计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(11):175−181 [doi: 10.15888/j.cnki.csa.009302]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

联邦学习下高效的隐私保护安全聚合方案①

王　珊, 荆　桃, 肖淦文, 张新林

(长安大学 信息工程学院, 西安 710018)
通信作者: 王　珊, E-mail: 2915154547@qq.com

摘　要: 联邦学习能使用户不共享原始数据的情况下, 允许多个用户协同训练模型. 为了确保用户本地数据集不被泄露, 现有的工作提出安全聚合协议. 但现有的多数方案存在未考虑全局模型隐私、系统计算资源与通信资源耗费较大等问题. 针对上述问题, 提出了联邦学习下高效的强安全的隐私保护安全聚合方案. 该方案利用对称同态加密技术实现了用户模型与全局模型的隐私保护, 利用秘密共享技术解决了用户掉线问题. 同时, 该方案利用 Pedersen 承诺来验证云服务器返回聚合结果的正确性, 利用 BLS 签名保护了用户与云服务器交互过程中的数据完整性. 此外, 安全性分析表明该方案是可证明安全的; 性能分析表明该方案是高效且实用的, 适用于大规模用户的联邦学习系统.

关键词: 联邦学习; 安全聚合; 隐私保护; 同态加密; 完整性

引用格式: 王珊,荆桃,肖淦文,张新林.联邦学习下高效的隐私保护安全聚合方案.计算机系统应用,2023,32(11):175–181. http://www.c-s-a.org.cn/1003-3254/9302.html

Efficient Privacy-preserving Secure Aggregation Scheme for Federated Learning

WANG Shan, JING Tao, XIAO Gan-Wen, ZHANG Xin-Lin

(School of Information Engineering, Chang'an University, Xi'an 710018, China)

Abstract: Federated learning allows multiple users to collaboratively train models without sharing the original data. To ensure that users' local datasets are not leaked, the existing works propose secure aggregation protocols. However, most of the existing schemes fail to consider global model privacy, and the system is at a high cost of computational and communicational resources. In response to the above problems, this study proposes an efficient and secure privacy-preserving secure aggregation scheme for federated learning. The scheme uses symmetric homomorphic encryption to protect the privacy of the user model and the global model and adopts secret sharing to solve users' dropout. At the same time, the Pedersen commitment is applied to verify the correctness of the aggregation results returned by the cloud server, and the BLS signature is utilized to protect the data integrity during the interaction between the users and the cloud server. In addition, security analysis illustrates that the proposed protocol is of provable security; performance analysis indicates that the protocol is efficient and practical for federated learning systems with large-scale users.

Key words: federated learning; secure aggregation; privacy-preserving; homomorphic encryption; integrity

2016 年, 谷歌提出了一种分布式的、多方协同计算的机器学习框架-联邦学习[1]. 利用联邦学习框架, 用户在本地设备上利用本地数据集训练模型参数, 该参数被称为梯度. 用户将自身训练好的模型参数上传至云服务器. 云服务器收到这些模型参数并对其进行聚合, 随后将聚合后的全局模型同步分发给用户进行新一轮的训练. 相比传统的分布式学习, 联邦学习能在保护数据隐私和安全的情况下实现用户之间的协同训练,

① 收稿时间: 2023-05-03; 修改时间: 2023-06-06; 采用时间: 2023-06-26; csa 在线出版时间: 2023-08-29
CNKI 网络首发时间: 2023-08-30

因此被广泛应用于物联网[2]、车联网[3]等各个领域. 然而, 尽管联邦学习为用户提供了更加安全和高效的训练方式, 但仍存在恶意敌手能根据用户上传的梯度推断出用户隐私数据的安全威胁[4].

