Introduction to mathematical cryptography

Lecture 1: Classical cryptography and discrete logarithms

Sabrina Kunzweiler

Preliminary Arizona Winter School 2025

Inria

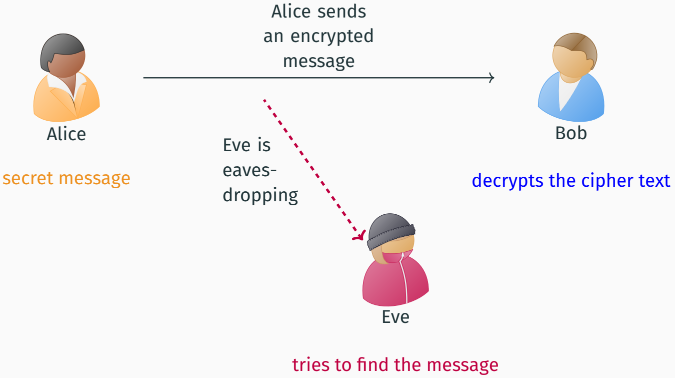
The University of Arizona

Institut de Mathématiques de Bordeaux

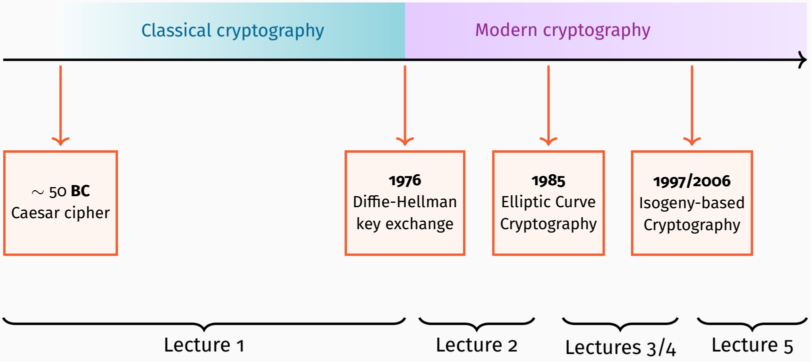
What is cryptography?

$$\underbrace{\kappa\rho\upsilon\pi\tau o\varsigma}_{\text{to hide}} + \underbrace{\gamma\rho\alpha\varphi\epsilon\iota\nu}_{\text{to write}}$$

Cryptography is used to obscure information from an eavesdropper.

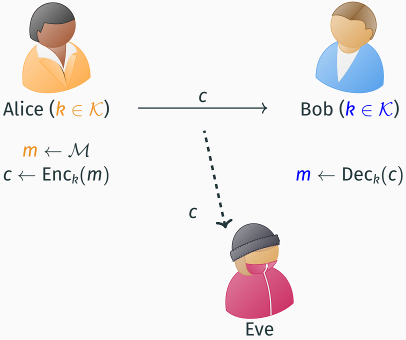


What will we learn in this course?



A brief introduction to cryptography

Encryption scheme



$\mathcal{K}$ key space

$\mathcal{M}$ message space

$\mathcal{C}$ cipher text space

Encryption function: $\text{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$

Decryption function: $\text{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

requirement: $\text{Dec}_k(\text{Enc}_k(m)) = m \quad \forall m \in \mathcal{M}, k \in \mathcal{K}$

Caesar cipher

historical cipher used by Julius Caesar (100 BC - 44 BC)

- **Idea**: Shift every letter in a word by 3 positions.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	...
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	...

- **Example**: HELLO $\mapsto$ KHOOR

Formal description

- $\mathcal{A} = \{A, \ldots, Z\} = \{0, \ldots, 25\}$
- $\mathcal{M} = \mathcal{C} = \mathcal{A}^* = \{a_1 \ldots a_n \mid a_i \in \mathcal{A}, n \in \mathbb{N}\}$
- $\text{Enc}(m_1 \ldots m_n) = (c_1 \ldots c_n)$ with $c_i = m_i + 3 \pmod{26}$
- $\text{Dec}(c_1 \ldots c_n) = (m_1 \ldots m_n)$ with $m_i = c_i - 3 \pmod{26}$

⚠ There is no key.
Anyone knowing the encryption method can decrypt!

