Seguridad Informatica - Fall 2024

Module IV: Physical security

Lecture 1

Introduction

Marco Guarnieri
IMDEA Software Institute

Organization

- ***3 classes***
  - December 2nd 12:00—14:00 (Monday)
  - December 9th 12:00—14:00 (Monday)
  - December 16th 12:00—14:00 (Monday)

- Each lecture is complemented by ***reading assignments***

  - ***Read*** (and try to understand) ***them!***

- Final ***exam***: January 13 at 11:00

Organization

- Lecture 1 — Introduction
- Lecture 2 — Cache-based side channel attacks
- Lecture 3 — Speculative execution attacks
- Lecture 4 — Non-interference
- Lecture 5 — Automated detection of speculative leaks

Organization

December 2nd

- Lecture 1 — Introduction
- Lecture 2 — Cache-based side channel attacks
- Lecture 3 — Speculative execution attacks
- Lecture 4 — Non-interference
- Lecture 5 — Automated detection of speculative leaks

Organization

- Lecture 1 — Introduction

December 9th

- Lecture 2 — Cache-based side channel attacks
- Lecture 3 — Speculative execution attacks
- Lecture 4 — Non-interference
- Lecture 5 — Automated detection of speculative leaks

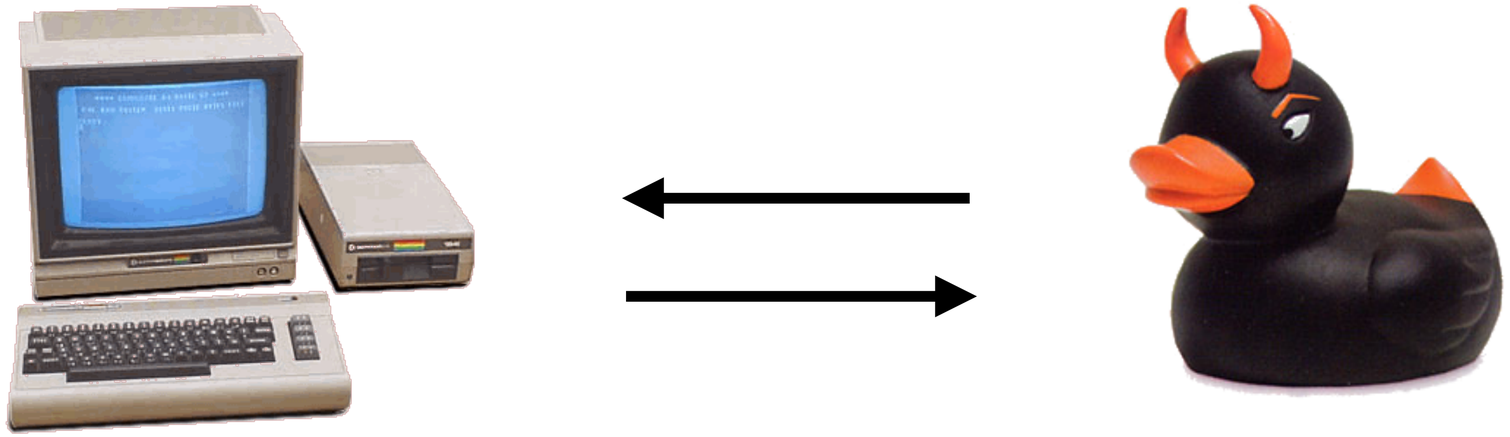
Organization

- Lecture 1 — Introduction
- Lecture 2 — Cache-based side channel attacks
- Lecture 3 — Speculative execution attacks

December 16th

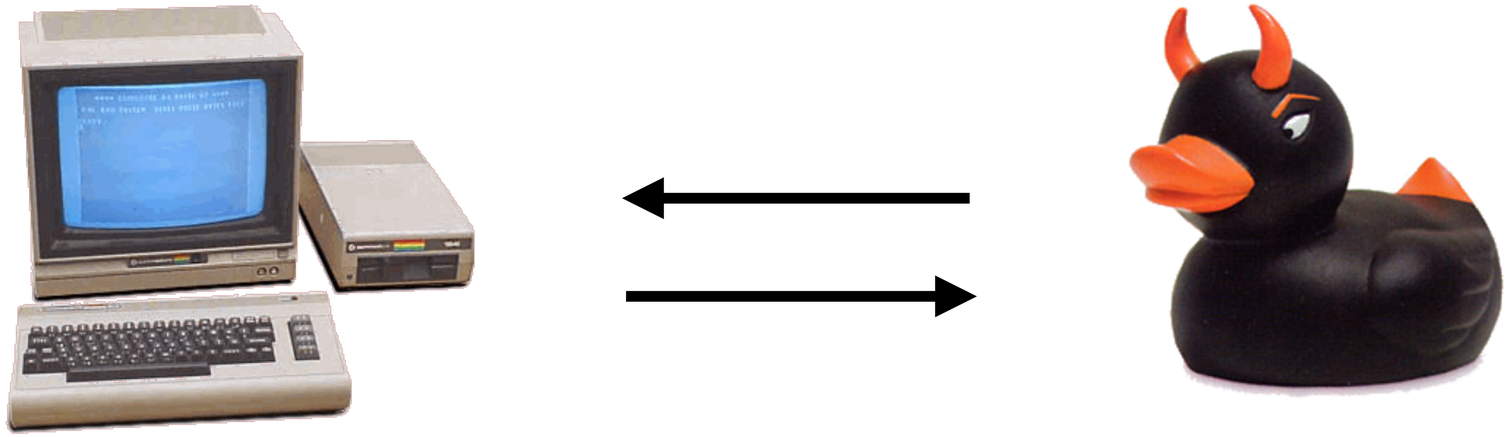
- Lecture 4 — Non-interference
- Lecture 5 — Automated detection of speculative leaks