为解决联邦学习中恶意敌手通过用户上传梯度获取用户隐私的问题, 联邦学习下的隐私保护安全聚合协议被提出[5–9]. 文献 [5,6] 基于安全多方计算提出使用双掩码安全聚合协议保护用户的本地模型隐私, 且能有效地支持用户掉线问题. 针对联邦学习中云服务器是半诚实的, 它可能为会节省计算资源甚至伪造聚合结果来影响最终模型训练. Xu 等[7] 提出可验证的安全聚合协议来验证云服务器返回聚合结果的正确性, 但验证过程中通信代价会随梯度维数线性增长, 导致较差的系统性能. Guo 等[8] 改进了 Xu 等[7] 的协议, 利用承诺方案与线性同态哈希函数验证聚合结果的正确性, 提高了系统的计算效率. 文献 [9] 利用 EI Gamal 同态加密技术结合 Diffie-Hellman 密钥交换协议[10] 和 Shamir 秘密共享算法[11], 提出可容忍用户掉线且能抵抗参与者合谋攻击的方案. 上述方案仅考虑了用户本地模型的隐私, 文献 [12] 指出已知全局模型的云服务器通过模型反演攻击能推断出用户的隐私信息, 这将使系统安全性降低. 文献 [13] 使用全同态加密技术保护了全局模型的隐私. 文献 [14] 利用 Paillier 同态加密技术结合双线性聚合签名验证了服务器返回聚合结果的正确性, 但协议[13,14] 未能支持用户随时退出系统.

现有的安全聚合方案存在以下 3 种问题: 多数方案未考虑全局模型的隐私, 已知全局模型明文的云服务器能通过全局模型推断用户的隐私; 使用代价较高的公钥加密技术来保护全局模型隐私且不能支持用户掉线, 这不利于资源受限的用户参与训练. 若用户因网络等原因退出训练, 将会影响整个训练过程; 此外, 现有方案未考虑交互过程中的数据完整性, 这使得其他用户与云服务器不能获得该用户上传的原始数据.

针对上述所提问题, 本文结合对称同态加密与双掩码安全聚合技术提出联邦学习下高效的强安全的隐私保护安全聚合方案. 具体而言, 本文贡献如下.

(1) 现有的多数方案[5–9] 未能考虑全局模型隐私, 所提方案利用轻量级的对称同态加密技术, 同时保护用户的本地模型与全局模型的隐私.

(2) 所提方案增加了保护用户与云服务器交互过程中的数据完整性这一安全需求. 同时所提方案不仅能有效支持用户验证聚合结果的正确性, 即使用户退出协议, 云服务器仍能有效地实现聚合.

(3) 安全性分析表明, 所提方案是安全的. 性能分析表明, 在满足相同功能性需求的前提下, 所提方案具有较低的计算代价与通信代价.与其他相关方案相比, 具有更优的性能.

1 背景知识

1.1 对称同态加密

Li 等首次提出了对称同态加密技术[15], 该技术能够支持同态加法和有限次的同态乘法操作, 并且其效率远高于非对称同态加密. 该技术包含以下算法.

(1) 密钥生成算法: 输入安全参数λ, 选择两个大素数p,q, $p \gg q$. 输出密钥$SK=(s,d,p,q)$. 计算$N=pq$为公开参数. 选择一个随机数$s \in Z_q^*$. d是为密文等级, 是一个小的正整数.

(2) 加密算法: 将密钥SK与消息$m \in Z_q^*$作为输入, 选择一个大的正整数 r 使得$|r|+|q|<|N|$, 对消息m加密得$c=Enc(SK,m,r)=s^d(rq+m) \bmod N$.

(3) 解密算法: 输入密钥SK与密文c, 执行解密计算得$m=Dec(SK,c)=(s^{-d}c \bmod N) \bmod q$.

1.2 BLS 签名

BLS 签名[16] 是一种基于双线性映射的数字签名算法, 具有签名短、公钥短、安全性高以及可实现匿名认证等优点, 且能将多个签名聚合为一个签名, 降低系统的通信开销与计算开销. 该技术包含以下算法.

(1) 初始化算法: 签名者选择双线性映射$e:\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, 哈希函数$h:\{0,1\}^* \to \mathbb{G}_2$.

(2) 密钥生成算法: 签名者随机选择私钥x和计算公钥$y=g^x \in \mathbb{G}_1$.

(3) 签名算法: 签名者利用x和消息$M_i \in \{0,1\}^*$, 计算$h=h(M)$与签名$\sigma=h^x$.

(4) 验证算法: 给定签名者的公钥y、消息 M 和签名σ. 验证者计算$h=h(M)$, 如果$e(g_1,\sigma)=e(y,h)$成立, 则接受; 否则拒绝.

1.3 Pedersen 承诺

Pedersen 承诺在文献 [17] 中提出, 其绑定性依赖于离散对数困难问题假设. Pedersen 承诺包含以下 3 个阶段.

(1) 初始化算法: 可信分发者生成一个阶数为q大素数的乘法群$\mathbb{G}$, 其生成元为g,h. 可信分发者公开参数

(q,g,h).

