Introduction

Computer and Network Security

Emilio Coppa

Course details:

- Lectures:
  - Tuesday: 11.00 - 13.00 @ Aula 205 Marco Polo
  - Friday: 08.00 - 11.00 @ Aula 205 Marco Polo
- Homeworks (more details soon)
- Website: piazza.com/uniroma1.it/fall2020/cns
- Office hours: please send me an email at coppa@diag.uniroma1.it

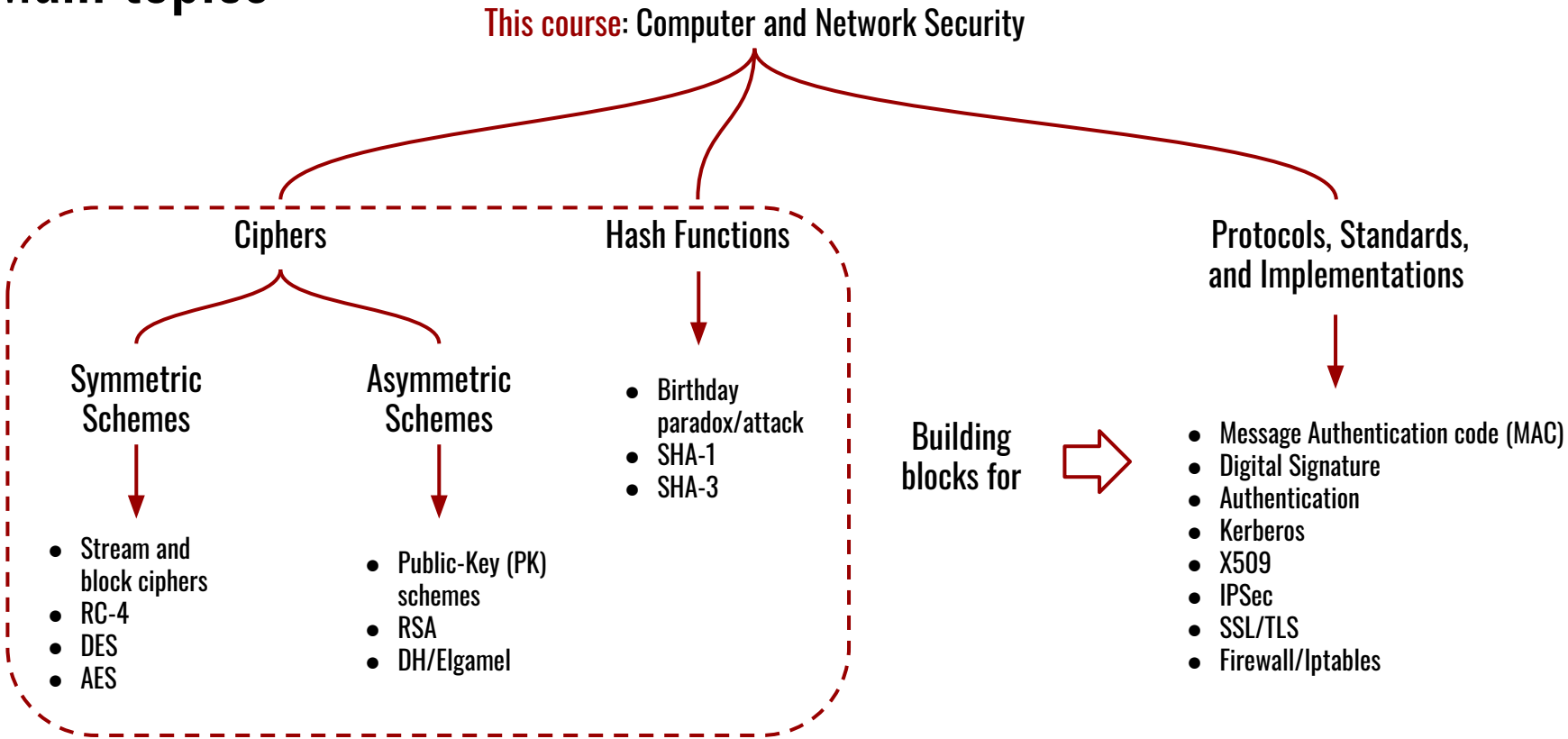
CNS Exam

- CNS is one module of a 12 CFU course:
  - Computer and Network Security (CNS): 6 CFU, taught by Prof. Emilio Coppa
  - Distributed Systems (DS): 6 CFU, taught by Prof. Silvia Bonomi

- In practice, CNS and DS are two independent courses:
  - different course websites, exams dates, exam rules, etc.
  - to pass the full 12 CFU exam, you need to pass both DS and CNS
  - the final grade is the average of scores from the two exams
  - if you pass one exam, the score is kept valid for one year (until end of 2021).