Kerckhoff's principle

> *Il faut qu'il [le système] n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi.*
>
> — Auguste Kerckhoff, *la cryptographie militaire, 1883*

→ The system must be secure, even if everything about it, except the secret key, is public knowledge.

→ Important for modern cryptography. Why?

- makes cryptography better (public peer review)
- keeping the scheme secret is unrealistic in most scenarios
- easier to change a secret key than changing the entire system

Opposite principle: Security through obscurity.

- unlikely to provide long-term security
- can be used to *complement* a (public) system

Improving the Caesar cipher?

Version 1: choose a secret shift $k \in \{0, \ldots, 25\}$ $\Rightarrow$ 26 keys.

Version 2: choose a secret linear transformation
$m \mapsto k_a \cdot m + k_b \ (mod\ 26)$

$\Rightarrow 26 \cdot 12$ keys

$\Rightarrow$ The key spaces are too small.
An attacker can test all possible keys until a valid text is found.

Exercise sage

Decipher: IFELTKH URFENHA FEEFSFU TSVGEDN ULTKFBF

Improving the Caesar cipher?

Version 3: choose a secret permutation of the letters

$\Rightarrow 26! \approx 2^{88}$ keys

$\Rightarrow$ Still insecure against **frequency analysis**.

- **Idea**: each language has a characteristic distribution of letters or other patterns
- **English language**
  - most common letters: *E*, *T*, *A*
  - most common pairs: *TH*, *ER*, *ON*
  - most common repeats: *SS*, *EE*, *TT*
- first sources from 8th century: رسالة في استخراج المعمى
  (A Manuscript on Deciphering Cryptographic Messages, Al-Kindi)

Exercise sage

Decipher: JIVQOJIV LEALAVQO KGOONDTV QOAELONE OAINYNGJ SOB-VQODB CLAVQOKG OONDTJIV QOJIVLEA EIBHTBLO YBLEQPIG AA

Symmetric cryptography vs public key cryptography

Symmetric cryptography

- Alice and Bob share the same secret key k
- Examples of symmetric encryption schemes:
  - variants of the Caesar cipher (historical, insecure)
  - AES = Advanced Encryption Standard (modern scheme, standardized in 2000)
- Alice and Bob need to agree on a secret key in advance

Public key cryptography

- Alice and Bob have their own secret key sk and a corresponding public key pk, related by a cryptographic one-way function

- important cryptographic primitive: Public key exchange
  $\Rightarrow$ allows Alice and Bob to find a shared secret key communicating over a public channel

Cryptographic one-way functions

Definition

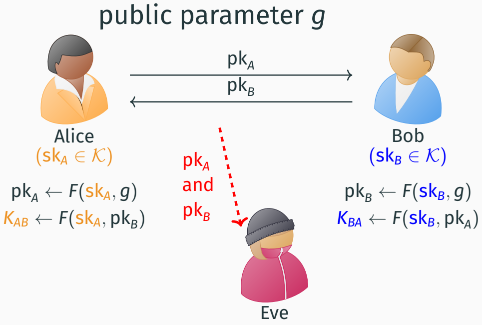
A function $f : X \to Y$ is a cryptographic one-way function if

1. f is easy to compute,
2. f is hard to invert.

Conjectured (!) examples

- Multiplication: $F : \mathcal{P} \times \mathcal{P} \to \mathbb{Z}$, where $\mathcal{P}$ is the set of primes.
  - Given $p, q \in \mathcal{P}$, we can compute $p \cdot q$ in polynomial time
  - Factoring $N = p \cdot q$ is computationally (!) hard.
- Modular exponentiation (Section 2)
- Elliptic curve multiplication (Section 3)
- Isogenies (Section 4)

Public key exchange



- sk: secret key
- pk: public key
- F: one-way functions
  - $\Rightarrow$ Given pk_A it is hard to find sk_A.

requirement: $F(\mathrm{sk}_A, F(\mathrm{sk}_A, g)) = F(\mathrm{sk}_B, F(\mathrm{sk}_A, g))$ $\forall \mathrm{sk}_A, \mathrm{sk}_B$, so that $K_{AB} = K_{BA}$

