

The background of the slide is an aerial photograph of the EPFL campus in Lausanne, Switzerland, taken at sunset. The sun is low on the horizon, casting a warm orange glow over the city and the surrounding Lake Léman. The campus buildings, including the iconic circular building, are visible in the foreground and middle ground. The lake and distant mountains are in the background.

# Privacy by design in an agile world

Wouter Lueks  
SPRING Lab  
EPFL

21.10.2020

# A collaborative (continued) sprint

March 2020 - **Start**

April 2020 – **GAEN is announced**

May 2020 – **Final version DP3T**

June 2020 – **Pilot SwissCovid  
(& other EU apps)**

July 2020 – **SwissCovid launch**

August / Sept 2020 – **Towards  
international interoperability**

## Decentralized Privacy-Preserving Proximity Tracing

Version: 25 May 2020.

Contact the first author for the latest version.

**EPFL:** Prof. Carmela Troncoso, Prof. Mathias Payer, Prof. Jean-Pierre Hubaux, Prof. Marcel Salathé, Prof. James Larus, Prof. Edouard Bugnion, Dr. Wouter Lueks, Theresa Stadler, Dr. Apostolos Pyrgelis, Dr. Daniele Antonioli, Ludovic Barman, Sylvain Chatel

**ETHZ:** Prof. Kenneth Paterson, Prof. Srdjan Čapkun, Prof. David Basin, Dr. Jan Beutel, Dr. Dennis Jackson, Dr. Marc Roeschlin, Patrick Leu

**KU Leuven:** Prof. Bart Preneel, Prof. Nigel Smart, Dr. Aysajan Abidin

**TU Delft:** Prof. Seda Gürses

**University College London:** Dr. Michael Veale

**CISPA:** Prof. Cas Cremers, Prof. Michael Backes, Dr. Nils Ole Tippenhauer

**University of Oxford:** Dr. Reuben Binns

**University of Torino / ISI Foundation:** Prof. Ciro Cattuto

**Aix Marseille Univ, Université de Toulon, CNRS, CPT:** Dr. Alain Barrat

**IMDEA Software Institute:** Prof. Dario Fiore

**INESC TEC:** Prof. Manuel Barbosa (FCUP), Prof. Rui Oliveira (UMinho), Prof. José Pereira (UMinho)



# The protocol... A small piece in the puzzle

**Interdisciplinary team  
(30+ researchers, 10 countries)**

Privacy, Systems, Cryptography,  
Wireless security, SW Security,  
Requirements engineering,  
Epidemiologists, Ethicists, Law

