CISC 468: CRYPTOGRAPHY

LESSON 2: BASIC CONCEPTS AND HISTORICAL CIPHERS

Furkan Alaca

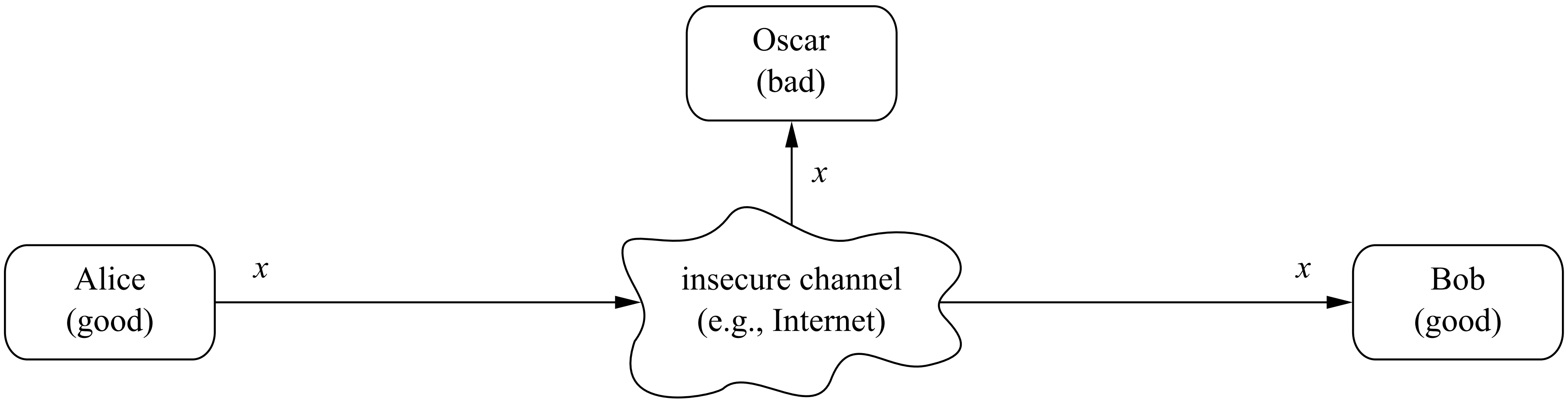
TODAY, WE WILL LEARN ABOUT...

1. Examples of historical ciphers
2. Different types of attacks against encryption
3. Security requirements for encryption
4. Why you should only well-established, well-scrutinized encryption algorithms

READINGS

- Section 1.2 (Symmetric Cryptography), Paar & Pelzl
- Section 1.3 (Cryptanalysis), Paar & Pelzl
- Section 1.4.3 (Caesar Cipher), Paar & Pelzl