- CNS:
  - written exam (2 hours, open questions + exercises)
  - homeworks for getting a bonus (+2 points)

Main topics



The content of the course is 95% consistent with previous editions taught by Prof. D'Amore

Textbook(s)

There several books that covers the topic of this course:

- Christof Paar and Jan Pelzl. Understanding Cryptography: A Textbook for Students and Practitioners. Springer. http://www.crypto-textbook.com/

- Charlie Kauffman, Radia Perlman, Mike Speciner, and Michael Speciner. Network Security: Private Communication in a Public World. Second edition. Prentice Hall.

- Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. Handbook of Applied Cryptography. CRC Press. Download it at: http://cacr.uwaterloo.ca/hac/

- William Stallings. Cryptography and Network Security Principles and Practices. Fourth Edition. Prentice Hall.

Slides

Slides will be based mainly on:

- Slides of Prof. D'Amore from CNS 2019-2020
- Book: Understanding Cryptography: A Textbook for Students and Practitioners.
- Wikipedia (english version)

Security

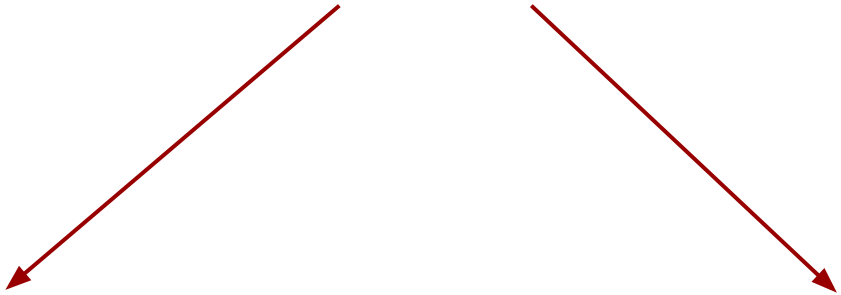
Security is freedom from, or resilience against, potential harm (or other unwanted coercive change) caused by others. Different from safety, which instead prevents from unintentional accidents.

Physical security prevents from unauthorized access to facilities, equipment and resources. Also it protects persons and their properties. Logical security involves software safeguards, it is a subset of computer security, and includes information security.

Computer security, also known as cybersecurity or IT security, refers to the security of computing devices such as computers and smartphones, as well as computer networks such as private and public networks, and the Internet.

Cryptography provides some building blocks for implementing Computer security.

Cryptology



Cryptography

Techniques for secure communications in the presence of third parties called adversaries.

algorithms and other “building blocks”

Cryptanalysis

Study of crypto systems to learn their properties, break them, etc.

attacks and proof of correctness

“Building blocks” are used to offer Security Services:

1. Confidentiality
2. (Data) Integrity
3. Availability

} CIA

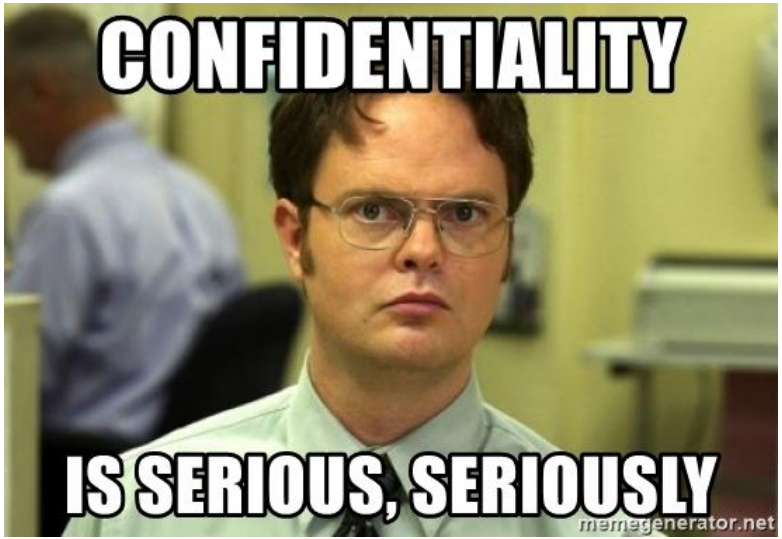
4. Message Authentication
5. Entity Authentication (identification)
6. Non repudiation

