Part I

Introduction

1. Introduction

Instructor

Instructor: Giacomo Verticale

Office hours: Mon 14:00-16:00. Please send email in advance to check that I am available.

email: giacomo.verticale@polimi.it

phone: 022399 ext. 3569

skype: g.verticale

web page: verticale.faculty.polimi.it

course material: Check the BEEP platform

Welcome

Course objectives

- Learn how crypto primitives work
- Learn how to use them correctly and reason about security

Course material

Book: Nigel Smart. Cryptography an Introduction 3rd ed. http://www.cs.bris.ac.uk/~nigel/Crypto_Book/

Book: Katz, Lindell. Introduction to Modern Cryptography. 2nd ed. Chapman & Hall

Book: Dan Boneh and Victor Shoup. A Graduate Course in Applied Cryptography. draft 0.2 https://crypto.stanford.edu/~dabo/cryptobook/

Exercise book: on the BEEP platform

Slides and other material: on the BEEP platform

Syllabus (in brief)

- Definitions of Security
  - Information-Theoretic Security
  - Semantic Security (Pseudorandom Generators and Functions)
- Using Symmetric Cryptography
  - One-Time Key vs Many-Time Key
  - Message Integrity, Authenticated Encryption
- Public-key Cryptography and Key Agreement Protocols
- Some Advanced Cryptography (Commitments, Secret Sharing, Multiparty Computation)
- Privacy Enhancing Techniques

Course Organization

Course Activities

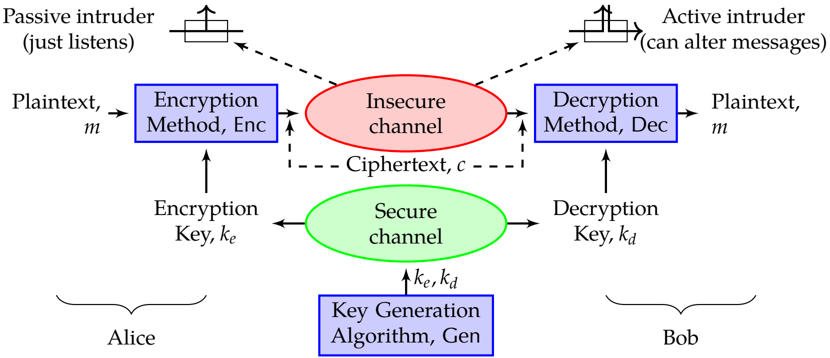
- Lectures
- Numerical Exercises
- Short programs using `sagemath` (`www.sagemath.org`)

Exam

The exam will be pen-and-paper. You will be asked to solve numerical exercises and write short programs. The exam includes a colloquium if requested by the student or by the instructor.

1. Introduction

General Model of a Secure Communications System



Based on Claude Shannon, 1949. Extended by Merkle, Diffie, and Hellmann in 1976, who defined asymmetric encryption.

Symmetric Cryptography

- The two parties share some secret information (the *key*): $k = k_e = k_d$.
- Three algorithms

Gen is a probabilistic algorithm that outputs a key k.
$k = \mathsf{Gen}()$

Enc takes as input a key k and a plaintext message m, outputs a ciphertext c. It may be probabilistic or deterministic.
$c = \mathsf{Enc}(k, m)$

Dec takes as input a key k and a ciphertext c, outputs a plaintext m. It is deterministic.
$m = \mathsf{Dec}(k, c)$

Symmetric Cryptography

The role of the key

- All algorithms are public. Security depends on the strength of the algorithms and on the secrecy of the key. (Kerckhoffs, circa 1880).
  - impossible to prevent the enemy from learning the algorithms (information leakage, bribes, etc.)
  - changing a compromised algorithm is impractical, expensive or impossible.
- "One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them" (Shannon's maxim).
- Need ways to distribute the key (chicken-and-egg problem!). But
  - keys are smaller than messages
  - keys are distributed before communication needs

Symmetric Encryption

Definitions

- The *key space* is $\mathcal{K}$.
- The *message space* is $\mathcal{M}$.
- An *encryption scheme* is defined by $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ and $\mathcal{M}$.
- Enc, $\mathcal{K}$, and $\mathcal{M}$ define all the possible ciphertexts $\mathcal{C}$. Often, but not always, $\mathcal{M} = \mathcal{C}$.