(2) 承诺算法: 承诺方选择一个随机数$r \in Z_q$, 承诺的消息为m. 计算承诺$C = Commit(m,r) = g^m h^r$, 将承诺值$C$发送给验证者.

(3) 打开承诺算法: 将(m,r)作为输入, 验证者检查承诺值C是否等于$g^m h^r$. 若等式成立, 则输出 1. 否则输出 0.

1.4 系统模型

本文系统模型包含 3 个实体: 可信中心 (TA), 云服务器 (CS), 用户集合 (Users). 系统模型图如图 1 所示.

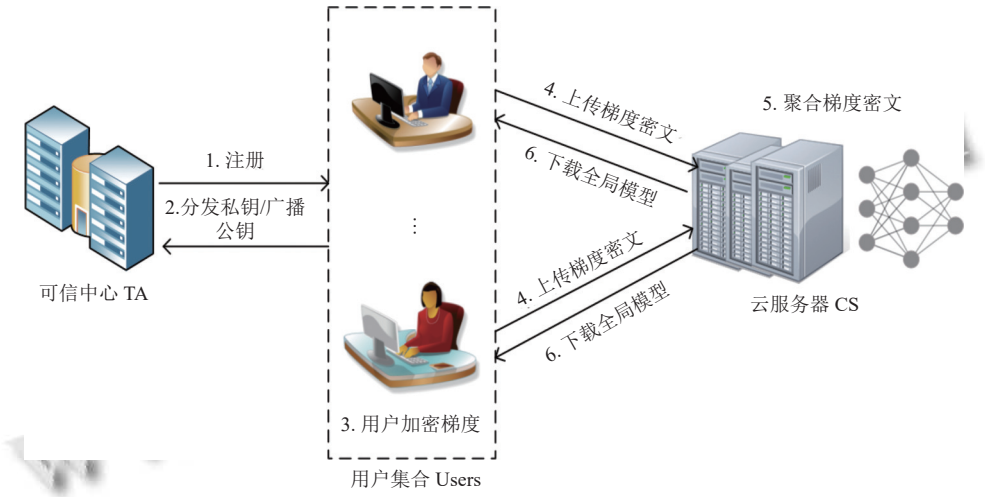


图 1 系统模型

(1) TA: 可信实体, 负责生成系统参数, 完成对用户$u \in U$的注册. 同时为用户生成公私钥对.

(2) CS: 不可信实体, 负责聚合从用户收到的梯度密文, 生成全局模型.

(3) Users: 半可信实体, 负责上传梯度参数给 CS. 同时验证从 CS 返回的聚合结果的正确性.

1.5 安全需求

(1) 机密性: 敌手即使截获了用户上传的梯度密文数据, 其也不能获取有效的明文数据.

(2) 认证性: 敌手可能会伪装成合法用户影响系统训练过程, 因此方案应对用户数据来源进行身份认证.

(3) 完整性: 敌手可能会篡改或伪造在公开信道传输的数据内容. 完整性确保服务器收到的信息就是用户发送的信息.

(4) 掉线鲁棒性: 由于用户设备是资源受限的, 在训练过程中可能会退出系统.

(5) 抵抗多个用户间合谋攻击: 半可信的用户可能尝试通过合谋来获取其他用户的隐私数据.

(6) 抵抗用户与云服务器合谋攻击: 用户与云服务器合谋可能会获取其他诚实用户的数据, 泄露该用户的隐私信息. 同时生成错误的聚合结果欺骗诚实用户.

1.6 设计目标

(1) 隐私保护: 仅有用户知道自身梯度, 其他任何实体不能获得用户梯度. 并且聚合结果由用户解密获得, 服务器不能得到全局模型.

(2) 可验证性: 用户应验证云服务器返回聚合结果的正确性再进行后续训练.

(3) 高效性: 由于用户的设备是受限的, 因此所提方案在满足上述安全需求的前提下, 系统的计算代价与通信代价尽可能降低.

2 方案描述

2.1 初始化阶段

(1) TA 选择两个阶数为q的乘法循环群$\mathbb{G}_1$与$\mathbb{G}_2$, 相应的生成元分别为g_1, g_2.

(2) TA 设置双线性映射$e: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, 同时选择一个安全的哈希函数$H: \{0,1\}^* \to \mathbb{G}_2$.