COMMUNICATION OVER AN INSECURE CHANNEL

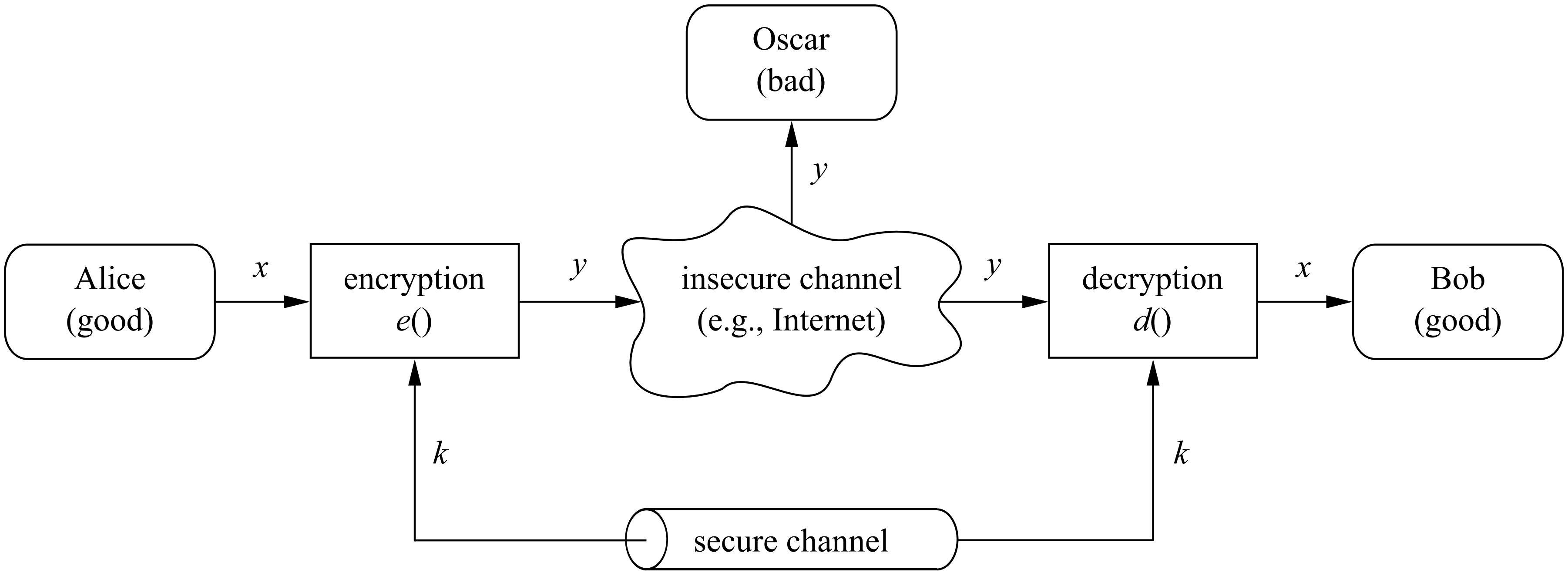


- *Goal:* Alice and Bob must securely exchange data
- *Problem:* The communication channel is insecure
  - An eavesdropper can intercept and read all data
- *Solution:* Before sending data, it must be transformed into a representation that is unintelligible to attackers

ENCRYPTION AND DECRYPTION

- *Encryption algorithms* transform data (*plaintext*) into a form (*ciphertext*) that is unintelligible to eavesdroppers
- This operation is reversible using the corresponding *decryption algorithm*

SYMMETRIC-KEY ENCRYPTION



- `e()` and `d()` are parametrized by a secret key `k`
- `k` must be agreed upon in advance between Alice and Bob over a secure channel, e.g., in-person meeting

CAESAR CIPHER

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

- Shift each plaintext letter by `k` positions to obtain the corresponding ciphertext letter; i.e., for $x, y, k \in \mathbb{Z}_{26}$,

$$e_k(x) \equiv x + k \mod 26,$$
$$d_k(y) \equiv y - k \mod 26.$$

- For `k=1`, the plaintext `HELLO` encrypts to `IFMMP`

ANALYZING THE SECURITY OF THE CAESAR CIPHER (1)

How many possible keys are there?

- The number of possible keys (i.e., key space) must be great enough that an attacker should not be able to guess it
  - How does the attacker know that they have guessed the correct key?

ANALYZING THE SECURITY OF THE CAESAR CIPHER (2)

Given a ciphertext, can you deduce any information about the plaintext?

- Given a ciphertext y but without knowing k, an attacker must be unable to deduce **any** information about the message x
- The ciphertext must be **indistinguishable from randomly-generated data**
- Does the Caesar Cipher have this property?

ANALYZING THE SECURITY OF THE CAESAR CIPHER (3)

Given a plaintext-ciphertext pair, can you deduce any information about the key?

- Given a plaintext-ciphertext pair `(x,y)` encrypted using `k`, an attacker should be unable to deduce any information about `k`
- Does the Caesar Cipher have this property?

SYMMETRIC-KEY CRYPTOSYSTEMS: DEFINITION

A symmetric-key cryptosystem can be defined by a five-tuple of finite sets $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where:

- $\mathcal{P}$ is the set of all possible plaintexts
- $\mathcal{C}$ is set of all possible ciphertexts
- $\mathcal{K}$ is the *key space*, or set of all possible keys
  - For the Caesar cipher, $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$
- $\mathcal{E} = \{E_k : k \in \mathcal{K}\}$ is a set of encryption functions parametrized by $k \in \mathcal{K}$ where $E_k : \mathcal{P} \to \mathcal{C}$
- $\mathcal{D} = \{D_k : k \in \mathcal{K}\}$ is a set of decryption functions parametrized by $k \in \mathcal{K}$ where $D_k : \mathcal{C} \to \mathcal{P}$
- For all $k \in \mathcal{K}$ and $x \in \mathcal{P}$, we must have $D_k(E_k(x)) = x$

POLYALPHABETIC SUBSTITUTION CIPHERS

- The Caesar Cipher is monoalphabetic, because it uses a single substitution rule over the entire message
- Polyalphabetic ciphers use multiple substitution rules
  - i.e., the same plaintext letter may be assigned different substitutes in the same message
  - Earliest discussion dates back to Al-Qalqashandi (1355-1418)
  - The Enigma machine used in World War II was also a polyalphabetic substitution cipher