Symmetric Encryption

Key Generation and Correctness

Key Generation Assumption

Gen chooses a key uniformly at random from the key space.

Correctness

A Symmetric Encryption Scheme is correct if

$$\boxed{\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m \qquad \forall k \in \mathcal{K}, \forall m \in \mathcal{M}}$$

Correctness does not imply security!

Symmetric Cipher

Definition (Symmetric Cipher)

A set of efficient algorithms:

Gen	generates a key $k \in \mathcal{K}$
Enc	$\mathcal{K} \times \mathcal{M} \to \mathcal{C}$
Dec	$\mathcal{K} \times \mathcal{C} \to \mathcal{M}$

with $\mathrm{Dec}(k, \mathrm{Enc}(k, m)) = m, \quad \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$

Efficient means either that (choose one):

- its computation time is below a given threshold for a given message size. The threshold depends on current technology, usage scenario etc. (Concrete model)
- its asymptotic time and space complexities are polynomial vs the message size and the key size. (Asymptotic model)

Definition of Security

Any algorithm, even the most trivial, that follows the definition can be called a Symmetric Cipher. We need to discuss when a given Symmetric Cipher is secure.

Schneier's Law: "Anyone, from the most clueless amateur to the best cryptographer, can create an algorithm that he himself can't break." (Bruce Schneier, 1998)

Before defining security we must say:

1. the attacker's capabilities (who is the adversary and over which time frame)
2. the use case (how we use the algorithm)
3. what we want to prevent. This is the most difficult and error-prone task.

What is Cryptography

"Cryptography is not the weakest link in the security chain."
... yet there are many crypto failures.

So, what is cryptography?

- the crypto primitives (i.e. the equations)
- the implementation (i.e. the running code)
- the library APIs (i.e. the specifications)
- the protocols
- the usage (i.e. the assumptions)
- ...

Many fascinating ways to get it wrong!

Typical Problems with Primitives

- homebrew crypto (i.e. designers not knowledgeable enough)
- right crypto, used wrong, e.g. WiFi WEP
- right crypto stops being right, e.g. MD5, AES-256

Practical Example: GSM

- A5/0: no encryption
- A5/1: based on LFSR theoretically broken in 2000, practically broken in 2009
- A5/2: weakened A5/1 broken
- A5/3 (KASUMI): new for 3G theoretically broken in 2010

Typical Problems with Implementation

- coding errors (bounds checking, error checks, ...)
- leftovers of test phase (backdoors, shortcuts, ...)
- assumptions of library behavior
- bad key management
- bad random generators
- untrusted platforms
- side-channel attacks
- non-tamper-proof hardware

Typical Problems with Usage

- weak password choices, refusal to change defaults
- loss of keys
- dishonest behavior

Core Objectives

Confidentiality: protects data against passive attackers

Integrity: protects data against modification

Authentication:
- Data origin authentication: protects data against forgery
- Entity authentication: protects entities against impersonation

Non-repudiation: allows to prove integrity and authentication to a third party

Newer objectives are rising: privacy control, proof of knowledge, immutability, etc.

Basic Principles of Modern Cryptography

Principle 1. Formulation of Exact Definitions

- Importance for Design (what we want / do not want to achieve)
- Importance for Usage (does this construction fits the application?)

Principle 2. Reliance on Precise Assumptions

- Attacker capabilities
- Mathematical properties of particular functions

Principle 3. Rigorous Proofs of Security

- *If the Assumptions are true, Construction X is secure according to the Definition*