AI and Cryptography

Lecture 1 – Course Overview and Review of Crypto

Luca Mariot

Semantics, Cybersecurity and Services Group, University of Twente

l.mariot@utwente.nl

Trieste, June 26, 2023

This Lecture

Course Overview

Basic concepts of cryptography

Security Models in Cryptography

Symmetric cryptosystems

Public-key cryptosystems

Who Am I?

Luca Mariot
https://lucamariot.org

Assistant Professor at University of Twente

Research Interests:

- Cryptography
- Evolutionary Computing
- Cellular Automata
- Security of AI

Course Overview

Goal

Give an overview of research problems at the intersection of:

- Artificial Intelligence (AI)
- Cryptography

Two directions:

- AI methods for sound cryptography
- Cryptographic techniques for secure and private AI

Learning Objectives

By the end of this course, you should be able to:

1. Employ AI methods to:
   1.1 **Design** strong cryptographic primitives
   1.2 **Assess** the security of such primitives
2. Employ cryptographic techniques to:
   2.1 **Analyze** relevant security and privacy threats in AI models
   2.2 **Apply** cryptographic countermeasures to mitigate such threats

Lectures Plan & Timetable

Location: Room 4B, H2bis building

Date/Time	Lecture
26/6, 14:30-16:30	1. Course Overview and Review of Cryptography
27/6, 10:00-12:00	2. AI to Design Cryptographic Primitives (1)
27/6, 14:30-16:30	3. AI to Design Cryptographic Primitives (2)
28/6, 10:00-12:00	4. Adversarial Examples in ML
28/6, 14:30-16:30	5. Differential Privacy for Adversarial Robustness
29/6, 10:00-12:00	6. Deep Learning-based Side-Channel Analysis*
29/6, 14:30-16:30	7. Secure Multiparty Computation for Private ML
30/6, 10:00-12:00	8. Wrap-up and discussion of possible research/assessment topics

* **Only for lecture 6**: Room 2A Morin, H2bis building

Guest Lecture

Lecture 6 on Deep-Learning-based Side-Channel Analysis will be given by Stjepan Picek

Stjepan Picek

Associate Professor at Radboud University

Research Interests:

- Symmetric Cryptography
- AI Security & Privacy
- Machine Learning
- Side-Channel Analysis

Prerequisites & Assessment

Prerequisites: Basic notions of Machine Learning. Knowledge of crypto is useful, but will be reviewed throughout the course

Assessment Description

Write a **short report** ($\approx$8 pages) on a research topic agreed with the instructor.

- Many topics cannot be covered in the span of this course
- Assessment can be both a theoretical/experimental contribution, or a concise survey of a particular topic
- Last lecture: discuss possible topics for the report

This Lecture

Course Overview

Basic concepts of cryptography

Security Models in Cryptography

Symmetric cryptosystems

Public-key cryptosystems

Reading material

Main topics:

- Basic concepts of cryptography
- Encryption: Security Models
- Examples of symmetric and public-key cryptosystems

References:

- Introduction to Modern Cryptography. J. Katz, Y. Lindell
- Cryptography: An Introduction. N. Smart.
- Cryptography: Theory and Practice. D. Stinson.

What is Cryptography?

Historically:

- The art of secret writing
- Mainly relied on unsound methods till the 20th century (e.g. monoalphabetic and polyalphabetic substitutions)

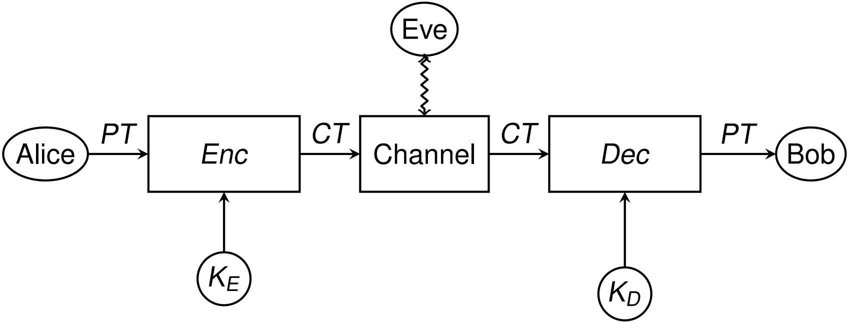
Modernly:

