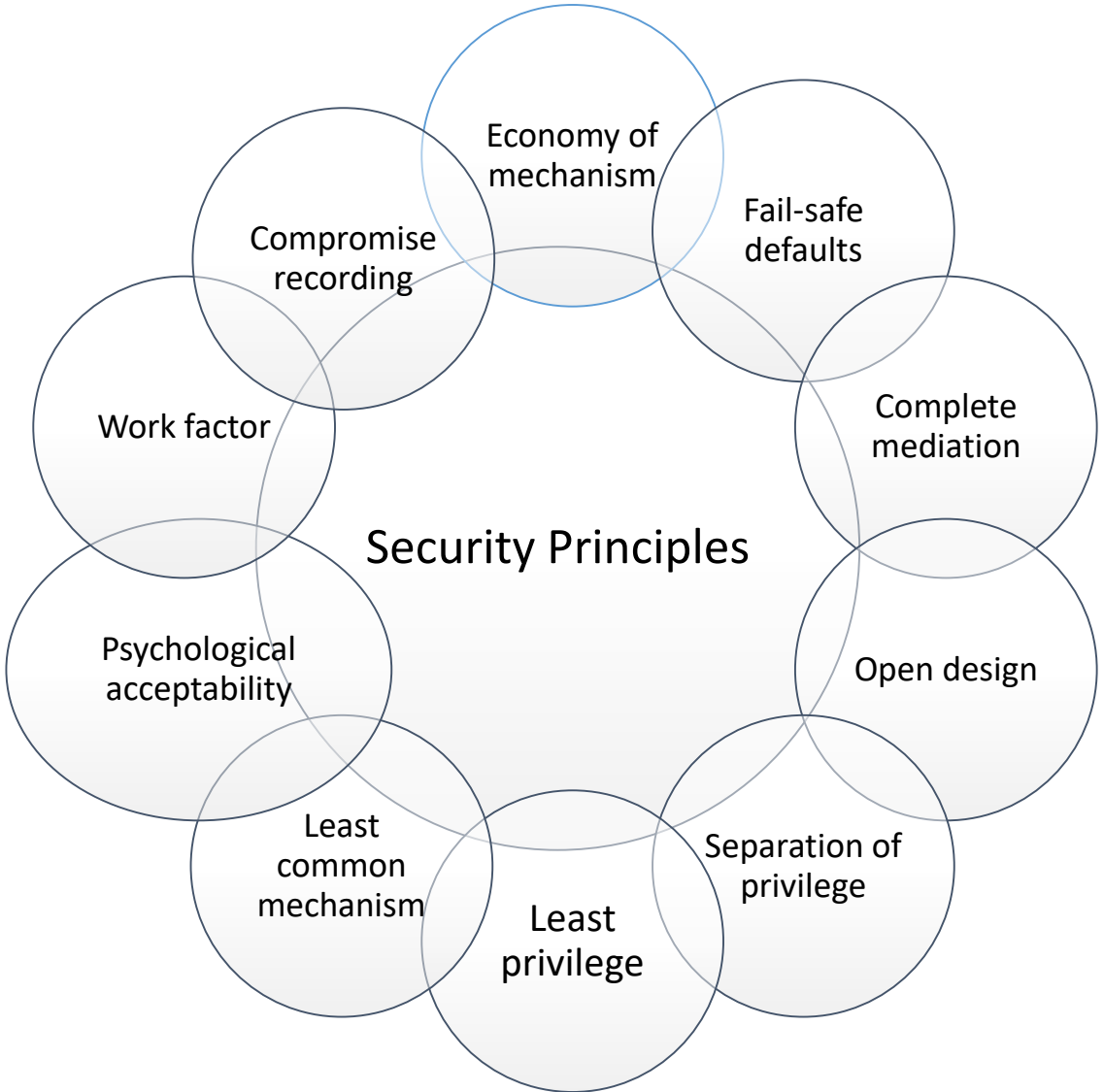
Authenticity

- Ability to determine whether statements, policies and permissions issued by a person / system are genuine (not forged)
- Primary mechanism: *digital signatures*
  - Cryptographic computations that allow people / systems to commit to the authenticity of their documents in a unique way that achieves non-repudation

Anonymity

- Property that certain records or transactions are not to be attributable to any individual
- Corporations want your history and personal info
  - eg, to sell targeted ads
  - eg, to avoid providing insurance to high-risk individuals
- Government agencies can track your online activities
  - see **Tor** Browser