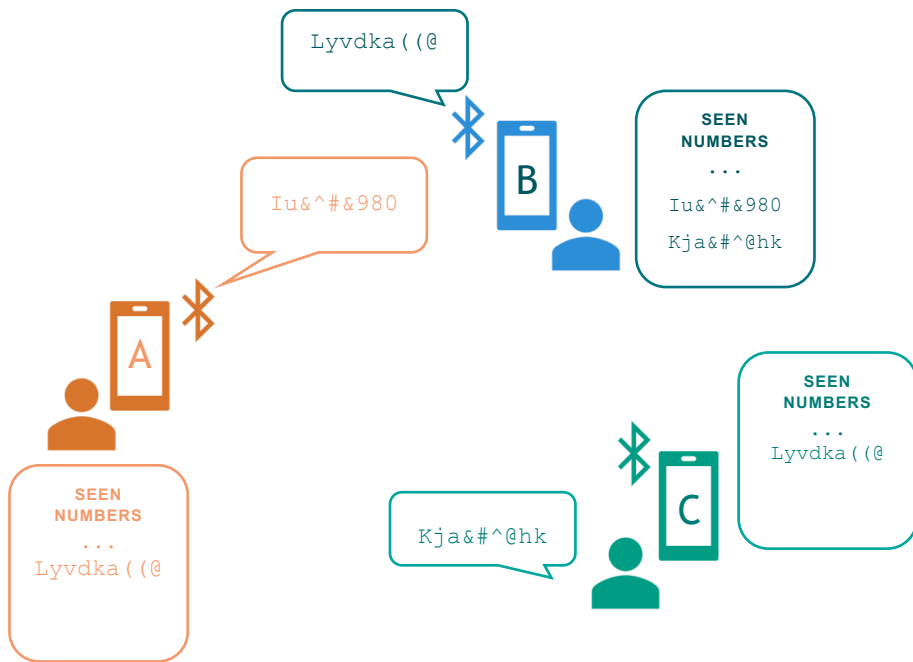
**EPFL**

**ETH zürich**



# How it works

## Walking around



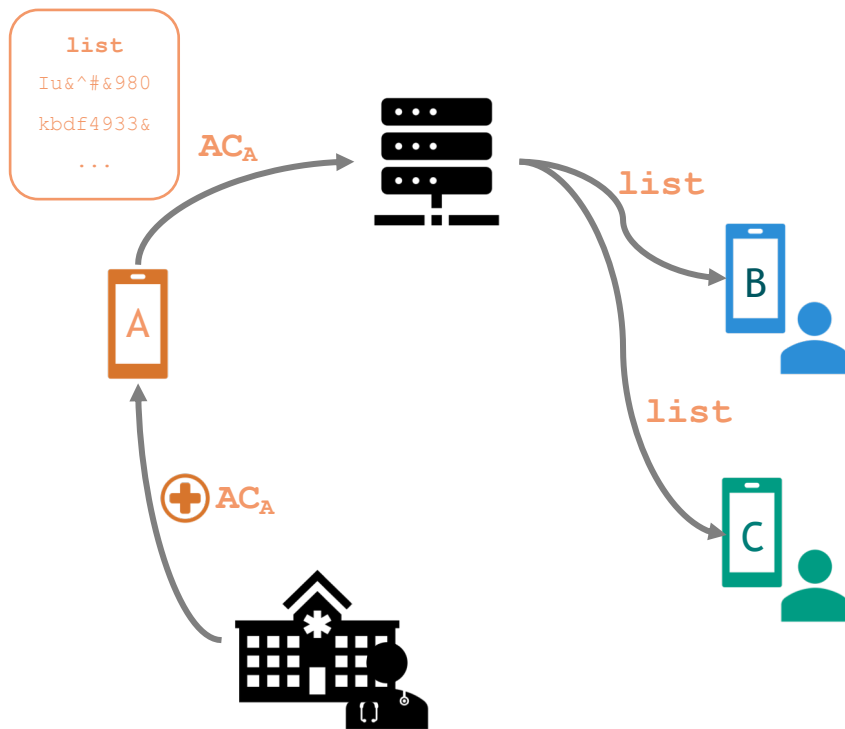
When a phone with the app hears a random identifier from a nearby app, it records having seen that number.

- **A** is nearby **B**: records **B**'s number
- **B** is nearby **A** and **C**: records **A,C**'s number
- **C** is nearby **B**: records **B**'s number



# How it works

## Exposure notification



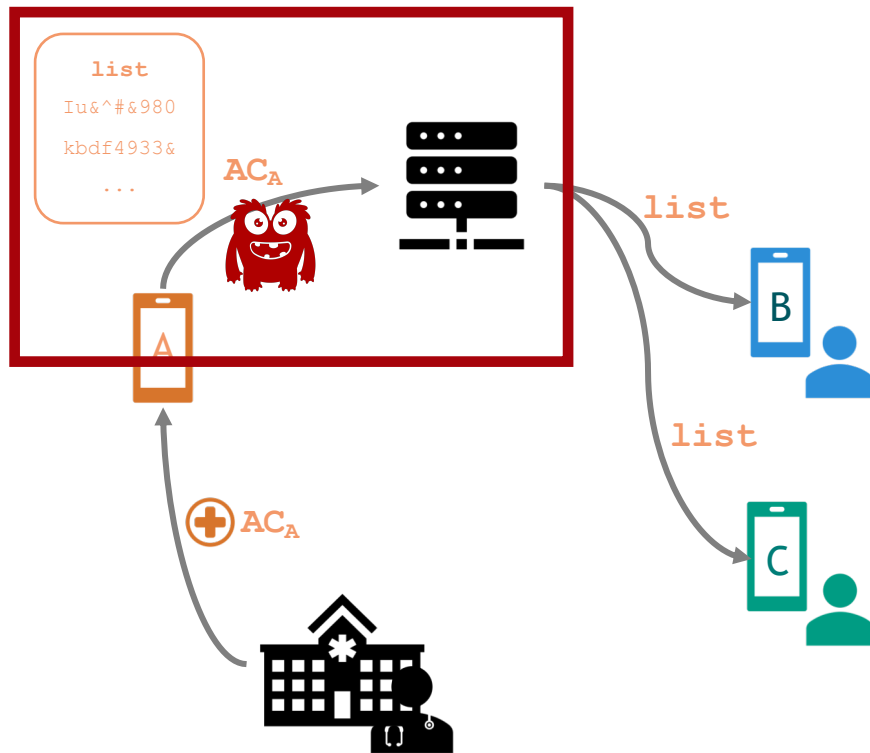
A user with a positive diagnosis:

- Uploads their own list
- Other phones download this list, and compare against stored identifiers.