- *Precise* definitions, and *rigorous* proofs of security
- Expanded the scope beyond encryption to include *authenticity*, *integrity*, and *secure protocols*

Encompassing Definition

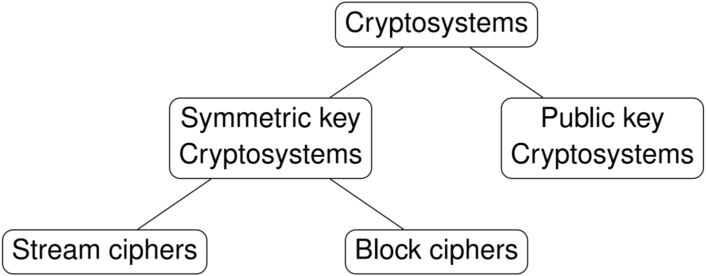
Construction and analysis of schemes that should be able to withstand any abuse, even in presence of malicious adversaries.

Encryption – The Basic Scenario



- *PT*: plaintext
- *Enc*: encryption function
- K_E: encryption key
- *CT*: ciphertext
- *Dec*: decryption function
- K_D: decryption key

Classification of Cryptosystems



- **Symmetric**: the same key is used for encryption and decryption
- **Public key**: different encryption and decryption keys

This Lecture

Security Models of Cryptosystems

Definition

A *security model* defines how difficult is for Eve to attack a cryptosystem.

- **Adversarial goal**: what is Eve's objective
- **Attack model**: the amount of information that Eve has
- **Security level**: the (computational) resources that Eve has

Kerchoff's Principle

The specification of the encryption and decryption algorithms is assumed to be public. The only secret information is the key.

Adversarial goals

- **Total break**: Eve is able to discover the decryption key
- **Partial break**: Eve is able to decrypt some ciphertexts with a certain probability, without knowing the decryption key
- **Distinguishing break**: without knowing the decryption key, Eve is able to distinguish between the encryption of two plaintexts with a probability higher than $1/2$

Attack models

- **Known ciphertext attack** (KCA): Eve only knows some ciphertexts encrypted with the same unknown key
- **Known plaintext attack** (KPA): Eve gains access to some pairs of plaintext/ciphertext encrypted under the same unknown key
- **Chosen plaintext attack** (CPA): Eve is able to choose some plaintexts, and to obtain the corresponding ciphertexts encrypted under the same unknown key
- **Chosen ciphertext attack** (CCA): Eve is able to choose some ciphertexts, and to obtain the corresponding plaintexts decrypted under the same unknown key

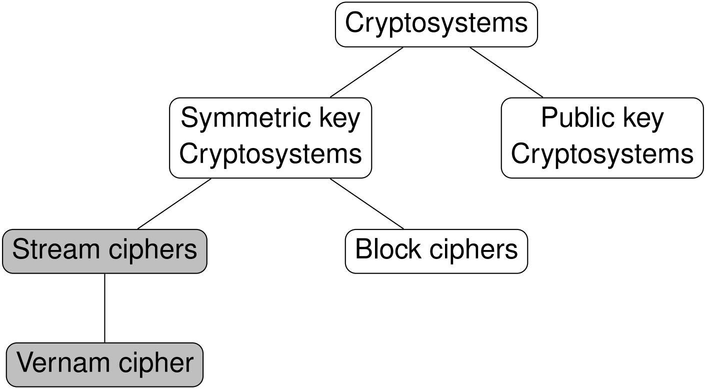
Security Levels

- **Computational security**: the best attack that Eve can apply on the cryptosystem requires at least N operations, with N being a very large number
- **Provable security**: A mathematical problem is reduced to the task of breaking the cryptosystem. Hence, breaking the cryptosystem is at least as difficult as efficiently solving the problem (which is thought to be computationally hard)
- **Unconditional security**: The cryptosystem cannot be broken under the assumed attack model, even if Eve has infinite computational resources

