HTCS6702 - Cryptography

Introduction

These slides are partially based on slides from J. Katz, author of text book: *Introduction to Modern Cryptography*

Goals

- Understand real-world crypto via a rigorous approach
- When you encounter crypto in your career:
  - Understand the key terms
  - Understand the security guarantees provided
  - Know how to use crypto
  - Understand what goes on “under the hood”
- “Crypto mindset”

Non-goals

- Designing your own crypto schemes
- Implementing your own crypto for real-world use

- Course goal:
  realize when to consult an expert!

Cryptography (historically)

"…the art of writing or solving codes…"

- Historically, cryptography focused exclusively on ensuring *private communication* between two parties sharing secret information in advance (using "codes" aka *private-key encryption*)

Modern cryptography

- Much broader scope!
  - Data integrity, authentication, protocols, …
  - The *public-key setting*
  - Group communication
  - More-complicated trust models
  - Foundations (e.g., number theory, quantum-resistance) to systems (e.g., electronic voting, cryptocurrencies)

Modern cryptography

*Design, analysis, and implementation of **mathematical techniques** for securing information, systems, and distributed computations against adversarial attack*

Modern cryptography

- Cryptography is ubiquitous
  - Passwords, password hashing
  - Secure credit-card transactions over the internet
  - Encrypted WiFi
  - Disk encryption
  - Digitally signed software updates
  - Bitcoin
  - …

Cryptography (historically)

"…the art of writing or solving codes…"

- Historically, cryptography was an *art*
  - Heuristic, unprincipled design and analysis
  - Schemes proposed, broken, repeat…

Modern cryptography

- Cryptography is now much more of a *science*
  - Rigorous analysis, firm foundations, deeper understanding, rich theory

- The “crypto mindset” has permeated other areas of computer security
  - Threat modeling
  - Proofs of security

Rough course outline

	Secrecy	Integrity
Private-key setting	Private-key encryption	Message authentication codes
Public-key setting	Public-key encryption	Digital signatures

- Building blocks
  - Pseudorandom (number) generators
  - Pseudorandom functions/block ciphers
  - Hash functions
  - Number theory


Unitec
Institute of Technology
TE WHARE WĀNANGA O WAIRAKA
HTCS6702 - Cryptography
Classical Cryptography

Motivation

- Allows us to “ease into things…,” introduce notation
- Shows why unprincipled approaches are dangerous
- Illustrates why things are more difficult than they may appear

Classical cryptography

- Until the 1970s, exclusively concerned with ensuring *secrecy* of communication