Several standards defines these concepts, define their terminology and requirements.
E.g., ISO/IEC 27000:2018, CNSS glossary, ISACA.

Security Service: Confidentiality

Information is kept secret from all except authorized parties.

- critical requirements in most application scenarios
- ciphers are designed to offer this
- several “flavours”: legal, medical, commercial, banking
- strictly tied to privacy (GDPR regulation)



Security Service: Data Integrity

Information is not tampered while in transit

- crucial for ensuring that data has not been altered by a third party
- cryptographic hash functions are often used to offer this
- crucial in most application scenarios: e.g., data corruption in a storage system

Security Service: Availability

The system or the information is reliably available to the end users

- crucial property in most systems
- security point of view:
  - prevent attacks that can affect availability
  - firewalls can help

Security Service: Message Authentication

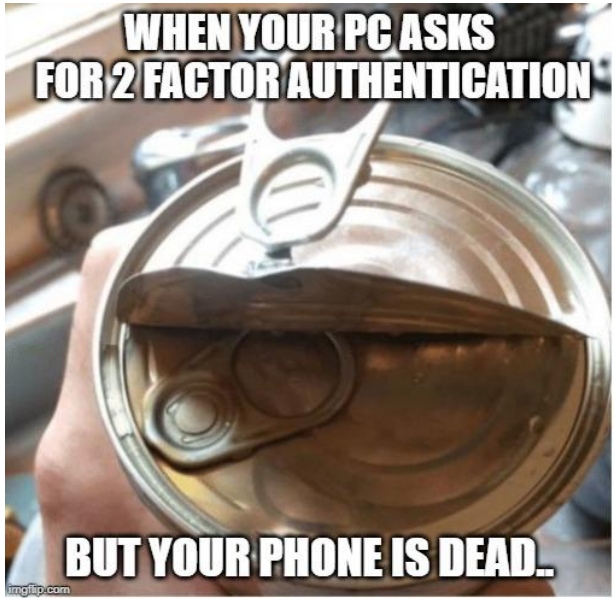
The sender/creator of the message is authentic

- Often includes integrity of the message
- Often offered with:
  - Message Authentication Code
  - Digital Signature
  - Authenticated Encryption (AE)

Security Service: Entity Authentication

Establish and verify the identity of an entity

- passwords are one common way
- single- vs multi- factor authentication: (e.g., 2FA):
  - the knowledge factors: Something the user knows (e.g., password)
  - the ownership factors: Something the user has (e.g., security token)
  - the inherence factors: Something the user is or does (e.g., fingerprint)



Security Service: Non repudiation

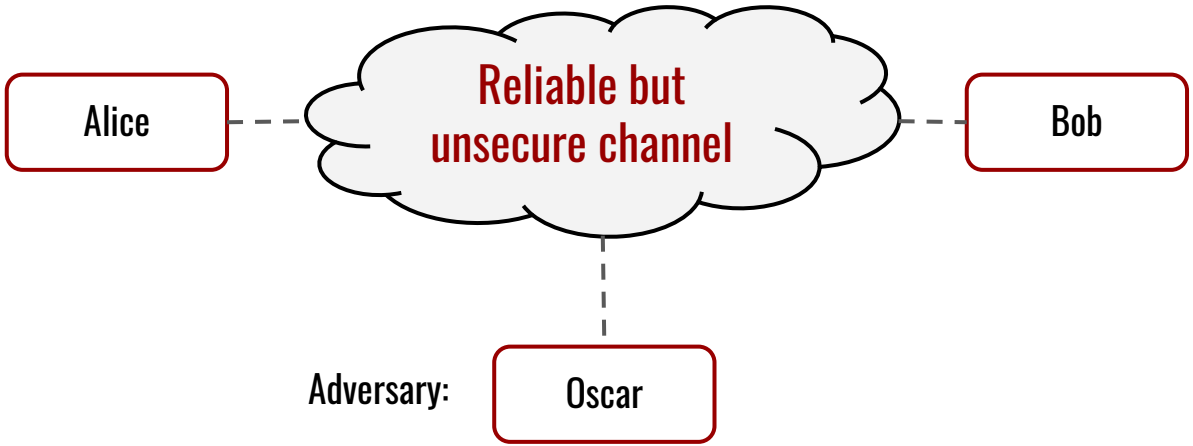
The sender of a message cannot deny the creation of the message