VIGENÈRE CIPHER

- The Vigenère Cipher is a polyalphabetic substitution cipher, with $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_{26})^m$ and $m \in \mathbb{Z}, m > 1$
- For $k \in \mathcal{K}$, we have

$$E_k(x_1, \ldots, x_m) = (x_1 + k_1 \mod 26, \ldots, x_m + k_m \mod 26)$$
$$D_k(x_1, \ldots, x_m) = (x_1 - k_1 \mod 26, \ldots, x_m - k_m \mod 26)$$

- Equivalent to Caesar Cipher with a sequence of m keys
- For `k=BCDEF`, the plaintext `HELLO` encrypts to `IGOPT`
- What to do to encrypt plaintext that is longer than m letters?
  - What kind of attacks arise? Hint

ANALYZING THE SECURITY OF THE VIGENÈRE CIPHER

1. How many possible keys are there?
2. Given a ciphertext, can you deduce any information about the plaintext?
3. Given a plaintext-ciphertext pair, can you deduce any information about the key?

HOW LARGE OF A KEY SPACE IS LARGE ENOUGH?

Key Size (bits)	Key Space (Number of Keys)	Cracking Time @ 10^6 trials/sec	Cracking Time @ 10^{18} trials/sec
32	$2^{32} = 4.3 \times 10^9$	$\sim$ 35 mins	$\sim$ 2 picoseconds
56	$2^{56} = 7.2 \times 10^{16}$	$\sim$ 1000 years	$\sim$ 36 milliseconds
128	$2^{128} = 3.4 \times 10^{38}$	$\sim 10^{24}$ years	$\sim 1.7 \times 10^{20}$ years

- Average time to brute-force a key: $\frac{1}{2} \times \frac{\text{Number of keys}}{\text{Trials/sec}}$
- Bitcoin network hash rate: $\sim 10^{20}$ Hashes/sec
- Age of the universe: 1.38×10^{10} years

SECURITY OF CRYPTOSYSTEMS: GOALS

1. *Semantic security:* Given a ciphertext, any probabilistic polynomial-time algorithm cannot obtain any non-negligible information about the plaintext.
2. *Indistinguishability:* Given two plaintexts and one ciphertext, any adversary cannot make a better-than-random guess as to which of the two plaintexts was encrypted.
3. *Non-malleability:* Given a ciphertext corresponding to a plaintext, any adversary cannot construct a second ciphertext of a plaintext that is meaningfully related to the first.

SECURITY OF CRYPTOSYSTEMS: ATTACK MODELS

- Ciphertext-only attack: Attacker has a collection of ciphertext
- Known-plaintext attack: Attacker has a collection of ciphertext-plaintext pairs
- Chosen plaintext attack: Attacker can obtain ciphertext corresponding to any plaintext of their choice
- Chosen ciphertext attack: Attacker can obtain plaintext corresponding to any ciphertext of their choice

WHAT DO WE MEAN BY "SECURE" ALGORITHMS? (1)

- An algorithm that is *information-theoretically secure* is **impossible** to break, even with unlimited computing power
  - Theoretically possible, but impractical for real-world use

WHAT DO WE MEAN BY "SECURE" ALGORITHMS? (2)

- An algorithm that is *computationally secure* is **infeasible** to break for a computationally bounded adversary
  - Practically feasible, thanks to the computational difficulty of inverting various mathematical or algorithmic operations, e.g., factoring large numbers (RSA)

SECURITY BY DESIGN VS. SECURITY THROUGH OBSCURITY

> *A cryptosystem should be secure even if an attacker knows everything about the system, with the exception of the secret key.*

Kerckhoffs' Principle

> *The enemy knows the system.*

Claude Shannon

DESIGN PRINCIPLE P3: OPEN-DESIGN

- Don't rely on secret designs or attacker ignorance
- Cryptographic algorithms are typically standardized following extensive scrutiny via global competitions
- The only secret should be the cryptographic key
  - However, avoid disclosing information that can lead to key compromise (e.g., timing data that may aid cryptanalysis)

DESIGN PRINCIPLE P9: TIME-TESTED-TOOLS

- Golden rule of crypto: Don't roll your own crypto
  - Don't invent your own cryptographic algorithm
  - Don't write your own cryptographic library
  - ... Assuming you are not a cryptographer
- Widely-used, heavily-scrutinized mechanisms are less likely to retain flaws
- Use highly-vetted libraries like OpenSSL
  - Open-source may not be a silver bullet, but it has advantages

RECAP

- At a high level, in order for a symmetric-key cryptosystem to be secure it must:
  - Offer a large enough keyspace that is computationally infeasible to do an exhaustive search on
  - Generate ciphertext that is computationaly infeasible to distinguish from random data
    - Should not leak information about the plaintext or key
- It is computationally infeasible to do an exhaustive search on a keyspace of 128 bits
- Don't roll your own crypto — rely instead on publicly-scrutinized and time-tested algorithms