Practical Secure Aggregation for Privacy-Preserving Machine Learning

Kallista A. Bonawitz, Vladimir Ivanov etc.

Google

CCS 2017

Background

- **背景**：联邦学习依然面临隐私问题（例如基于梯度的成员推理攻击）
- **贡献**：提出了SecAgg，利用秘密共享保护参与方的梯度
- **基础**：FedAvg

Algorithm 1 FederatedAveraging. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
  initialize w_0
  for each round $t = 1, 2, \dots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): // *Run on client* k
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

Preliminary

- **Secret Sharing**
  - 将一个秘密分为若干份，由多方共同存 储
  - t-out-of-n：只知道t份秘密即可恢复s
    - a_0是秘密，如有k份f(i)，a_0可解方程组求得

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{k-1} x^{k-1}$$

- **PKI / Signature / Authenticated**
  - 将上述用来混淆的 样本来自公开的大数据集
- **Pseudorandom Generator**
- **条件：**
  - honest-but-curious
  - trusted third party

Method

- 假设用户 u 的秘密是 x_u
- 每个用户对 u<v 协商一个随机数 s_u,v（DH协议等） 计算y_u:

$$y_u = x_u + \sum_{v \in \mathcal{U}: u<v} s_{u,v} - \sum_{v \in \mathcal{U}: u>v} s_{v,u} \pmod{R}$$

$$s_{u,v} \leftarrow \mathbf{KA.agree}(s_u^{SK}, s_v^{PK})$$

- 服务端计算即可安全求和
- s_u会以秘密分享发给

  其它用户，防止u掉线

$$\begin{aligned} z &= \sum_{u \in \mathcal{U}} y_u \\ &= \sum_{u \in \mathcal{U}} \left(x_u + \sum_{v \in \mathcal{U}: u<v} s_{u,v} - \sum_{v \in \mathcal{U}: u>v} s_{v,u} \right) \\ &= \sum_{u \in \mathcal{U}} x_u \pmod{R} \end{aligned}$$

Method

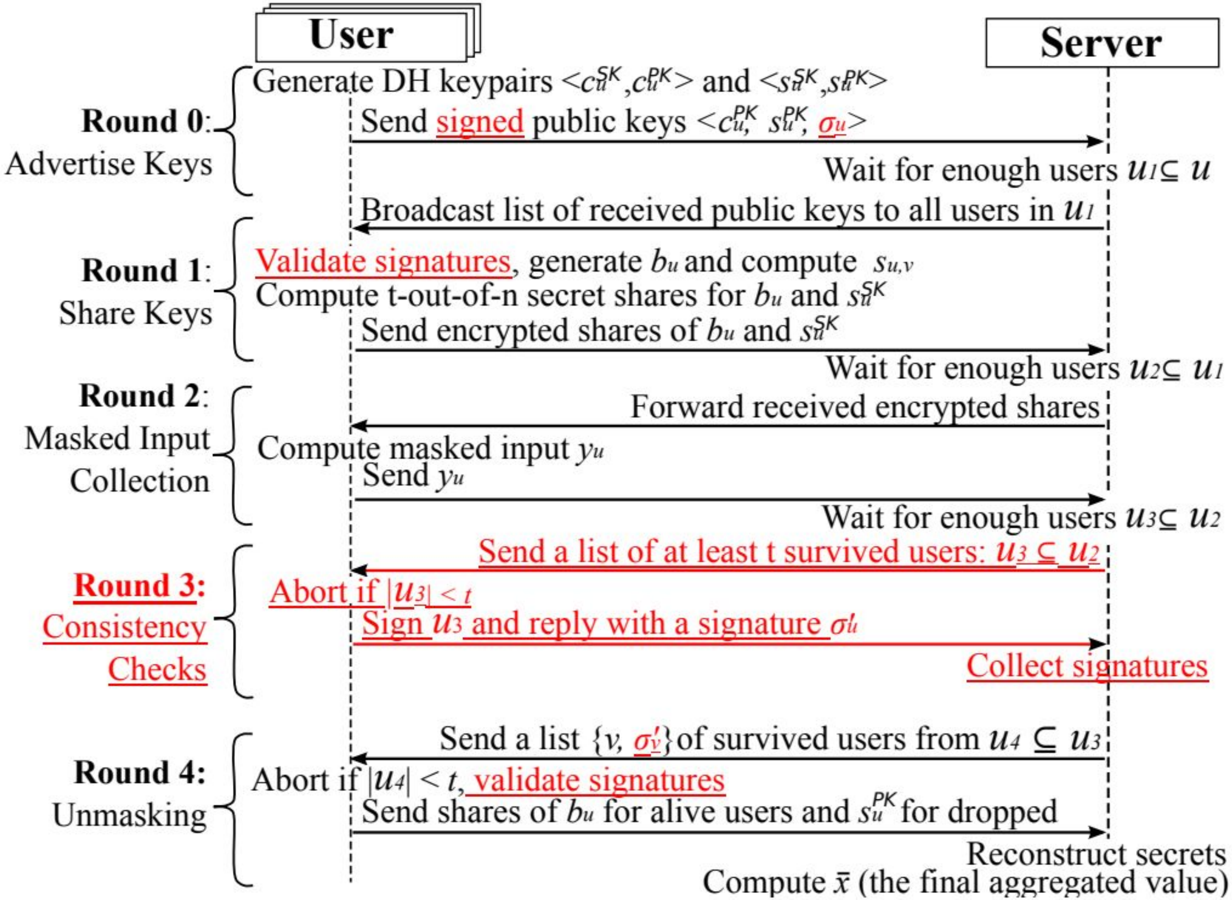
- 为防止某用户 u 由于网络延迟等，使得服务器在其发送数据前向其它用户请求恢复秘钥 s_u 成功，从而破获 x_u，引入随机种子 b_u，嵌套第二层秘密，新的计算方式如下：

一个诚实用户u不会同时提供一个用户v的s_v share和b_v share

$$
\begin{aligned}
\boldsymbol{y}_u = \boldsymbol{x}_u &+ \mathbf{PRG}(b_u) \\
&+ \sum_{v \in \mathcal{U} : u < v} \mathbf{PRG}(s_{u,v}) \\
&- \sum_{v \in \mathcal{U} : u > v} \mathbf{PRG}(s_{v,u}) \pmod{R}
\end{aligned}
$$

Protocol

- 基于秘钥交换



EXP

	User	Server[5]
computation	$O(n^2 + mn)$	$O(mn^2)$
communication	$O(n + m)$	$O(n^2 + mn)$
storage	$O(n + m)$	$O(n^2 + m)$

Figure 3: Cost summary for the protocol.

	Num. Clients	Dropouts	AdvertiseKeys	ShareKeys	MaskedInputColl.	Unmasking	Total
Client	500	0%	1 ms	154 ms	694 ms	1 ms	849 ms
Server	500	0%	1 ms	26 ms	723 ms	1268 ms	2018 ms
Server	500	10%	1 ms	29 ms	623 ms	61586 ms	62239 ms
Server	500	30%	1 ms	28 ms	514 ms	142847 ms	143389 ms
Client	1000	0%	1 ms	336 ms	1357 ms	5 ms	1699 ms
Server	1000	0%	6 ms	148 ms	1481 ms	3253 ms	4887 ms
Server	1000	10%	6 ms	143 ms	1406 ms	179320 ms	180875 ms
Server	1000	30%	8 ms	143 ms	1169 ms	412446 ms	413767 ms

Conclusion

- 属于比较直接的将一整套密码体系应用到FL的工作
- 创新点在于两层mask
- 其写法十分详细，公式、符号非常多且全(容易头秃)