# Focus: protecting who is SARS-CoV-2 positive

# How it works

## Notification

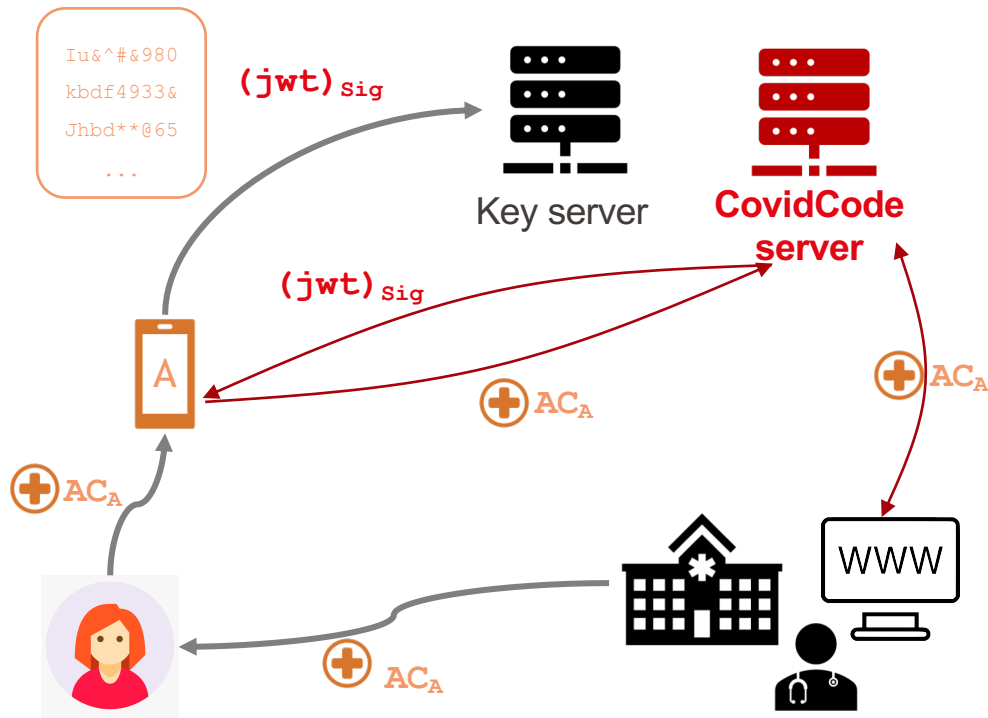


A user with a positive diagnosis:

- Uploads their own list
- Other phones download this list, and compare against stored identifiers.