This Lecture

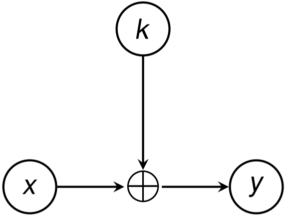
- Course Overview
- Basic concepts of cryptography
- Security Models in Cryptography
- Symmetric cryptosystems
- Public-key cryptosystems

Classification of Cryptosystems

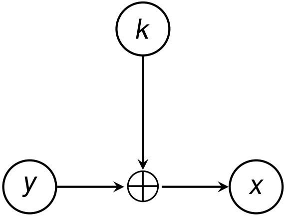


An unconditionally secure cipher – The Vernam Cipher

- A symmetric stream cipher that encrypts one bit at a time



(a) Encryption



(b) Decryption

- $x \in \{0,1\}^n$: plaintext
- $y \in \{0,1\}^n$: ciphertext
- $k \in \{0,1\}^n$: symmetric key
- $\bigoplus$: bitwise XOR

- The Vernam cipher is also called **One-Time Pad** (OTP): each plaintext is encrypted with a *different* key

Security of the OTP

- The OTP is unconditionally secure under KCA:

Theorem (Shannon 1949)

The OTP is the only cipher satisfying perfect secrecy: for any plaintext $x \in \mathcal{P}$ and ciphertext $y \in \mathcal{C}$, it holds

$$Pr[x|y] = Pr[x]$$

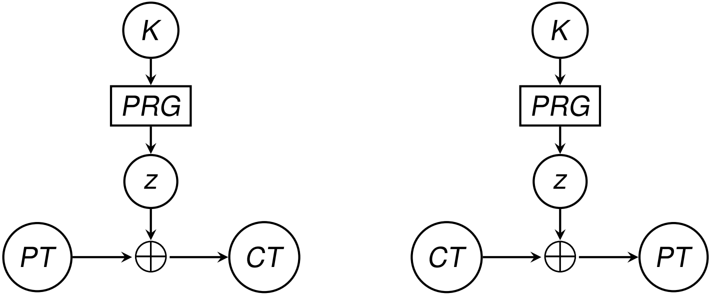
In other words, knowing the ciphertext y gives no information on the plaintext x

Drawbacks:

- The key must be as long as the plaintext
- Keys cannot be reused (hence the name "One-time")
- Keys must be *random*

Vernam-like Stream Cipher

- **PRG**: Pseudorandom generator that stretches a short secret key K into an arbitrary long keystream z

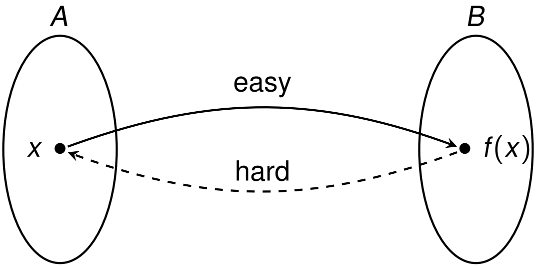


(a) Encryption (b) Decryption

- Keystream eventually repeats (therefore, no more unconditional security)

One-Way Functions

- A one-way function $f : A \rightarrow B$ has the following properties:
  - Given $x \in A$, it is "easy" to compute $f(x)$
  - Given $y \in B$, it is "hard" to compute x such that $f(x) = y$