(3) TA 选择大素数p, 定义秘密共享有限域为Z_p, 其中$0 < t \leqslant n < p$. n为用户数量. t为秘密共享阈值. 输入安全参数λ, 生成对称同态加密密钥$K = \{s, d, \widehat{p}, \widehat{q}\}$, 计算$N = \widehat{p}\,\widehat{q}$.

(4) 公开系统参数$< p, q, g_1, g_2, h, \mathbb{G}_1, \mathbb{G}_2, n, t, N >$.

2.2 聚合阶段

(1) 注册算法: 所有用户需要向 TA 注册并获得相应的密钥. 用户i输入其身份信息, TA 输出安全参数与对应用户的私钥.

1) 用户i随机选择$x_i \in Z_q$, 计算公钥$y_i = g_1^{x_i}$.

2) 用户i输入其身份信息ID_i, 签名公钥y_i与当前时间戳T_i, 计算签名$\sigma_i = H(ID_i \| y_i \| T_i)^{x_i}$, 并发送$(\sigma_i, ID_i, y_i, T_i)$给 CS.

3) 收到消息$(\sigma_i, ID_i, y_i, T_i)$后, TA 计算$e(g_1, \sigma_i) = e(y_i, H(ID_i \| y_i \| T_i))$. 若上述等式验证通过, 则允许用户参与后续训练. 定义当前用户集合为U.

(4) TA 为每个用户生成两对公私钥(N_i^{PK}, N_i^{SK})与(P_i^{PK}, P_i^{SK}).

(5) TA 输出安全参数$(ID_i, N_i^{PK}, P_i^{PK}, y_i)_{i \in U}$, 并将密钥$(K, N_i^{SK}, P_i^{SK})$通过安全信道发送给相应用户.

(2) 随机数共享算法: 用户i $(i \in U)$. 随机选择随机数, 并与其他用户进行秘密分享. 随后, 用户输入随机数的份额, 计算并输出相应份额的密文.

1) 用户i随机选择$b_i \in Z_p$, 构造$t-1$次多项式$f_i(x) = b_i + \sum_{m=1}^{t-1} c_{im} x^m \bmod p$, 其中$c_{im} \in Z_p$, $f_i(j) = b_{i,j}$是用户i给j关于b_i的共享.

2) 用户i构造$t-1$次多项式$g_i(x) = N_i^{SK} + \sum_{m=1}^{t-1} d_{im} x^m \bmod p$, 其中$d_{im} \in Z_p$. $g_i(j) = N_{i,j}^{SK}$是用户i给j关于N_i^{SK}的共享.

3) 对于$j \in U \backslash \{i\}$, 用户i利用 Diffie-Hellman 密钥协商协议计算成对对称密钥$key_{i,j} \leftarrow KA.agree(P_i^{SK}, P_j^{PK})$.

4) 用户i输入份额$b_{i,j}, N_{i,j}^{SK}$, 利用对称加密计算$P_{i,j} \leftarrow AE.Enc(key_{i,j}, i \| j \| b_{i,j} \| N_{i,j}^{SK})$, 输出密文$P_{i,j}$.

(3) 加密及掩码算法: 用户i $(i \in U)$输入自身梯度v_i并对其进行加密, 输出相应的梯度密文并将该密文上传至 CS.

1) 用户i计算梯度的承诺$E_i = g^{v_i} h^{b_i}$.

2) 对于$j \in U \backslash \{i\}$, 用户u_i计算另一个协商密钥作掩码$e_{i,j} \leftarrow KA.agree(N_i^{SK}, N_j^{PK})$.

3) 用户i输入自身梯度值v_i, 随机选择r_i满足$|r_i| + |q| < N$, 结合伪随机生成器并计算$C_i = s^d (r_i q + v_i) + PRG(b_i) + \sum_{j \in U_1} \Delta_{i,j} PRG(e_{i,j}) \bmod N$. 若$i < j$, $\Delta_{i,j} = 1$; $i > j$, $\Delta_{i,j} = -1$; $i = j$, $\Delta_{i,j} = 0$.

4) 用户i计算签名$\sigma_i^1 = H(ID_i \| C_i \| E_i \| P_{i,j})^{x_i}$, 输出$(\sigma_i^1, ID_i, C_i, E_i, P_{i,j})$给 CS.

