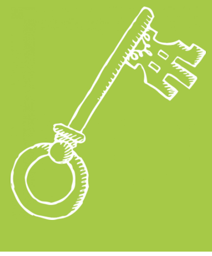
**IAIK** | Institute of Applied Information Processing and Communications

TU Graz



# Secure Classification as a Service

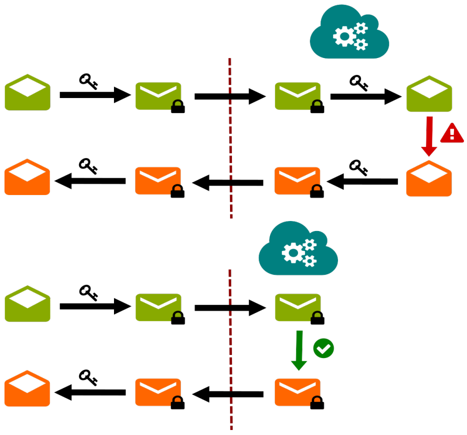
Advisor: **Daniel Kales, Roman Walch**

## Motivation

Fully homomorphic encryption (FHE) is often called the "holy grail" of cryptography, enabling outsourcing computations on encrypted data to a powerful cloud computer. One concrete application is the so-called "Classification as a Service", where a cloud provider has a trained machine learning model and wants to offer it as a service to clients. However, in many scenarios, the client's data is privacy sensitive (e.g., personal or medical data) and should be kept secret from the cloud service provider, preventing the client from using the service. Fully homomorphic encryption can fix the privacy issues in this case, enabling the client to send encrypted data to the cloud service, who evaluates his machine learning algorithm on the encrypted input and sends back an encrypted result to the client. The goal of this thesis is to use state-of-the-art FHE libraries to implement a small demonstrator of this approach.

## Goals and Tasks

- Understand necessary background (FHE, AI)
- Evaluate different FHE libraries and AI algorithms
- Implement a web-based client and server backend
- Package into repeatable demonstrator



Top: Classification without FHE, Bottom: Classification with FHE
Figures provided by Michael Steiner.

## Literature

- R. Gilad-Bachrach et al.
  CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy
  ICML
- I. Chillotti et al.
  TFHE: Fast Fully Homomorphic Encryption Over the Torus
  J. Cryptol. 2020

## Courses & Deliverables

☑ **Introduction to Scientific Working**
Short report on background
Short presentation

☑ **Bachelor Project**
Project code and documentation

☑ **Bachelor's Thesis**
Project code
Thesis
Final presentation

## Recommended if you're studying

☑ CS ☑ ICE ☑ SEM

## Prerequisites

- Interest in cryptography and mathematical basics
- python/C++/Rust programming, optional: Web Assembly

## Advisor / Contact

daniel.kales@iaik.tugraz.at