- I.e., *encryption*

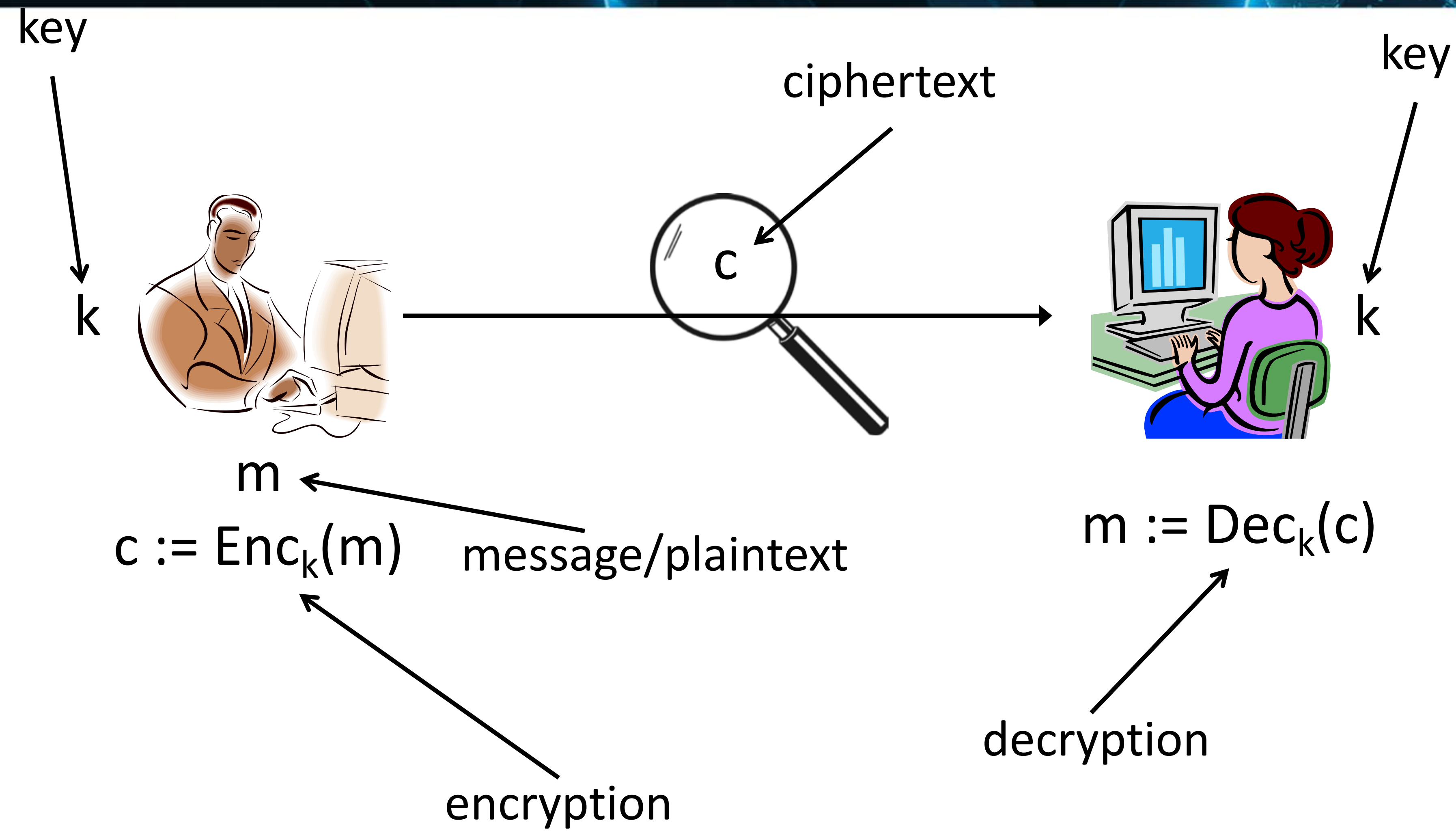
Classical cryptography

- Until the 1970s, relied exclusively on secret information (a *key*) shared in advance between the communicating parties