Example based on factorizationNon-example

Public parameter: $p \in \mathcal{P}$ large prime

Alice

$(sk_A = q_A \in \mathcal{P})$

$\mathrm{pk}_A = N_A \leftarrow q_A \cdot p$

$K_{AB} \leftarrow q_A \cdot N_B$

$\xrightarrow{N_A}$

$\xleftarrow{N_B}$

Bob

$(\mathrm{sk}_B = q_B \in \mathcal{P})$

$\mathrm{pk}_B = N_B \leftarrow q_B \cdot p$

$K_{BA} \leftarrow q_B \cdot N_A$

- correctness ✓
  $K_{AB} = q_A \cdot N_B = q_A \cdot (q_B \cdot p) = q_B \cdot (q_A \cdot p) = q_B \cdot N_A = K_{BA}$
- security x[1]
  Given N_A, the secret key q_A is efficiently computed as $q_A = N_A/p$.
  $\Rightarrow f_p : \mathcal{P} \rightarrow \mathbb{N}$ with $f_p(q) = p \cdot q$ is <u>not</u> a one-way function.

[1] It has proven to be difficult to construct key exchange based on factorization, but there is an important public key encryption scheme related to this problem: RSA.

Discrete logarithm problem and Diffie-Hellman key exchange

New directions in cryptography

644 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-22, NO. 6, NOVEMBER 1976

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

- 1976: Whitfield Diffie and Martin Hellman propose the *first* key exchange protocol
- Marks the beginning of “modern cryptography”: *changing the ancient art into a science*
- Diffie-Hellman key exchange is the basis of many modern protocols

Modular exponentiation and the discrete logarithm problem

In this lecture, we consider modular exponentiation for some prime field $\mathbb{F}_p$:

$$\exp_g : \mathbb{Z} \to \mathbb{F}_p, \quad a \mapsto g^a,$$

Discrete Logarithm Problem (DLP)

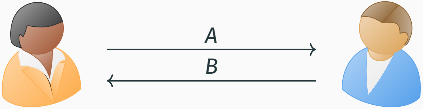
For $g \in \mathbb{F}_p^*$ primitive root, $A \in \mathbb{F}_p^*$,
the DLP asks to find $a \in \mathbb{Z}$ so that $\exp_g(a) = A$.
Notation: $a = \mathrm{dlog}_g(A)$.

- $\exp_g$ is easy to compute (square-and-multiply techniques)
- No polynomial-time algorithms for computing dlog_g are known (next lecture)

$\Rightarrow$ $\exp_g$ is a (conjectured) cryptographic one-way function.

Diffie-Hellman key exchange

public parameter: $g \in \mathbb{F}_p^*$



Alice	Bob
($a \in \mathbb{Z}$)	($b \in \mathbb{Z}$)
$A \leftarrow \exp_g(a)$	$B \leftarrow \exp_g(b)$
$K_{AB} \leftarrow \exp_B(a)$	$K_{BA} \leftarrow \exp_A(b)$

correctness ✓

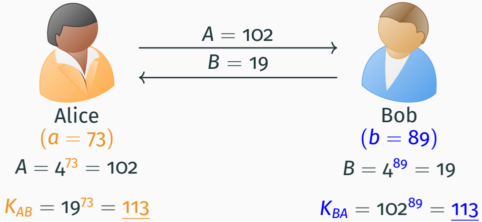
$K_{AB} = \exp_B(a) = (g^b)^a = (g^a)^b = \exp_A(b) = K_{BA}$

security

- Given g and $\text{pk}_A = A$, it is hard to compute $\text{sk}_A = a = \text{dlog}_g(A)$ **if** the DLP is hard in $\mathbb{F}_p$.
- Given g, $\text{pk}_A = A$, $\text{pk}_B = B$, it *seems hard* to compute $K_{AB} = K_{BA}$ without solving the DLP (slide 17).

Example: Diffie-Hellman key exchange

public parameter: $g = 4 \in \mathbb{F}_{227}$



Note $ord(4) = 113 \neq 226 = p - 1$

- This choice is made on purpose in order to work in a prime-order subgroup of $\mathbb{F}_p^*$

For $g \in \mathbb{F}_p^*$, $A \in \langle g \rangle$, the DLP and the notation $\mathrm{dlog}_g(A)$ are well-defined (analogous to the definition on slide 13).

Why work in a prime order subgroup?

How hard is it to solve the DLP for some parameters $g \in \mathbb{F}_p^*$ and A?