- crucial in several application contexts
- digital signature can offer this service

Terminology used across slides

- encryption function (and algorithm): E
- decryption function (and algorithm): $D \text{ or } E^{-1}$
- encryption key: k_1
- decryption key: k_2
- key space: number of bits used for keys
- message space: number of bits used for message
- $m = D_{k_2}(E_{k_1}(m))$
- when $k_1 = k_2$ then symmetric scheme
- when $k_1 \neq k_2$ then asymmetric scheme
- message m is also called plaintext, encrypted m is also called ciphertext

Communication Model



(Oscar as Opponent, also called Trudy or Eve)

Adversary Model

- Passive (e.g., packet sniffing):
  - adversary reads all messages exchanged in the channel

- Active (e.g., ip spoofing):
  - adversary can also forge messages (e.g., altering IP's sender) and create new messages
  - adversary may perform an attacks just by creating requests: a large number of request may lead to a Denial of service (DoS), which is hard to prevent when done with a distributed approach (DDoS). E.g., use of SYN packets.

Adversary Model (2)

We may assume that the adversary knows:

- Algorithms and protocols (no security through obfuscation)
- Message and key space

Adversary does not know the (secret) key(s).

> Kerckhoffs' Principle. A cryptosystem should be secure even if the attacker knows all details about the system, with the exception of the secret key. In particular, the system should be secure when the attacker knows the encryption and decryption algorithms.

Very hard to prove that a crypto scheme is secure. The best way is to make it open and see whether cryptanalysts are able to break it. Still, its design should be based on some good properties.

Attack models

- Eavesdropping: secretly listening to private conversation of others without their consent
- Known plaintext: attacker has samples of both plaintext and its encrypted version (ciphertext) and is at liberty to make use of them to reveal further secret information such as secret keys
- Chosen plaintext: attacker has the capability to choose arbitrary plaintexts to be encrypted and obtain the corresponding ciphertexts. The goal of the attack is to gain some further information which reduces the security of the encryption scheme. In the worst case, a chosen-plaintext attack could reveal the scheme's secret key

Attack models (2)

- Adaptive chosen plaintext: the cryptanalyst makes a series of interactive queries, choosing subsequent plaintexts based on the information from the previous encryptions.

- Chosen ciphertext: the cryptanalyst gathers information, at least in part, by choosing a ciphertext and obtaining its decryption under an unknown key.

- Physical access

Common Attacks

- Replay: a valid data transmission is maliciously or fraudulently repeated or delayed.
- Reflection: some protocols are based on “challenge-response” authentication and this attack attempts to trick an entity into proving the right answer to its own challenge.
- Man-in-the-Middle (MITM): attacker secretly relays and possibly alters the communications between two parties who believe that they are directly communicating with each other.

Security Goals

- If the secret keys are unknown to the adversary, it should be “hard” to:
  - retrieve information on the message m:
    - No adversary can determine message m (not enough)
    - No adversary can determine some information about m (not enough)
    - No adversary can determine any meaningful information about m (good)
    - This should be true even in probabilistic sense
  - retrieve the keys (even when adversary knows both plaintext and ciphertext)
- “hard” means unfeasible to compute in a reasonable amount of time given sufficient computational power (brute force all keys to decrypt: 2^{64} is feasible, 2^{80} is unfeasible)

Credits

These slides are based on material from:

- Slides of Prof. D'Amore from CNS 2019-2020
- Christof Paar and Jan Pelzl. Understanding Cryptography: A Textbook for Students and Practitioners. Springer. http://www.crypto-textbook.com/
- Wikipedia (english version)