Introduction



1. In Module 2, you studied how to analyze programs for ***security bugs***

Introduction



1. In Module 2, you studied how to analyze programs for ***security bugs***
2. In Module 3, you studied adversaries that can ***(ab)use logical interfaces*** to break into systems

Introduction

1. In Module 2, you studied how to analyze programs for ***security bugs***
2. In Module 3, you studied adversaries that can ***(ab)use logical interfaces*** to break into systems
3. In this module we will consider adversaries that take into account the ***physics of computation***

Introduction

1. In Module 2, you studied how to analyze programs for ***security bugs***
2. In Module 3, you studied adversaries that can ***(ab)use logical interfaces*** to break into systems
3. In this module we will consider adversaries that take into account the ***physics of computation***

Introduction

1. In Module 2, you studied how to analyze programs for ***security bugs***
2. In Module 3, you studied adversaries that can ***(ab)use logical interfaces*** to break into systems
3. In this module we will consider adversaries that take into account the ***physics of computation***

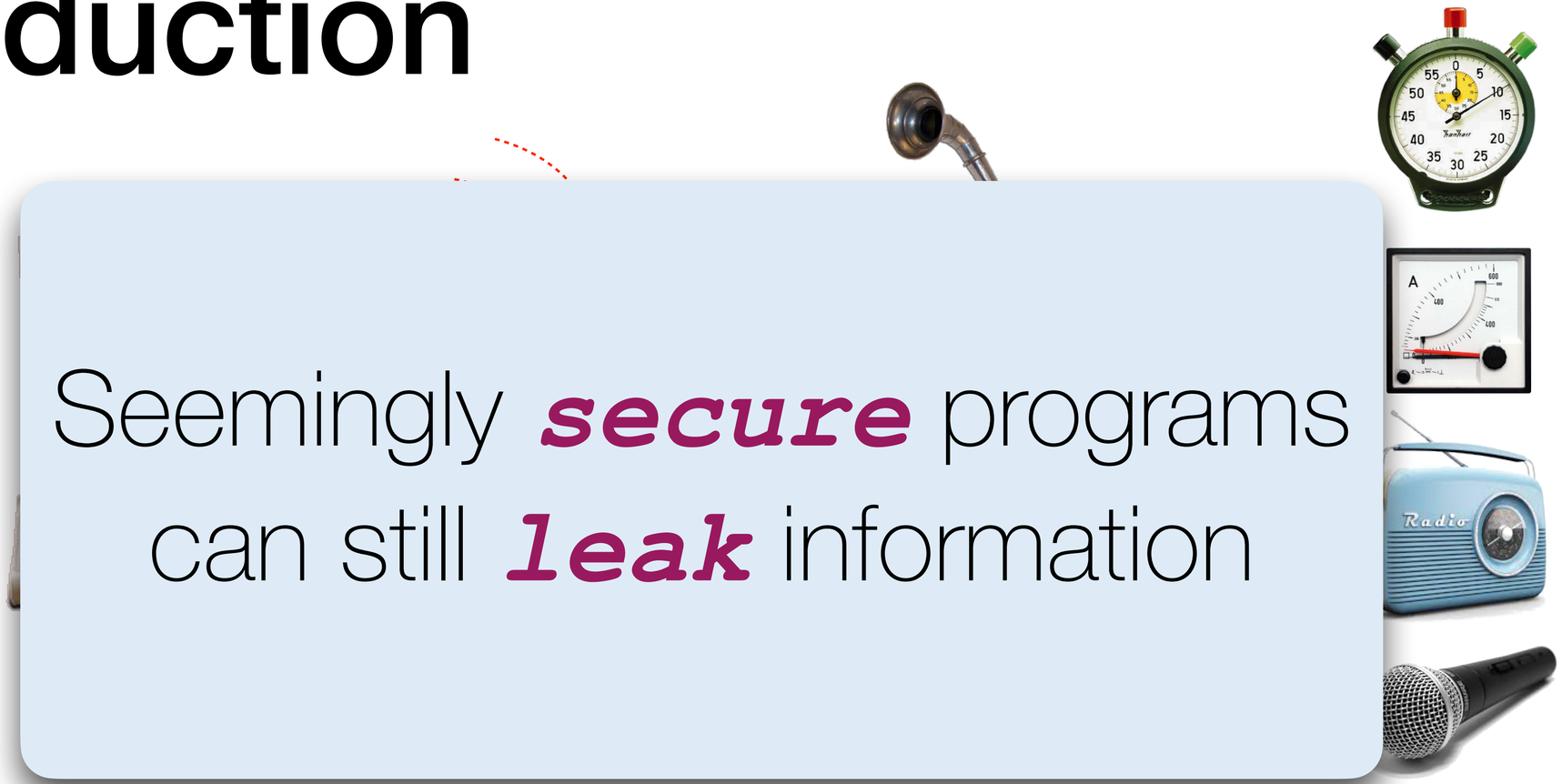
Introduction

1. In Module 2, you studied how to analyze programs for ***security bugs***
2. In Module 3, you studied adversaries that can ***(ab)use logical interfaces*** to break into systems
3. In this module we will consider adversaries that take into account the ***physics of computation***

Introduction

1. In Module 2, you studied how to analyze programs for ***security bugs***
2. In Module 3, you studied adversaries that can ***(ab)use logical interfaces*** to break into systems
3. In this module we will consider adversaries that take into account the ***physics of computation***

Introduction

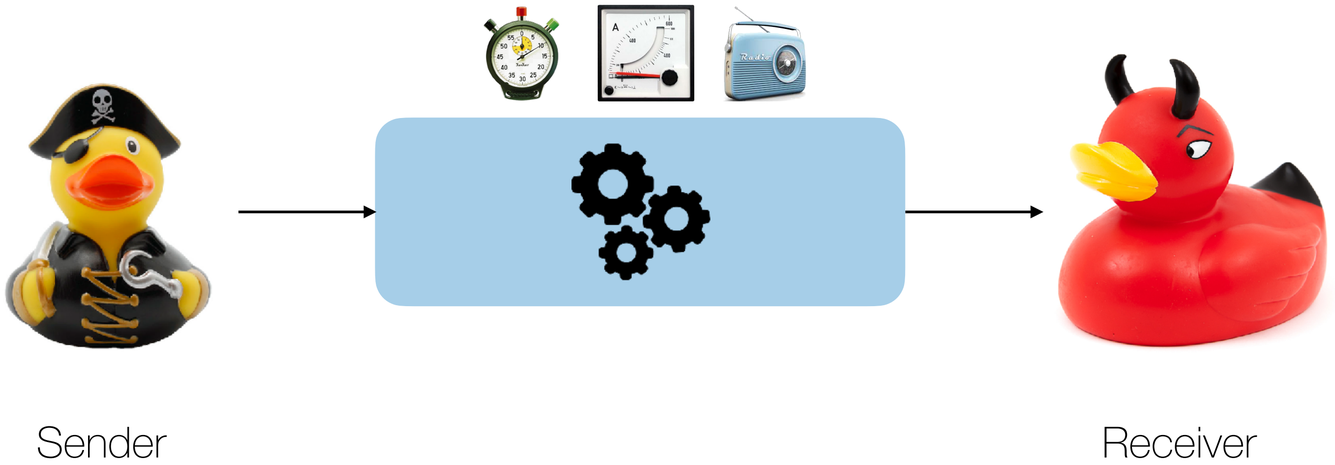


1. In Module 2, you studied how to analyze programs for ***security bugs***
2. In Module 3, you studied adversaries that can ***(ab)use logical interfaces*** to break into systems
3. In this module we will consider adversaries that take into account the ***physics of computation***