Ten Security Principles



1: Economy of mechanism

- As *simple* design and implementation of security measures as possible
- Applies to all software development, not just security
- Particularly important for security reasons
  - Easier to find errors and prevent critical bugs
- Note: used solutions can be very *complex* (eg, cryptography), but should use existing libraries, not develop your own

2: Fail-safe defaults

- Default settings in applications and systems should be *conservative*
  - eg., a newly created user should have only limited privileges
- Privileges should be explicitly granted, and not be on by default
  - eg., execute code downloaded from internet
- Problem: many applications favors *usability* over security

3: Complete mediation

- Every access to a resource should be checked for compliance
- Possible performance issues, and usability concerns
- Can have timeouts
  - eg, once logged in, allow operations for 15 minutes before asking for login/password again

4: Open Design

- Security algorithms should be *open,* and not secret
- Open implementations lead more people reviewing the code and find issues
- What kept *secret* should be the *keys* used in these algorithms, not the algorithms themselves
- Opposite of “Security by Obscurity”, which traditionally has been very poor

5: Separation of privilege

- Multiple conditions required for access
- Eg, instead of permissions to access all resources, can have permissions for X that are different than for Y
- Separation of components: a component that can access X, should not access unrelated Y. If compromised, only X would be accessed

6: Least privilege

- Users should have only the *minimum* privileges to do the operations s/he needs
- If user/application does not need to work on X, should not have access to X
  - eg, usually a mobile app game should not need to send SMS
- Reason: if a machine is comprised, only the minimum amount of resources can be altered/exploited

7: Least common mechanism

- Resource sharing should be minimized
- Shared resources is a way in which information can be transferred
- Problem when resource sharing involves confidential information by mistake

8: Psychological acceptability

- User interfaces (UI) should be well designed
- Confusing UI shouldn’t lead to misconfigure security settings
- If security of an application is hard to configure, users will not use it
  - eg, email encryption

9: Work factor

- Choice of security measures should take into account the type of resources
- Eg, an online bank needs more security than a web forum

10: Compromise recording

- Sometimes more cost-effective to record details of attack than using more complex techniques to prevent it
- Tradeoff: might not be able (or be too expensive) to prevent all attacks, but at least should know when one happens

Steganography

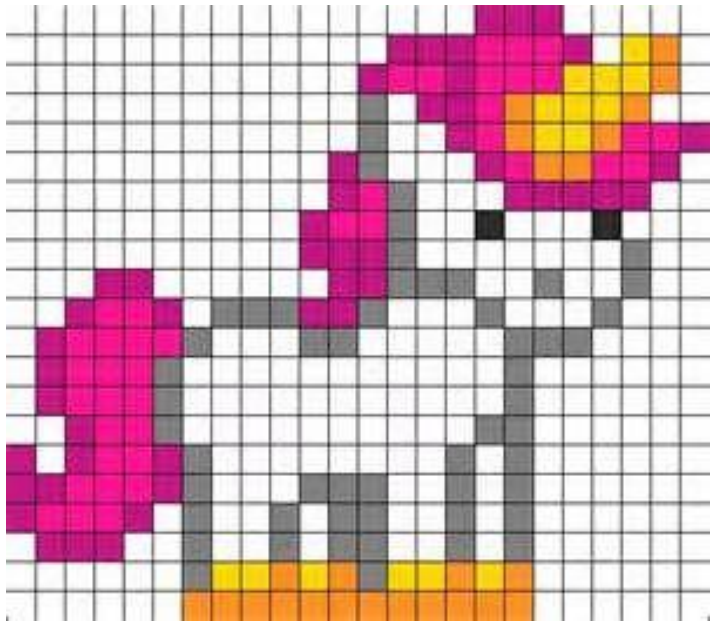
What is the difference between these 2 pictures?

Hidden Message

- The first is a picture of a cute cat
- The second is a picture of a cute cat, but with the text “A secret message” hidden in it using steganography
- But where is the text hidden???

Image formats

- There are many image formats: JPEG, PNG, BMP, etc.
- For simplicity, let's just consider an image as a bitmap, ie a composition of pixels, where each pixel has a color
- The color of each pixel is represented with a **number**
- Image file is a sequence of bits 0/1, like any other file



