Secure Machine Learning in the Cloud

Roland Hostettler, Subhrakanti Dey, Anders Ahlén

Division of Signals and Systems
Department of Electrical Engineering
Uppsala University

VINNOVA
Sveriges innovationsmyndighet

UPPSALA
UNIVERSITET

Background

Machine learning

- Important in many industries (process, automotive, ...)
- Requires (sensitive) user/company data
- Can be computationally complex
- Can be implemented in the cloud
  - Scalable and flexible
  - Cost-efficient
  - Low-maintenance
  - Machine learning as a service (MLaaS)

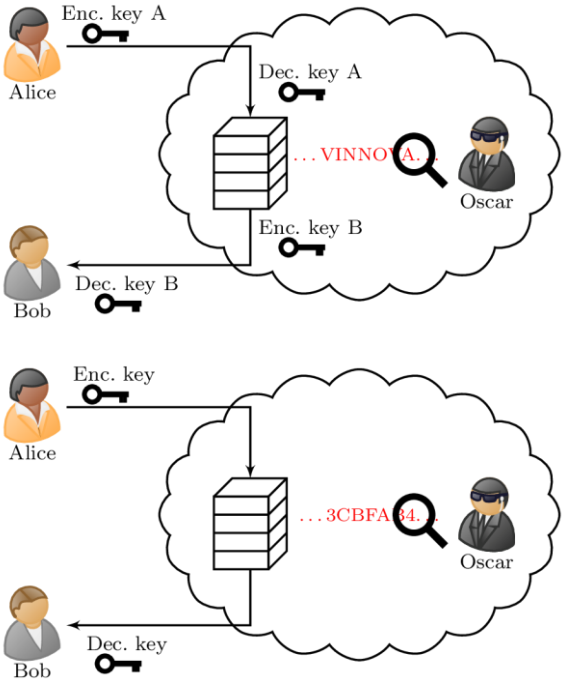
Data security threats

- Industrial/governmental espionage
- User privacy and integrity
- Local legislation
- Data breaches or leaks

UPPSALA
UNIVERSITET

Project goals

1. Data security: Homomorphic encryption
    - Allows processing encrypted data without decrypting it
2. Privacy/integrity: Differential privacy
    - Ensures that individual data points can not be distinguished
3. Platform requirements and demonstration
    - Computational requirements, network/system architecture



UPPSALA UNIVERSITET

Homomorphic encryption

Homomorphic encryption:

$$f(x \circ y) = f(x) \circ f(y) \;\Leftrightarrow\; \mathrm{Enc}(x \circ y) = \mathrm{Enc}(x) \circ \mathrm{Enc}(y)$$

Fully homomorphic encryption (FHE):

- Supports addition and multiplication
- Computationally complex
- Limited to, e.g., integers or in number of consecutive operations (levels)

UPPSALA
UNIVERSITET

Partially Homomorphic Encryption

- Supports limited types of operations (e.g., addition or multiplication)
- Computationally less complex
- Several schemes need to be combined or some operations need to be done in plain text

FHE: Cheon-Kim-Kim-Song scheme

- Fully homomorphic scheme
  - supports addition & multiplication
- Uses approximate arithmetic: $\mathrm{Enc}(x \circ y) \approx \mathrm{Enc}(x) \circ \mathrm{Enc}(y)$
- Leveled
  - Can be extended to support unlimited operations (bootstrapping)
- Complex functions can be implemented using polynomial approximations
  - Approximation must not exhaust the number of levels

Example: kNN using CKKS (1/2)

k-Nearest Neighbors (kNN):

- Simplistic approach for classification/prediction, based on finding k closest points to a test input
- Computational primitives:
  - Calculating the distance between data points (e.g., Euclidean distance)
  - Comparing distances and sorting
  - Weighted average / majority vote

Example: kNN using CKKS (2/2)

Some conclusions:

- Individual multiplications may be computationally complex (depending on the security parameters) but not a problem
- Approximation of functions using polynomials consumes many levels due to the multiplications
  - Good approximations often require high polynomial orders → consume many levels
  - Bootstrapping (refreshing the ciphertext) is expensive, and the main challenge is taking decisions and acting upon the mathematical output

Overview of results

- Understanding of HE and CKKS, their challenges and complexity
- Design studies for secure distributed computation network
- Implementation of PoC ongoing

UPPSALA
UNIVERSITET

Partners and funding

UPPSALA
UNIVERSITET

SWEDOME

intel

VINNOVA
Sveriges innovationsmyndighet

UPPSALA
UNIVERSITET

Collaboration Opportunities

- Research
- Industrial stakeholders
  - Use-cases
  - Reference group
- Verification and auditing

UPPSALA
UNIVERSITET