5) 收到来自至少t个用户的消息$(\sigma_i^1, ID_i, C_i, E_i)(i \in U_1)$, CS 首先计算$\sigma_1 = \prod_{i \in U_1} \sigma_i^1$并验证签名$e(g_1, \sigma_1) = \prod_{i \in U_1} e(y_i, H(ID_i, C_i, E_i, P_{i,j}))$. 若等式成立, 广播$(E_i, P_{i,j})$ $(i \in U_1)$给U_1中的每个用户.

(4) 掩码去除及聚合算法: CS 将收到的密文结果作为输入并对其进行聚合, 输出去除掩码后的聚合结果值.

1) 用户i收到$(E, P_{i,j})(i \in U_1)$, 检验若$|U_1| < t$, 则中止. 否则, 对于用户$j \in U \backslash \{i\}$, 执行解密算法得$j' \| i' \| N_{j',i'}^{SK}, \| b_{j',i'} \leftarrow AE.Dec(key_{i,j}, P_{i,j})$, $j = j', i = i'$.

2) 用户i根据解密得到的份额$N_{j,i}^{SK}$与$b_{j,i}$计算签名$\sigma_i^2 = H(ID_i \| b_{j,i(j \in U_1)} \| N_{j,i(j \in U \backslash U_1)}^{SK})^{x_i}$, 并发送$(\sigma_i^2, ID_i b_{j,i(j \in U_1)}, N_{j,i(j \in U \backslash U_1)}^{SK})(i \in U_1)$给 CS.

3) 收到来自至少t个用户的消息, CS 令U_2 $(U_2 \subseteq U_1)$表示用户集合, 检验若$|U_2| < t$, 则中止. 否则, CS 计算$\sigma_2 = \prod_{i \in U_2} \sigma_i^2$. 验证签名的有效性$e(g_1, \sigma_2) = \prod_{i \in U_2} e(y_i, H(ID_i \| b_{i,j_{(j \in U_1)}} \| N_{i,j(j \in U \backslash U_1)}^{SK}))$. 若等式成立, 对于$i$ $(i \in U / U_1)$, CS 通过拉格朗日插值多项式方法[11] 重构N_i^{SK}. 对于u_i $(i \in U_1)$, CS 通过拉格朗日插值多项式方法重构b_i.

4) CS 计算$\sum_{i \in U_1} b_i$, 利用已知的N_i^{SK}与N_j^{PK}得出$PRG(e_{i,j})$, 计算得$\sum_{j \in U_1, i \in U \backslash U_1} PRG(e_{i,j})$.

5) 输入梯度密文$C_i (i \in U_1)$, CS 计算聚合密文$c = \sum_{i \in U_1} C_i - \sum_{i \in U_1} b_i + \sum_{j \in U_1, i \in U_2 \backslash U_1} PRG(e_{i,j}) \bmod N$. 输出$\{c, \sum_{i \in U_1} b_i, U_2\}$给用户.

(5) 解密算法: 输入聚合密文值c, 用户i $(i \in U_2)$解密并输出聚合梯度的明文.

1) 用户i收到集合U_2. 检验若$|U_2| < t$, 则中止.

2) 用户i利用收到的聚合密文值c作为输入, 计算:

$$\sum_{i \in U_1} v_i = ((s^{-d} c) \bmod N) \bmod \widehat{q} \tag{1}$$

输出聚合梯度明文$\sum_{i \in U_1} v_i$.

2.3 验证阶段

用户i $(i \in U_2)$计算:

$$g^{\sum_{i \in U_1} v_i} h^{\sum_{i \in U_1} b_i} = E = \prod_{i \in U_1} E_i \tag{2}$$

若上述等式成立, 则表示 CS 返回的聚合结果是正确的.

所提方案中等式 (1) 与等式 (2) 的正确性验证如下.

用户根据等式 (1) 来解密来自 CS 返回的密文结

果. 此外, 用户根据等式 (2) 是否成立验证 CS 返回的聚合结果是否准确.

等式 (1) 的正确性:

$$\begin{aligned}&((s^{-d}c) \bmod N) \bmod \widehat{q}\\&=\left(\left(s^{-d}\sum\nolimits_{i\in U_2}s^d(r_i\widehat{q}+v_i)\right) \bmod N\right) \bmod \widehat{q}\\&=\left(\sum\nolimits_{i\in U_2}(r_i\widehat{q}+v_i)\right) \bmod \widehat{q}\\&=\sum\nolimits_{i\in U_2}v_i\end{aligned}$$

等式 (2) 的正确性:

$$g^{\sum_{i\in U_1}v_i}h^{\sum_{i\in U_1}b_i}=\prod\nolimits_{i\in U_1}g^{v_i}h^{b_i}=\prod\nolimits_{i\in U_1}E_i$$

3 安全分析与性能评价

3.1 安全需求分析

(1) 机密性: 由于用户使用对称同态加密技术对梯度进行加密, CS 仅对密文进行操作不能得到全局模型参数. 因此, 所提方案提供了机密性.