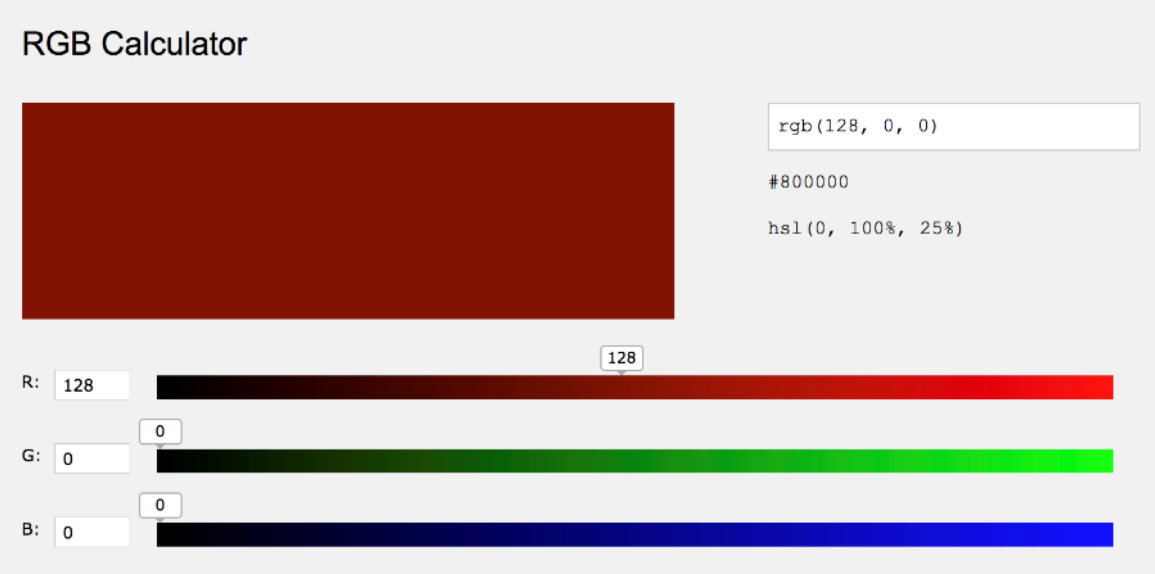
RGB: Red, Green and Blue

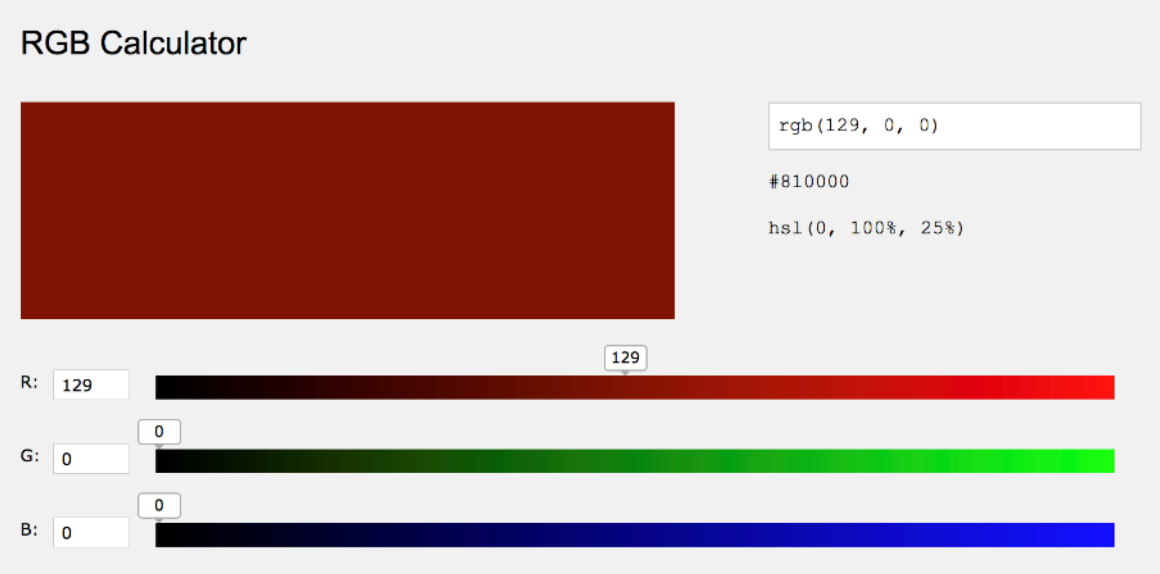
- A color can be represented with a composition of Red, Green and Blue
- Each base color has an *intensity*, eg from 0 to 255
- Each color can be represented has a composition of Red, Green and Blue
- Black: (0,0,0) no red, no green, no blue
- Red: (255, 0, 0) full red, no green, no blue
- Yellow: (255, 255, 0) full red, full green, no blue
- Grey: (127,127,127) all 3 having same value
- White: (255, 255, 255) all full

RGB: Cont.