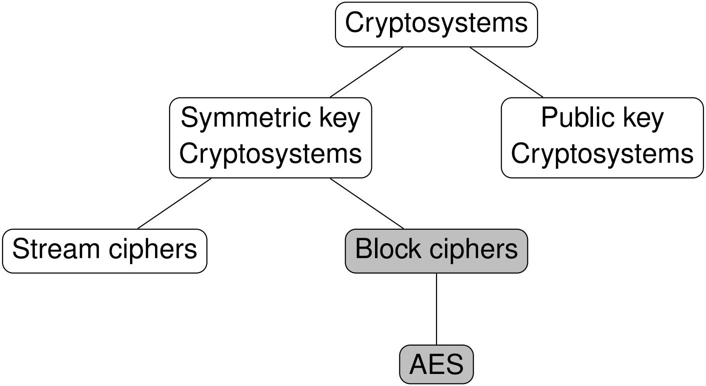


- "Easy" and "hard" $\Rightarrow$ in terms of **Computational Complexity**
- One-Way *Permutations* $\Rightarrow$ PRG

One-Way Functions and $\mathcal{P}$ vs. $\mathcal{NP}$

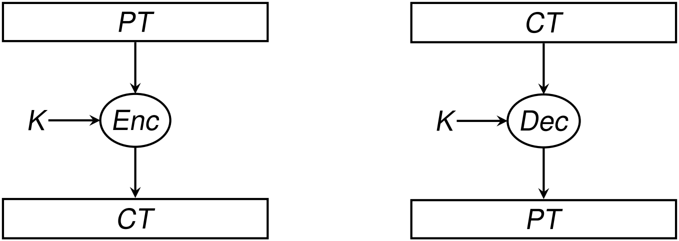
- Remark: if $\mathcal{P} = \mathcal{NP} \Rightarrow$ no one-way functions exist, and no computationally secure crypto is possible!
- On the other hand: if a one-way function exists $\Rightarrow \mathcal{P} \neq \mathcal{NP}$.
- If $\mathcal{P} \neq \mathcal{NP}$, it could still be that *no* one-way function exists: $\mathcal{NP}$ captures the problems that are hard to solve in the *worst case*.
- Since we do not know how to prove the (non-)existence of one-way functions, we simply assume that they exist.
- Examples of *candidates*: multiplication, modular exponentiation, block ciphers

Classification of Cryptosystems



Block Ciphers

- *Block ciphers* encrypt fixed-size *blocks* of plaintext, and generate ciphertext blocks of the same length

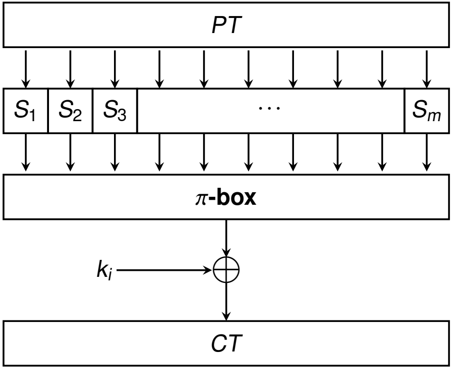


(a) Encryption (b) Decryption

- Longer messages are encrypted one block at a time

Substitution-Permutation Networks (SPN)

- Plaintext block size $b = mn$.
- Round of a SPN cipher:



- **Confusion layer**: S-boxes $S_i : \{0,1\}^n \rightarrow \{0,1\}^n$
- **Diffusion layer**: P-box $\pi : \{0,1\}^{mn} \rightarrow \{0,1\}^{mn}$

The Advanced Encryption Standard (AES)

- In 1997, the NIST called for a competition for selecting a new block cipher to supersede DES
- The winner cipher would have later been adopted as the *Advanced Encryption Standard* (AES)
- 15 candidates ciphers were submitted, out of which 5 finalists were selected:
  - RIJNDAEL
  - SERPENT
  - TWOFISH
  - RC6
  - MARS
- After a further selection round, RIJNDAEL was selected