Covert & side channels

Ducks from https://shop.justducks.co.uk/

Covert & side channels



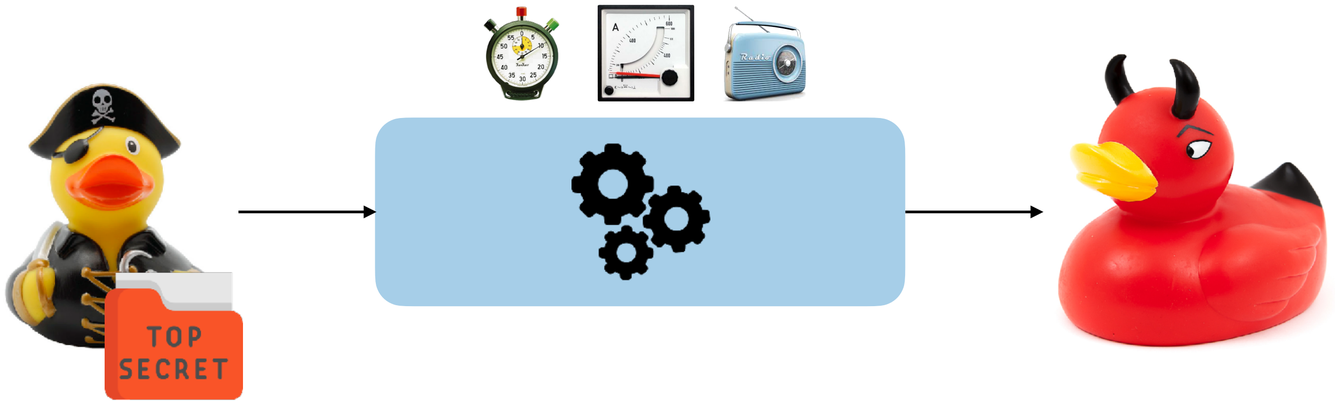
Sender

Receiver

Covert channel: Two collaborating parties exchange information over a ***channel not intended for communication***

Ducks from https://shop.justducks.co.uk/

Covert & side channels



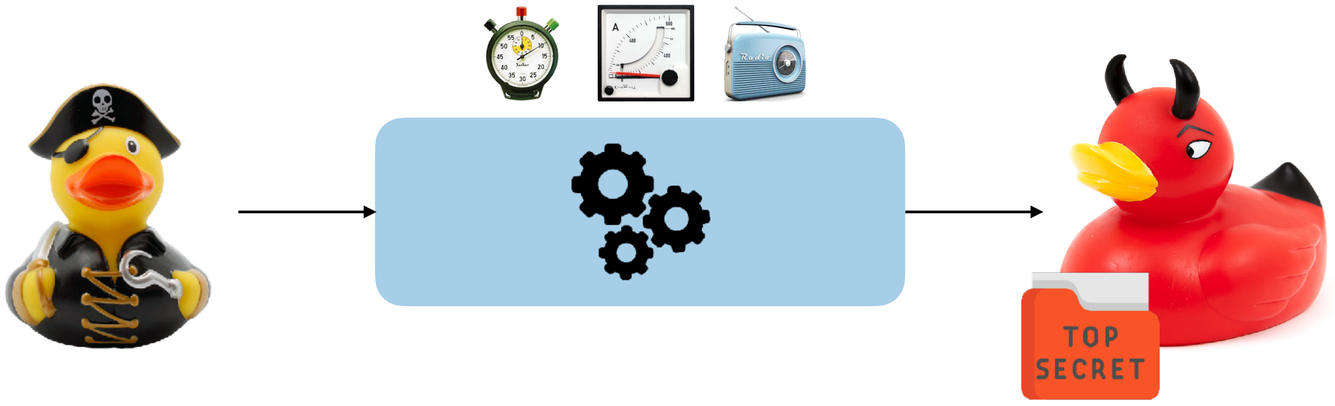
Sender

Receiver

Covert channel: Two collaborating parties exchange information over a ***channel not intended for communication***

Ducks from https://shop.justducks.co.uk/

Covert & side channels



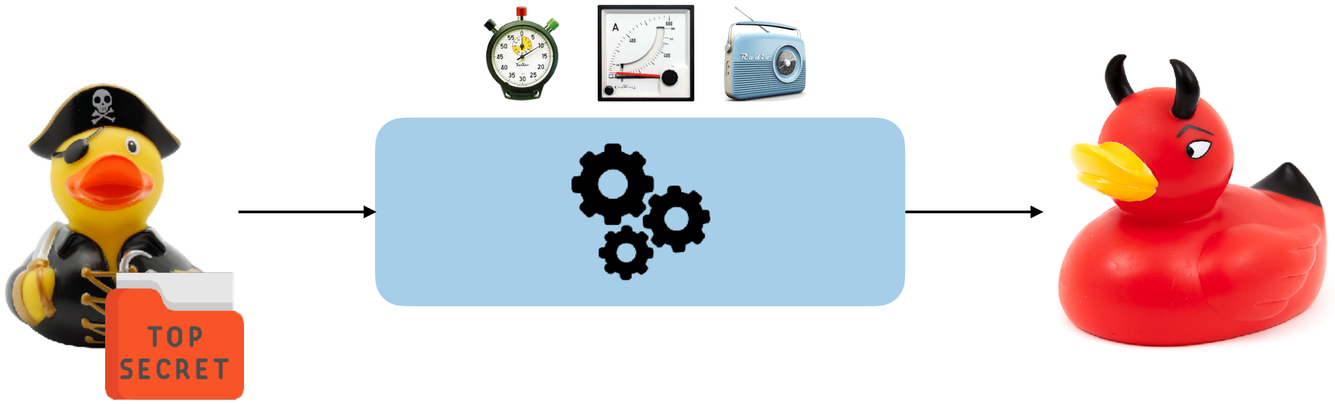
Sender

Receiver

Covert channel: Two collaborating parties exchange information over a ***channel not intended for communication***

