Towards Scalable, Efficient and Privacy Preserving Machine Learning
Rania Talbi, Sara Bouchenak

To cite this version:
Rania Talbi, Sara Bouchenak. Towards Scalable, Efficient and Privacy Preserving Machine Learning, Middleware '18 Doctoral Symposium, Dec 2018, Rennes, France. hal-01956155
Towards Scalable, Efficient and Privacy Preserving Machine Learning
Rania Talbi, Sara Bouchenak
INSA Lyon, France
 firstname.lastname@insa-lyon.fr

Context and Motivation

- Context:
  - Company A: Central Supervision Authority
  - Company B: Local bank
  - Data Mining for fraud detection

- Motivation:
  - Minimize the computational costs incurred by privacy preservation.
  - Provide an end-to-end privacy preserving outsourced data classification service.
  - Enable a set of mutually untrusted data owners to have a global vision on the union of their data without breaching the privacy of each one of them.
  - Enable dynamic data model updates when new training data samples are available.

Objective

- Preliminary results:
  - Data Mining for fraud detection in a B2B network.
  - This dataset contains 1000 bank transactions with 9 attributes each.
  - We compare our work to the Cipheredm framework [8].

Related work

- Different ML algorithms
  - Classification [1]
  - Association Rule Mining [2]

- Different Privacy-preservation objectives
  - ML output protection [3]
  - Original data protection [3]

- Privacy Preservation techniques
  - Cryptographic techniques (SMC/HE, GC, OT)

- Different architectures
  - Distributed [4]
  - Outsourced [5]

Design principles

- Cryptographic based protection (data model, training data, classification queries and responses)
- Partial homomorphic encryption (PHE ) based building blocks
- Combine PHE with cryptographic binding (DTPKC cryptosystem [6])
- We implemented the VDFT incremental decision tree learning algorithm [7]

- (1) Blind inputs
- (2) Partially or all decrypt blinded values
- (3) Decrypt blinded values
- (4) Run operation over blinded values

Naive approach: a combination of low level PP-building blocks
1st optimization: use inline building blocks
2nd optimization: Parallel computing

References