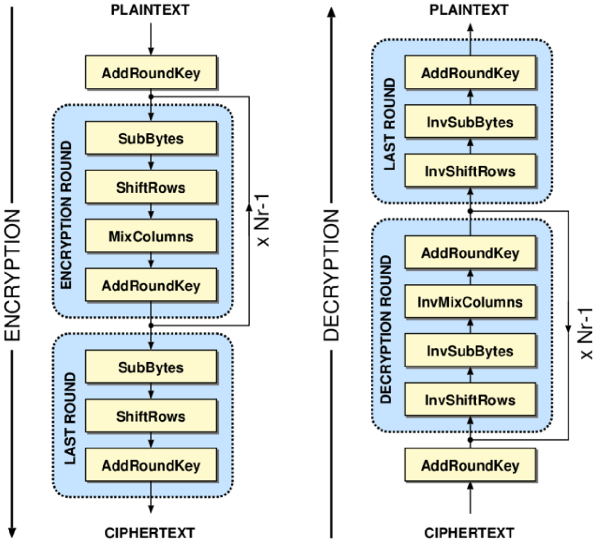
RIJNDAEL – The AES algorithm

- Designed by Joan Daemen and Vincent Rijmen
- SPN cipher with block size of 128 bits
- The number of rounds is determined as follows:

Key length	Rounds
128 bits	10
192 bits	12
256 bits	14

- Byte-oriented designed, which results in efficient software implementations

High-level Description – Block Scheme



Confusion: SubBytes

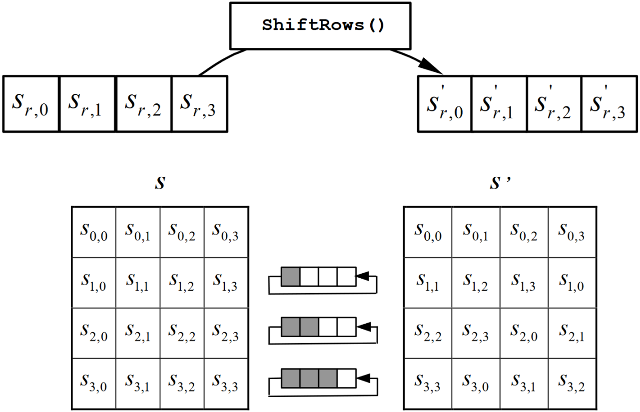
- **Nonlinear transformation**: multiplicative inverse in $\mathbb{F}_{2^8}$
- Each byte is represented as a polynomial in $\mathbb{F}_2[x]/(1+x+x^3+x^4+x^8)$

$$F(x) = \begin{cases} \frac{1}{x} = x^{254} & \text{, if } x \neq 0 \\ 0 & \text{, if } x = 0 \end{cases}$$

- **Affine transformation**: $A \cdot F(x) + c$, where:

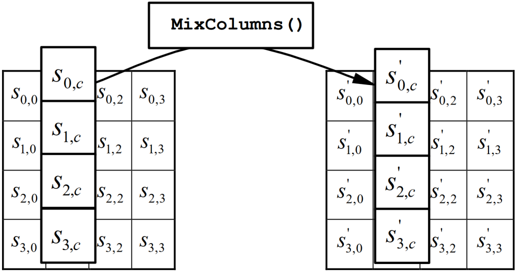
$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

- ShiftRows: shifts the rows of a 4×4 bytes matrix



Diffusion: MixColumns

- MixColumn: mixes the bytes of a 4-byte column
- Optimal diffusion by using a *MDS matrix*