Ducks from https://shop.justducks.co.uk/

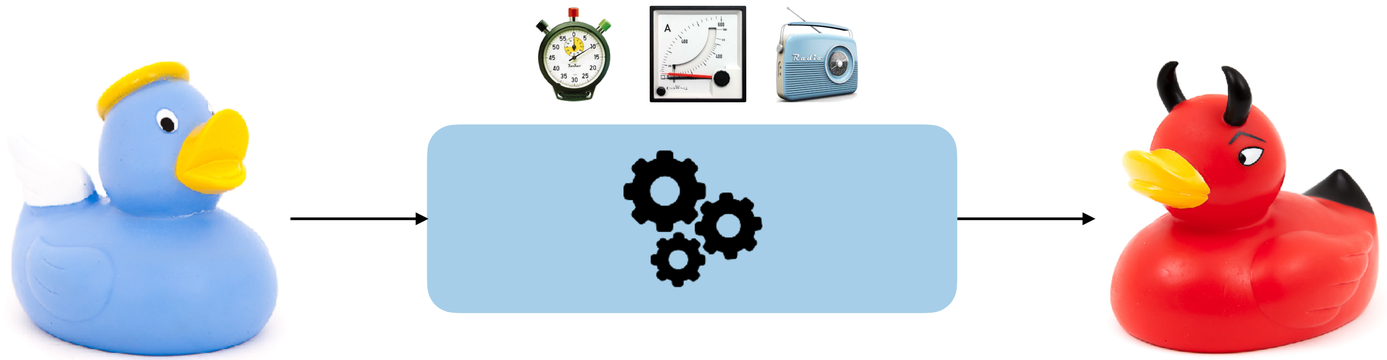
Covert & side channels



Sender

Receiver

Covert channel: Two collaborating parties exchange information over a ***channel not intended for communication***



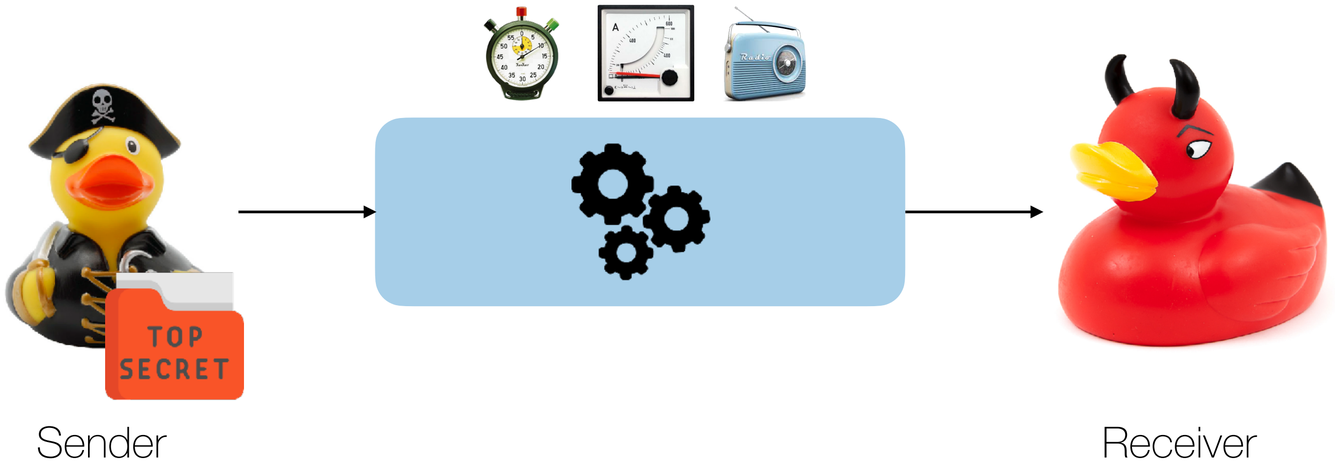
Observee

Observer

Side channel: Attacker extracts information about a victim's computation by observing a ***channel not intended for communication***