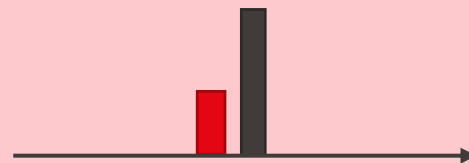
# Design

# How it works: Notification Reality

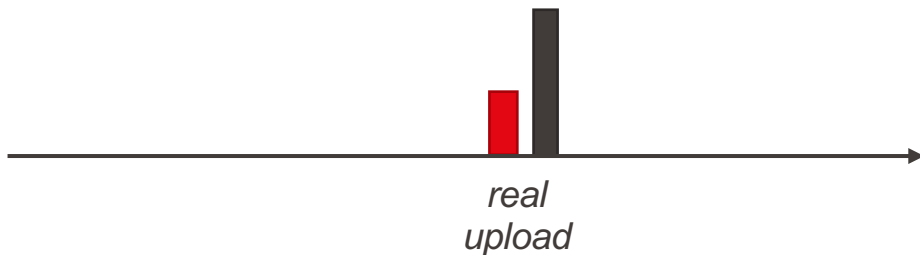


## What a network adversary sees:

- A request to the **CovidCode** server; then
- A request to the **Key** server







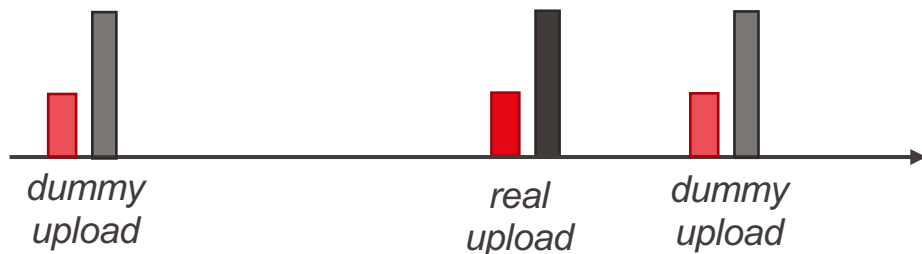
- Request to **CovidCode** server
- Request to **Key** server

**Add randomly timed dummy uploads so that observing an upload does not imply a positive diagnosis.**

These dummy uploads must:

- Have the same size,
- Sequence,
- Timing

# Dummy uploads



- Request to **CovidCode** server
- Request to **Key** server

Add randomly timed dummy uploads so that observing an upload does not imply a positive diagnosis.

These dummy uploads must be **indistinguishable** from real uploads:

- Have the same size,
- Sequence,
- Timing

# Design meets the development lifecycle

# Quiz: Is this a problem?

**Proposal:** Whenever the user opens the app, the app will retrieve the latest public announcement messages.

# Quiz: Is this a problem?

**Proposal:** Whenever the user opens the app, the app will retrieve the latest public announcement messages.



- Retrieve **announcements**
- Request to **CovidCode** server
- Request to **Key** server

**Add randomly timed dummy uploads so that observing an upload does not imply a positive diagnosis.**

Dummy uploads must be **indistinguishable** from real uploads

- Have the same size,
- **Sequence**,
- **Timing**



# Quiz: Is this a problem?

**Proposal:** Whenever the user opens the app, the app will retrieve the latest public announcement messages.



- Retrieve **announcements**
- Request to **CovidCode** server
- Request to **Key** server

**Add randomly timed dummy uploads so that observing an upload does not imply a positive diagnosis.**

Dummy uploads must be **indistinguishable** from real uploads

- Have the same size,
- **Sequence**,
- **Timing**

# Quiz: Is this a problem?

**Proposal:** Whenever the user opens the app, the app will retrieve the latest public announcement messages.



- Retrieve **announcements**
- Request to **CovidCode** server
- Request to **Key** server

Yes, this is a problem: Retrieve announcements in the background instead.

**Add randomly timed dummy uploads so that observing an upload does not imply a positive diagnosis.**

These dummy uploads must:

- Have the same size,
- **Sequence**,
- **Timing**

Cryptographic protocol is only a (small) part of the design

Deployed systems are **more complex and more agile**, impacting privacy-friendly designs.

Need to **iterate** with development teams to ensure properties remain.

Innocent-looking changes can have big consequences for privacy properties

Cryptographic protocol is only a (small) part of the design

Deployed systems are **more complex** and **more agile**, impacting privacy-friendly designs.

Need to **iterate** with development teams to ensure properties remain.

## Privacy Engineering Meets Software Engineering. On the Challenges of Engineering Privacy By Design

Blagovesta Kostova  
EPFL  
blagovesta.pirelli@epfl.ch

Seda Gürses  
TU Delft / KU Leuven  
f.s.gurses@tudelft.nl

Carmela Troncoso  
EPFL  
carmela.troncoso@epfl.ch

Innocent-looking changes can have big consequences for privacy properties

**Abstract**—Current day software development relies heavily on the use of service architectures and on agile iterative development methods to design, implement, and deploy systems. These practices result in systems made up of multiple services that introduce new data flows and evolving designs that escape the control of a single designer. Academic privacy engineering literature typically abstracts away such conditions of software production in order to achieve generalizable results. Yet, through a systematic study of the literature, we show that proposed solutions inevitably make assumptions about software architectures, development methods and scope of designer control that are misaligned with current practices. These misalignments are likely to pose an obstacle to operationalizing privacy engineering solutions in the wild.



Fig. 1: Privacy-preserving billing: research view vs. software practice. Legend:  $\mathcal{M}$  is a development team,  $\mathcal{D}$  is the device that communicates with the System,  $m_i$  is the measurements from the device,  $id$  is the id to identify the device. Grey elements are outside of the control of the main dev team, whereas black elements indicates what is under their control.



**Thank you for  
your attention**

**Wouter Lueks**