This Lecture

Course Overview

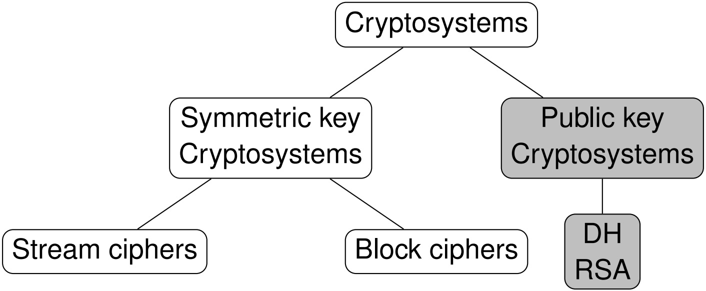
Basic concepts of cryptography

Security Models in Cryptography

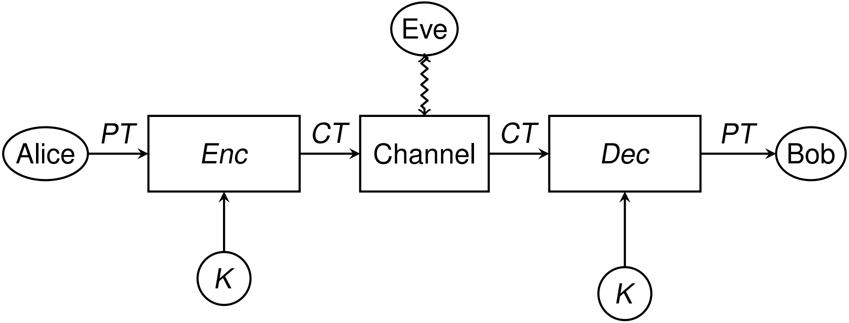
Symmetric cryptosystems

Public-key cryptosystems

Classification of Cryptosystems

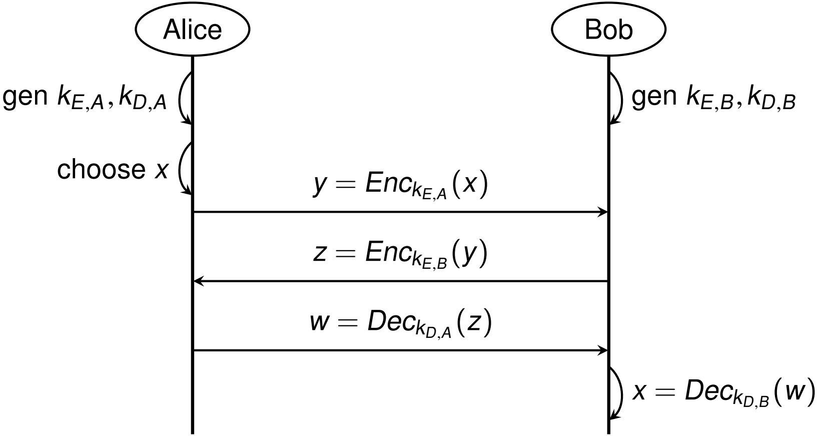


Public-Key Cryptography: The Basic Idea

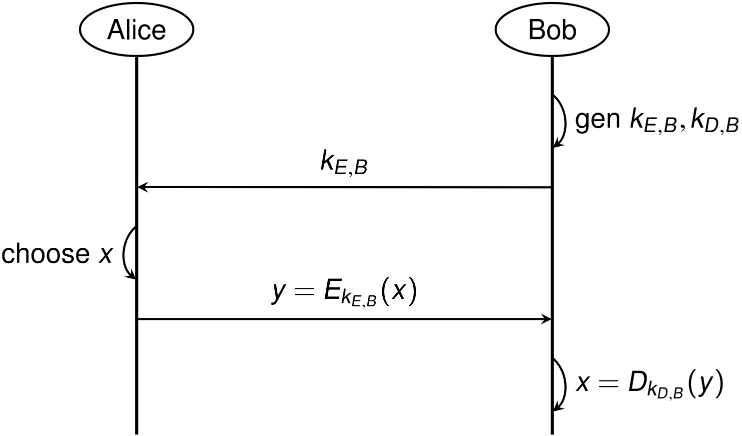


- **Problem:** how to exchange a symmetric key?