Ducks from https://shop.justducks.co.uk/

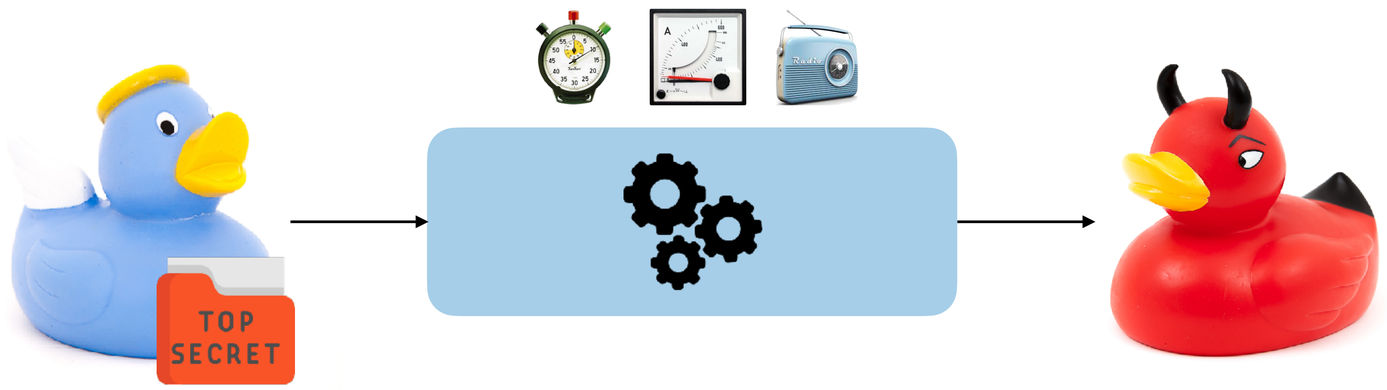
Covert & side channels



Sender

Receiver

Covert channel: Two collaborating parties exchange information over a ***channel not intended for communication***



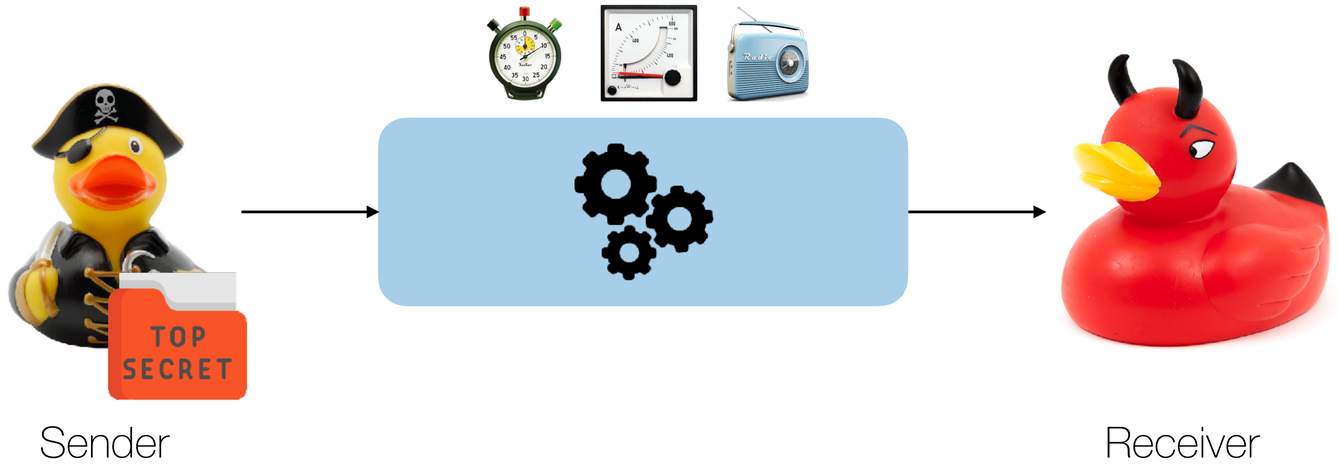
Observee

Observer

Side channel: Attacker extracts information about a victim's computation by observing a ***channel not intended for communication***

Ducks from https://shop.justducks.co.uk/

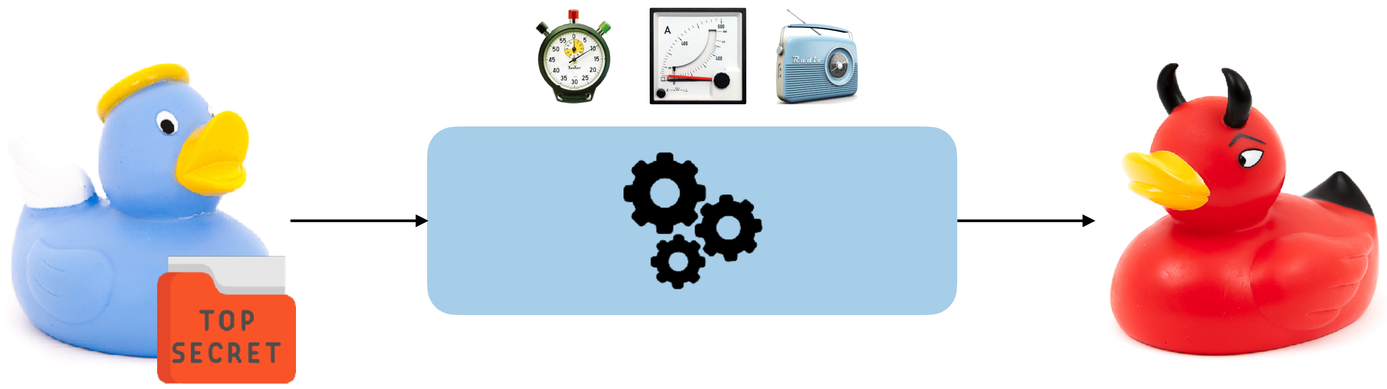
Covert & side channels



Sender

Receiver

Covert channel: Two collaborating parties exchange information over a ***channel not intended for communication***