- Naive approach: For all $a \in \{0, \ldots, q-1\}$ check if $\exp_g(a) = A$.
  better algorithms in the next lecture

$\Rightarrow$ Intuitively, the hardness depends on $q = ord(g)$.

- We can do better if $ord(g) = q$ is composite!

Example $p = 443$, $g = 2 \in \mathbb{F}_p^*$ with $ord(g) = 442 = 2 \cdot 13 \cdot 17$.
We want to find $a = \text{dlog}_g(A)$ with $A = 74$.

- $a \pmod 2$: Compute $A^{221} = 442 \neq 1$, hence $a \equiv 1 \pmod 2$.
- $a \pmod{13}$: Compute $A' = A^{2 \cdot 17} = 356$ and $g' = g^{2 \cdot 17} = 35$.
  $A' \in \langle g' \rangle$ and $ord(g') = 13$. We find $\text{dlog}_{g'}(A') = 6$, hence $a \equiv 6 \pmod{13}$.
- $a \pmod{17}$: Analogously, we find $a \equiv 4 \pmod{17}$.

$\Rightarrow$ Chinese remainder theorem: $a \equiv 123 \pmod{2 \cdot 13 \cdot 17}$

General approach: Pohlig-Hellman algorithm (see the exercises)

The computational Diffie-Hellman problem

Computational Diffie-Hellman problem (CDH)

For $g \in \mathbb{F}_p^*$, $A = g^a$, $B = g^b$ with (secret) a, b, the CDH asks to find $C = g^{ab}$.

- If CDH is hard, then DLP is hard (CDH reduces to DLP)
- Are the problems equivalent? open question
  - The best known algorithms to solve CDH rely on solving DLP
  - Maurer reduction: reduction from DLP in group A to CDH in group B (constructing B is not easy)
  - Algbraic group model: equivalence proven under the assumption that the adversary is *algebraic*
- **Food for thought**: Which of the following are easy to compute?

$$g^{a+b}, \quad g^{a-b}, \quad g^{a^2}, \quad g^{2a}, \quad g^{-a}, \quad g^{1/a}, \quad g^{a/b}$$

Man-in-the-middle-attack

public parameter: $g \in \mathbb{F}_p^*$

Alice	Mallory	Bob
$(a \in \mathbb{Z})$	$(c \in \mathbb{Z})$	$(b \in \mathbb{Z})$
$A \leftarrow \exp_g(a)$	$C \leftarrow \exp_g(c)$	$B \leftarrow \exp_g(b)$
$K_1 \leftarrow \exp_C(a)$	$K_1, K_2 \leftarrow \exp_A(c), \exp_B(c)$	$K_2 \leftarrow \exp_C(b)$

$\Rightarrow$ The Diffie-Hellman key exchange protocol is not secure against active adversaries

- additional authentication is required
- Diffie-Hellman key exchange serves as an important building block for such advanced protocols

El Gamal Encryption

Public key encryption scheme based on DLP*, proposed by Taher Elgamal in 1985.*

public parameter: $g \in \mathbb{F}_p^*$

Alice

Bob ($b \in \mathbb{Z}$)

Alice		Bob
	$\xleftarrow{\quad B \quad}$	$B \leftarrow g^b$
$m \leftarrow \mathbb{F}_p^*$		
$k \xleftarrow{\$} \mathbb{Z}$		
$(c_1, c_2) \leftarrow (g^k, m \cdot B^k)$	$\xrightarrow{\quad (c_1, c_2) \quad}$	$m \leftarrow m = c_2 / c_1^b$

Security

- If DLP or CDH are easy, then the ElGamal system is insecure.
- It can be shown that the system is *CPA secure* if the *Decisional Diffie-Hellman problem* is hard as well.

Summary of Lecture 1

Caesar cipher and variants

- symmetric: Alice and Bob possess the same secret key
- substitution ciphers (letters are encrypted individually)
- historic, today insecure

Diffie-Hellman key exchange

- asymmetric: Alice and Bob have different secret keys
- based on modular exponentiation in a finite field $\mathbb{F}_p$
- security is based on the hardness of DLP and CDH

Next lecture

- How hard is the DLP?
- Babystep-giantstep, Pollard's rho and index calculus algorithm