First Protocol – Scheme

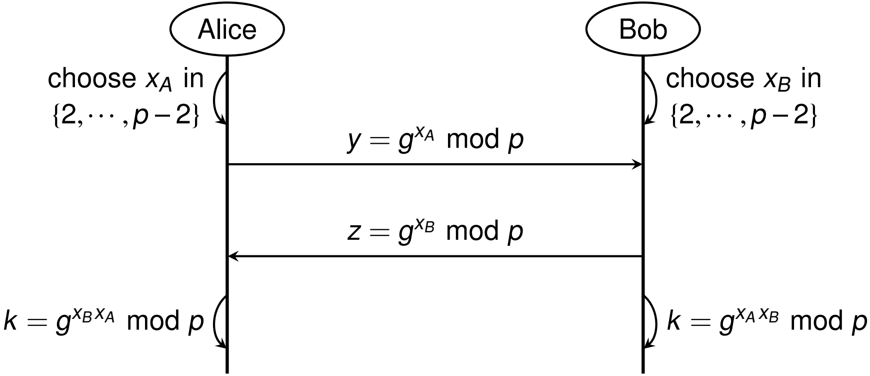


Second Protocol – Scheme



Diffie-Hellman Protocol – Scheme

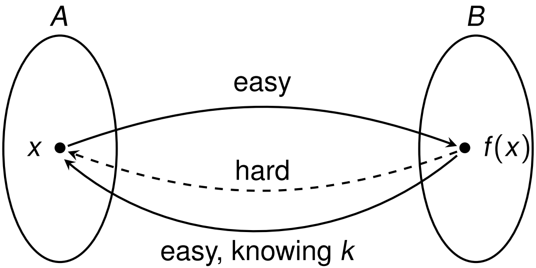
In public: choose large prime p, generator g of $\mathbb{Z}_p^*$



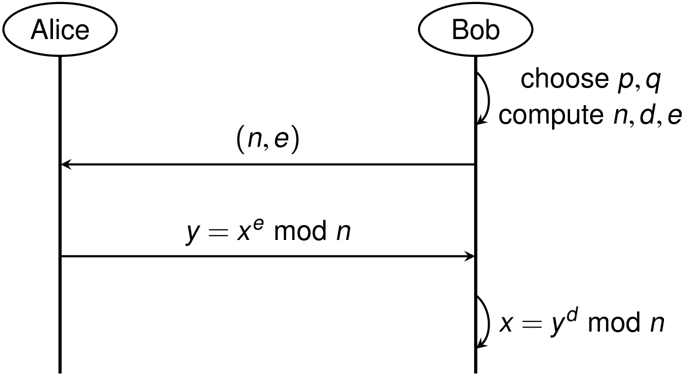
- Finding x_A from $g^{x_A} \bmod p$ or x_B from $g^{x_B} \bmod p$ requires solving a *discrete logarithm*

Trapdoor One-Way Permutations

- A one-way permutation $f : A \to B$ has a *trapdoor* if:
  - Given x, it is easy to compute $f(x)$
  - Given y, by knowing a secret information k it is easy to compute x such that $f(x) = y$
  - Given y, it is difficult to compute x such that $f(x) = y$ if one does not know k
- The secret information k plays the role of decryption key



RSA – Scheme



- Finding d from (n, e) requires *factorizing* n

Security Considerations for DH & RSA

- The largest number that has been factored using specialized factoring algorithms has 250 decimal digits (829 bits)[1]
- As of 2015, NIST recommends to use RSA keys of at least 2048 bits to be on the safe side
- Common key sizes for DH and RSA include 2048, 3072 and 4096 bits
- **Elliptic Curve DH**: uses a more efficient representation for the multiplicative group ⇒ key sizes of 128, 256, 384 bits

[1] see `https://en.wikipedia.org/wiki/RSA_Factoring_Challenge`, accessed on 24 Feb 2020

Quantum Computing & Post-quantum Crypto

- **Remark**: factorization and discrete logarithm ⇒ assumed hard with *classical* computers
- *Shor's algorithm* solves both problems in polynomial time on a *quantum computer*
- **Not clear** if quantum computers will ever scale enough to break RSA and DH
- **Post-quantum** encryption schemes based on other assumptions:
  - Lattice-based cryptography
  - Code-based cryptography
  - Isogenies

Recap

Summarizing:

- Modern cryptography: precise definitions and rigorous proofs
- The (assumed) existence of one-way functions is *central* for symmetric cryptography
- Public-key cryptography relies on the existence of one-way functions (DH) and trapdoor one-way permutations (RSA)

Other topics (beyond encryption):

- Authentication and digital signatures
- Zero-knowledge proofs
- Differential Privacy (covered in Lecture 5)
- Secure Multiparty Computation (covered in Lecture 7)