Practical Secure Aggregation for Privacy-Preserving Machine Learning

(Ch1∼3.1)

이종진

Seoul National University

ga0408@snu.ac.kr

July 15, 2020

Table of Contents

Ch1.Introduction

- Machine learning models trained on sensitive real-world data bring improvements to everything our lives.
- Wide-spread use of mobile devices
  - A large number of sensitive data can be used.
  - It entails risks $\rightarrow$ encryption is needed
  - We focus on the setting of mobile devices.
- This paper is about methods which are (machine learning + mobile devices + encryption).

Multiparty computation.

- Multiparty computation(MPC)
  - Encrypting the sensitive data by multiple people without sharing their inputs.
  - Multiparty means individual (mobile) devices.
- (Federated learning + Secure Aggregation)
  - Use securely combined the outputs of local machine learning on the users device in order to update global model.

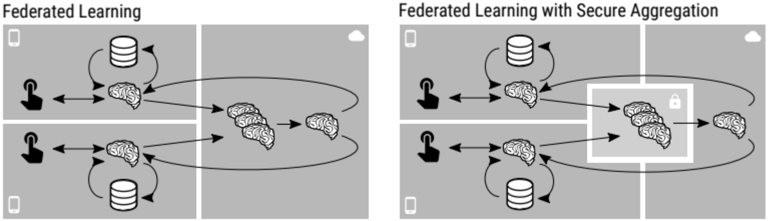
Federated Learning

- Consider training a deep neural network to predict the next word that a user will type.
- Users may be reluctant to upload their text messages to modeler's server.
- Federated Learning setting
  - Users maintains a private database and a shared global model.
  - Transfer highly processed, minimally scoped, ephemeral updates from users(ex, gradients).

Securely aggregation

- Although each update is ephemeral and contains no more direct information from the user's private dataset.
- It is possible to learn individual word that a user typed by inspecting that user's most recent update.
- Secure aggregation protocol is to compute weighted averages of updates from randomly selected user's.

Secure aggregation for federated learing



A proposed protocol

- Present a protocol for securely computing sums of vectors, which has a constant number of rounds, low communication overherad, robustness to failure
- Two varinants of the protocol:
  - One is more efficient and can be proven secure against honest but curious adversaries in the plain model.
  - The other guarantees privacy against active adversaries, but requires extra rounds, and is proven secure in the random oracle model.

Ch3. Crypotographic Primitives

- How to securely computing sums of vectors.
- Cryptographic Primitives
  1. Secret Sharing
  2. Key Agreement
  3. Authenticated Encryption
  4. Pseudorandom Generator
  5. Signature Scheme
  6. Public Key Infrastructure

- Shamir's t-out-of-n Secret Sharing, (t, n)-scheme.
  - A user split a secret s into n shares(n peoples).
  - any t shares can be used to reconstruct s.
  - but, any set of at most $t-1$ shares no information about s.
- Shamir uses the fact there exists unique polyinomial q(x) of degree $t-1$ that interpolates t distinct points $(x_i, y_i), i = 1, \ldots, t$
  - Given one point in a plane, there are a lot of lines passing through it.
  - If two points are given, there is a unique line passing through it.(coefficients are unique).

Shamir's t-out-of-n Secret Sharing

- Consider modular arithmetic instead of real arithmetic.
- A finite field $\mathbb{F}$(ex: $\mathbb{Z}_p$)
- (t, n)-scheme.
- Secret $s \in \mathbb{F}$, then construct $t-1$ polynomial

$$q(x) = s + a_1 x + \dots a_{t-1} x^{t-1}, \quad a_i \in \mathbb{F}$$

- Define $s_i = q(i) \pmod{p}$

Shamir's t-out-of-n Secret Sharing

- With at most t-1 number of s_i, we cannot reconstruct s(s could be $0, 1, \ldots, p-1$)
- With more then t number of s_i, we can reconstruct S
- Lagrange polynomial(Lagrange interpolation)

$$L_i(x) = \left(\frac{\prod_{j \in \mathcal{U}} (x - s_j)}{\prod_{i \neq j} (s_i - s_j)} \right)$$

Example

- Secret $s = 3$, parties $n = 4$, Field $\mathbb{F} = \mathbb{Z}_5$
- Construct (2,4) sheme.
- $\ell(X) = 2X + 3$
- $s_1 = \ell(X = 1) = 2 \times 1 + 3 = 0 \pmod{5}$
- $s_2 = \ell(X = 2) = 2 \times 2 + 3 = 2 \pmod{5}$
- $s_3 = \ell(X = 3) = 2 \times 3 + 3 = 4 \pmod{5}$
- $s_4 = \ell(X = 4) = 2 \times 4 + 3 = 1 \pmod{5}$
- $\ell^{'}(X) = \frac{s^{(2)} - s^{(1)}}{i_2 - i_1} \cdot X + \left(\frac{i_2 s^{(1)} - i_1 s^{(2)}}{i_2 - i_1} \right)$

Shamir's t-out-of-n Secret Sharing

- The scheme consists of two algorithm
- $\mathcal{U}$: n field elements represent users IDs $(1, \ldots, n)$
- The sharing algorithm:

  $SS.share(s, t, \mathcal{U}) \rightarrow \{(u, s_u)\}_{u \in \mathcal{U}}$.
- $\mathcal{V} \subset \mathcal{U}, |\mathcal{V}| \geq t$
- The reconstruction algorithm:

  $SS.recon(\{(u, s_u)\}_{u \in \mathcal{V}}, \mathcal{U}) \rightarrow s$.

Next

1. Key Agreement
2. Authenticated Encryption
3. Pseudorandom Generator
4. Signature Scheme
5. Public Key Infrastructure

The end.