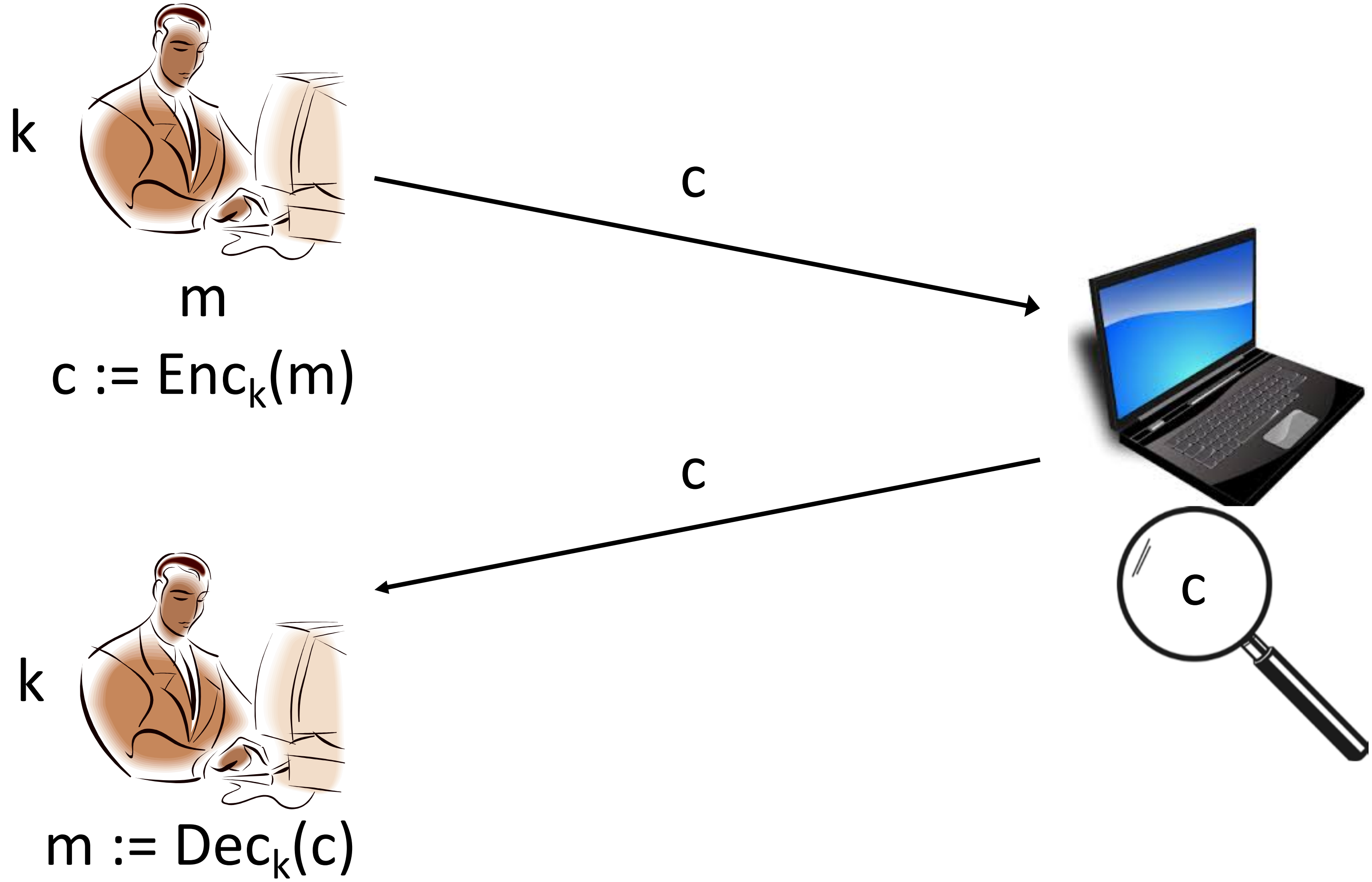
Private-key cryptography

  – aka secret-key / shared-key / symmetric-key cryptography

Private-key encryption



Private-key encryption



Private-key encryption

- A *private-key encryption scheme* is defined by a message space $\mathcal{M}$ and algorithms (Gen, Enc, Dec):
  - Gen (key-generation algorithm): outputs $k \in \mathcal{K}$
  - Enc (encryption algorithm): takes key k and message $m \in \mathcal{M}$ as input; outputs ciphertext c

$$c \leftarrow \mathrm{Enc}_k(m)$$

  - Dec (decryption algorithm): takes key k and ciphertext c as input; outputs m or "error"

$$m := \mathrm{Dec}_k(c)$$

For all $m \in \mathcal{M}$ and k output by Gen,

$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$$

Kerckhoffs's principle

- *The encryption scheme* is not secret
  - The attacker knows the encryption scheme
  - The only secret is the *key*
  - The key must be chosen at random; kept secret

- Some arguments in favor of this principle
  - Easier to keep *key* secret than *algorithm*
  - Easier to change *key* than to change *algorithm*
  - Standardization
    - Ease of deployment
    - Public validation

The shift cipher

- Consider encrypting English text
- Associate ‘a’ with 0; ‘b’ with 1; …; ‘z’ with 25
- $k \in \mathcal{K} = \{0, \ldots, 25\}$
- To encrypt using key k, shift every letter of the plaintext by k positions (with wraparound)
- Decryption just does the reverse

```
helloworld
jgnnqyqtnf
```

Modular arithmetic

- x = y mod N if and only if N divides x-y
- [x mod N] = the remainder when x is divided by N
  - I.e., the unique value $y \in \{0, \ldots, N-1\}$ such that x = y mod N

- 25 = 35 mod 10
- 25 ≠ [35 mod 10]
- 5 = [35 mod 10]

The shift cipher, formally

- $\mathcal{M}$ = {strings over lowercase English alphabet}
- Gen: choose uniform $k \in \{0, \ldots, 25\}$
- $\mathrm{Enc}_k(m_1 \ldots m_t)$: output $c_1 \ldots c_t$, where
  $$c_i := [m_i + k \bmod 26]$$
- $\mathrm{Dec}_k(c_1 \ldots c_t)$: output $m_1 \ldots m_t$, where
  $$m_i := [c_i - k \bmod 26]$$

- Can verify that correctness holds…

Is the shift cipher secure?

- No -- only 26 possible keys!
  - Given a ciphertext, try decrypting with every possible key
  - Only one possibility will “make sense”
  - (What assumptions are we making here?)

- Example of a “brute-force” or “exhaustive-search” attack

Example

- Ciphertext `uryybjbeyq`
- Try every possible key…
  - `tqxxaiadxp`
  - `spwwzhzcwo`
  - …
  - `helloworld`