- If each intensity has 256 different values, can represent it with 8 bits, 2^8=256
- Each RGB pixel would have 3*8=24 bits
- So 2^24=16,777,216 possible different colors
- A 1000x1000 pixel image would be 24M
  - Note: most image formats are *compressed* and would need less space

Can you see the difference?





Small changes in color intensity might not be detectable with your eyes

Image bits 0/1

- Each pixel is going to be 24 bits
- Use 23 bits for color, and 1 bit for your data
- 23 bits: 8 millions instead of 16 millions
- Image reader would still read 24 bits per pixel
- 23+1 bits should be similar (at times the same) color to the original 24 bit one
  - Human eye would likely not notice the difference
- If using ASCII with 7 bits per letter, can have one character per pixel
- “A secret message” is 16 chars, so can be hidden in a 16*7=112 pixel image

Steganography Tools

- Hide messages/images/videos inside other messages/images/videos
- As you are 1st year student, we will not see the code of how to do it
- SilentEye is one existing tool
  - http://silenteye.v1kings.io/

Git and GitHub

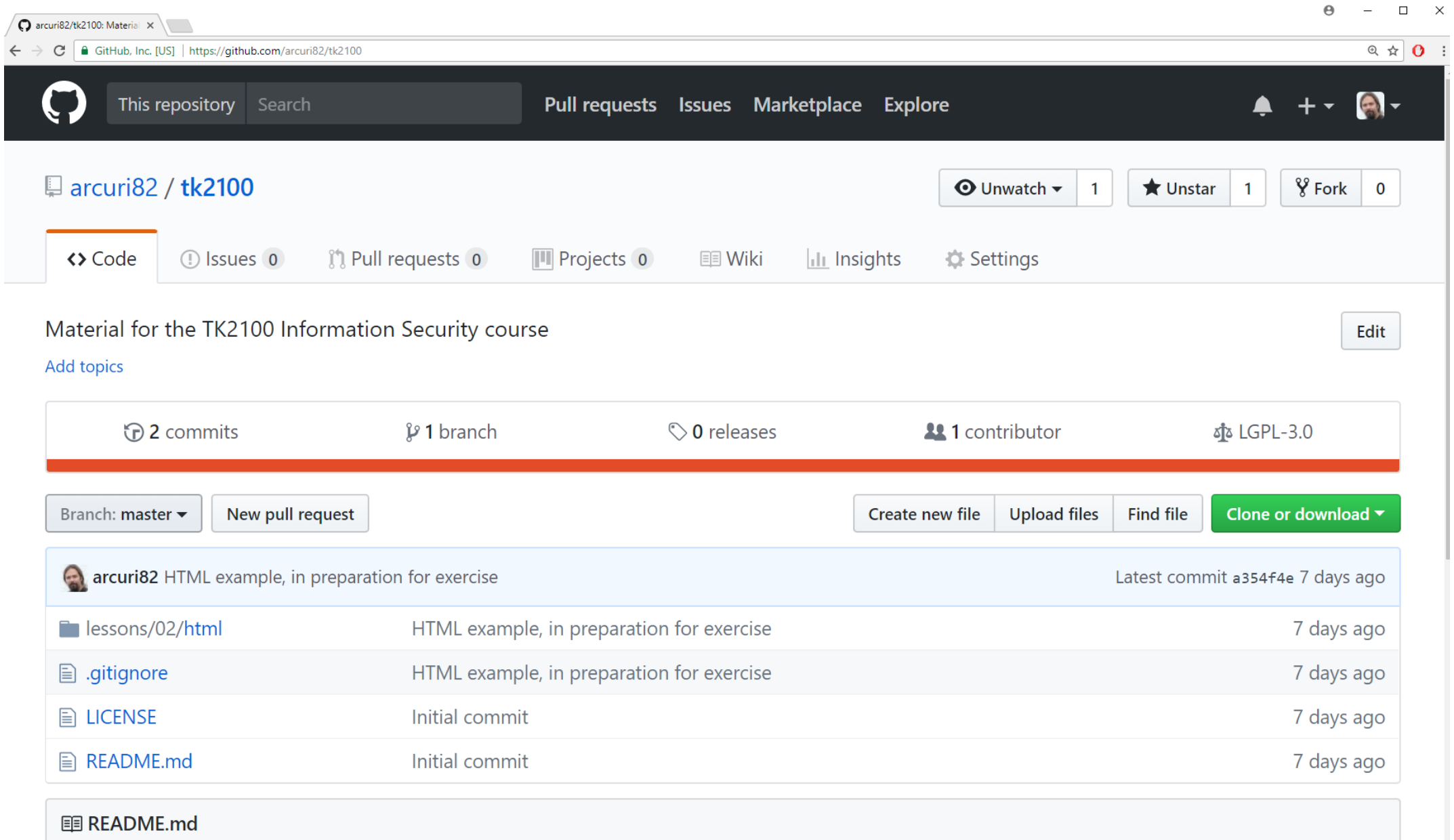
Git

- Git is a tool to share code among different developers in the same project
- Also useful for single developers to keep track of changes, and automatically have backups on remote servers
- You will see the details of Git in other courses…
- … but I am using Git to handle files in this course (eg exercises), so need to start to get used to it
- Note: usage of Git will **NOT** be part of the exam…