Observee

Observer

Side channel: Attacker extracts information about a victim's computation by observing a ***channel not intended for communication***

Ducks from https://shop.justducks.co.uk/

Ex: Control-flow side channels

Input: key `k`, ciphertext `c`

```
x:=1
for i=0 to |k|-1
    x:=x*x mod n
    if k[i]==1 then
        x:=x*c mod n
return x
```

RSA decryption
c^k mod n

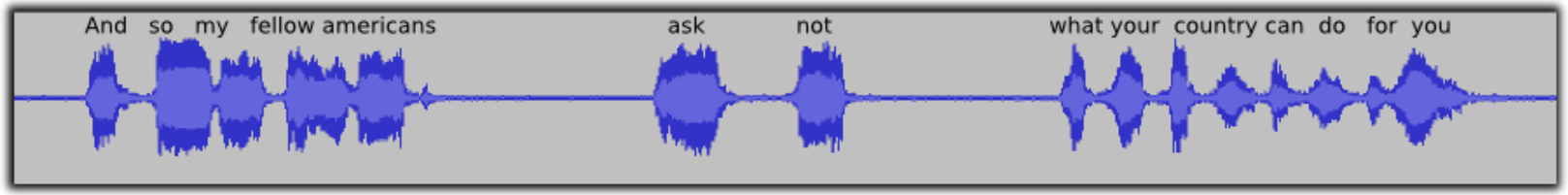
- ***First attack:*** Kocher (1996) shows how secret keys can be recovered from timing measurements
- ***Remote attack:*** Boneh & Brumley (2003) recover full keys from standard OpenSSL RSA implementation over network

Ex: Cache-based Side Channel

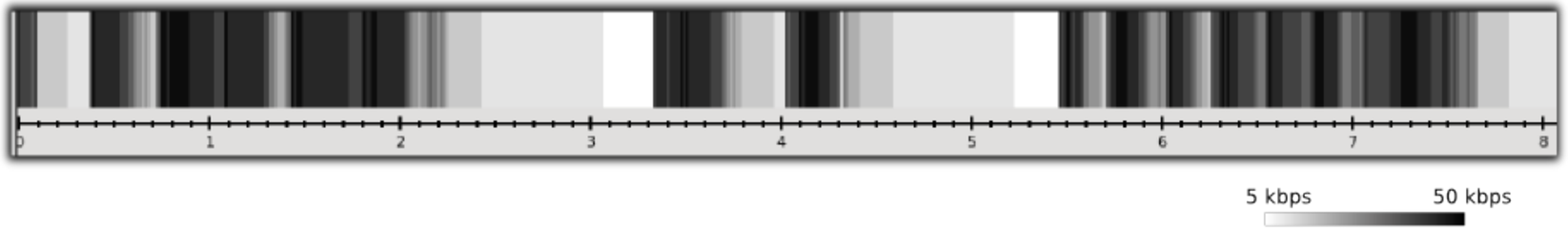
Ex: Speculative execution attacks

Ex: Traffic Side Channel

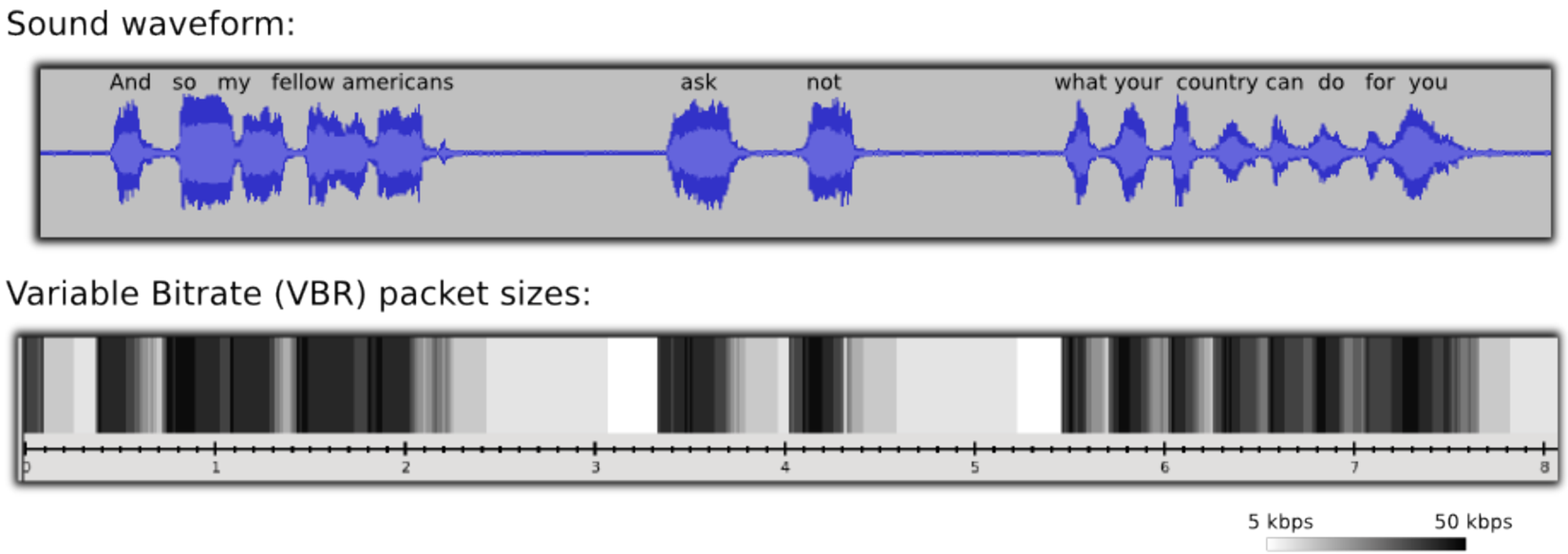
Sound waveform:



Variable Bitrate (VBR) packet sizes:



Ex: Traffic Side Channel

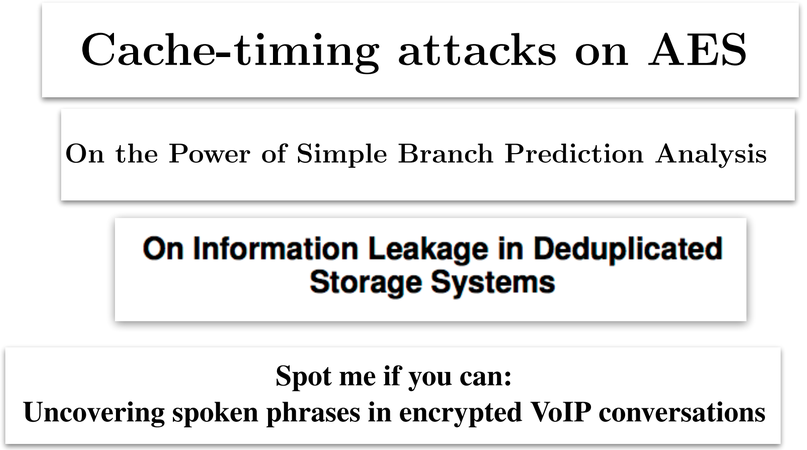


Has been exploited for identifying ***language*** (Wright '07), ***phrases*** (Wright '08), ***speakers*** (Backes '09), ***conversations*** (White '11)

Observation

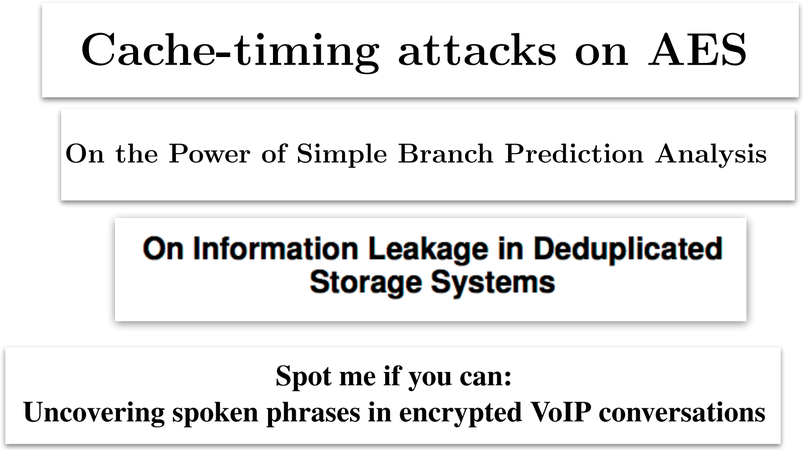
Observation

- Side-channels often introduced by ***performance-enhancing techniques***
  - Caching
  - Speculative execution
  - Deduplication
  - Compression
  - …



Observation

- Side-channels often introduced by ***performance-enhancing techniques***
  - Caching
  - Speculative execution
  - Deduplication
  - Compression
  - …



- Pattern or coincidence?
  - Performance-enhancing techniques reduce resource consumption ***on average***, but not in the ***worst case***.
  - This introduces ***variations into the consumption profile*** that can be exploited by adversaries, i.e., side-channels.

Closing side-channels, radically

CPUs
Compilers
Virtual machines
Networks
…

minimize

time
space
energy
…

consumption

Closing side-channels, radically

CPUs
Compilers
Virtual machines
Networks
…

flatten
~~minimize~~

time
space
energy

consumption

Dramatic performance penalties