GitHub

- Currently the main server repository for hosting open-source projects
  - Before, the main one was SourceForge
- GitHub provides a website in which projects can be browsed
- Projects on GitHub are handled with Git
- GitHub is most famous/used, but there are others as well
  - eg, BitBucket and GitLab

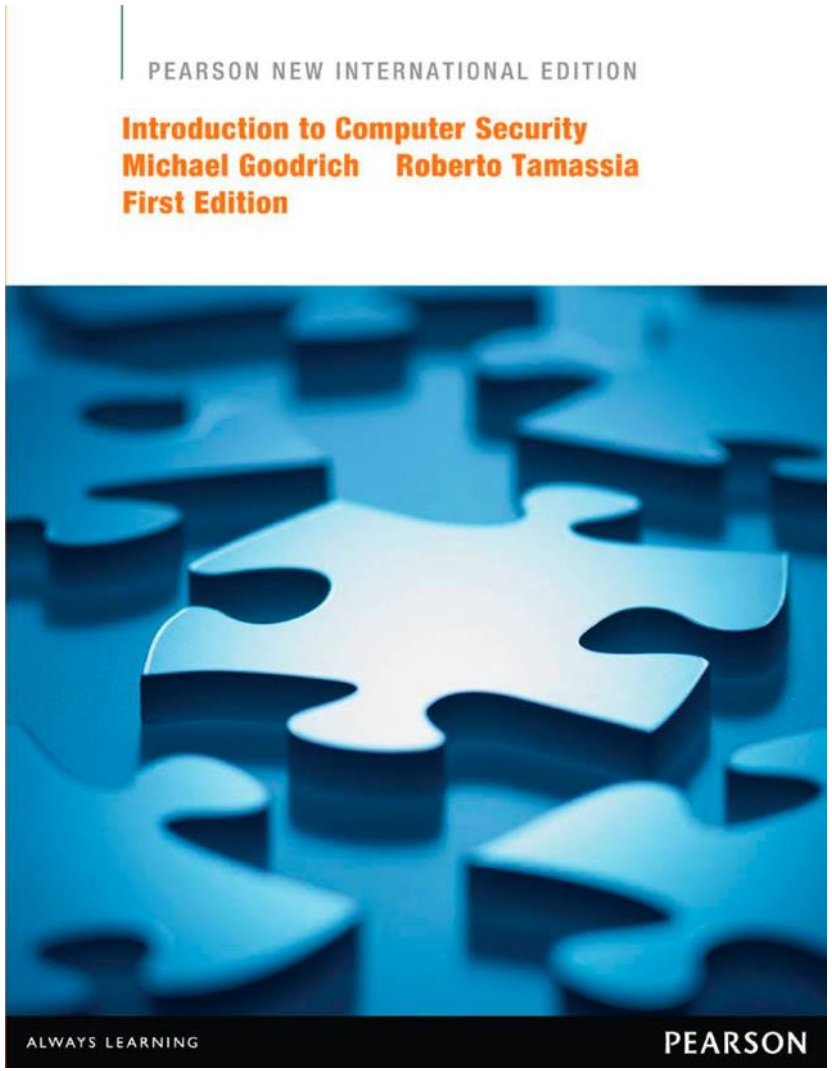
- https://github.com/arcuri82/tk2100
- Note: still under development, as not re-using material from previous years...



What you need to know/do…

- For now, you can just access the website of GitHub directly
- Once you learn Git details in other courses, and you install Git on your machine, can *clone* repository to your local machine
- “git clone https://github.com/arcuri82/tk2100.git”
  - make a copy of the repository on your hard-drive
- “git pull”
  - update your local copy with the latest changes in the repository
- Those commands can be run from command line, or from your IDE (eg, IntelliJ and WebStorm)

For Next Week



- Book pages: 2-16
- Note: when I tell you to **study** some specific pages in the book, it would be good if you also *read* the other pages in the same chapter at least once
- Note: steganography is not in the book
- Exercises for Lesson 01 on GitHub repository