(2) 认证性: 用户i提前利用自己的身份向 TA 完成注册. 当向其他用户与 CS 发送数据后, 用户j和 CS 在对数据签名验证的同时也对用户的身份实现了认证.

(3) 完整性: 利用 BLS 签名技术, 用户i对密文C_i进行签名并将(σ_i^1, C_i)发送给 CS. 由于敌手无法得知x_i, 则其不能产生的合法的σ_i^1. 若敌手伪造签名或修改数据内容, 在对签名进行验证时会被检测出来. 因此, 该方案实现了交互过程中的数据完整性保护.

(4) 抵抗多个用户合谋攻击: 利用 Shamir 秘密共享技术, 当小于t个用户合谋时, 不能得到用户掩码梯度密文时所使用的掩码值. 因此, 所提方案能抵抗多个用户合谋攻击.

(5) 抵抗用户与云服务器间合谋攻击: 恶意用户与 CS 合谋获取梯度密文, 但由于掩码的存在, 恶意用户无法得到真实梯度的值. 因此, 所提方案能抵抗用户与云服务器间合谋攻击.

3.2 功能比较

本节给出了所提方案与相关方案[9,14]的功能性比较, 如表 1 所示. 其中 F1 表示全局模型隐私; F2 表示抵抗合谋攻击; F3 表示可验证性; F4 表示数据完整性; F5 表示掉线鲁棒性. 文献 [9] 能抵抗合谋攻击且支持用户掉线. 但未能保护全局模型隐私且未能验证聚合结果的正确性. 文献 [14] 保护了全局模型隐私且支持对聚合结果的正确性验证, 但该方案不能支持用户掉线. 同时上述方案未能保护交互过程中的数据完整性. 本文所提方案能满足上述所有功能.

表 1 功能对比

功能	文献[9]	文献[14]	所提方案
F1	×	√	√
F2	√	×	√
F3	×	√	√
F4	×	×	√
F5	√	×	√

注: “√”表示满足, “×”表示不满足

3.3 计算代价

本节基于 Charm 密码库[18] 和 Java 语言模拟测试了相关密码学操作及执行时间如表 2 所示. 实验环境 i7-7700HQ (2.80 GHz) 的 CPU, 内存为 4 GB 的 64 位 Ubuntu 操作系统. 基于 128 bits 的安全性, 所提方案与文献 [14] 都使用了双线性映射, 选择双线性映射$e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$, $\mathbb{G}_1$是q阶循环群, q是 512 bits 的素数. 令n表示参与训练过程中的用户数量, n_d表示训练中掉线用户的数量, $t=0.6n$表示秘密共享协议的阈值. 表 3 为当掉线用户数量$n_d=0.3n$时, 所提方案与文献 [9,14] 在用户端与云服务器端的计算代价对比.

表 2 密码学操作执行时间 (ms)

符号	描述	执行时间
$T_{Pair_{\mathbb{G}}}$	双线性对操作	19.8595
$T_{Mul_{\mathbb{G}_1}}$	$\mathbb{G}_1$下的乘法操作	0.0016
$T_{Exp_{\mathbb{G}_1}}$	$\mathbb{G}_1$下的指数操作	0.7352
$T_{Mul_{\mathbb{G}_2}}$	$\mathbb{G}_2$下的乘法操作	0.0166
$T_{Exp_{\mathbb{G}_2}}$	$\mathbb{G}_2$下的指数操作	1.3258
$T_{Hash_{\mathbb{G}_2}}$	映射到$\mathbb{G}_2$的哈希操作	0.0906
$T_{Mul_{\mathbb{G}_T}}$	$\mathbb{G}_T$下的乘法操作	0.0274
$T_{Add_{Z_p}}$	Z_p下的加法操作	0.0003
$T_{Mul_{Z_p}}$	Z_p下的乘法操作	0.0007
$T_{Exp_{Z_p}}$	Z_p下的指数操作	0.0196
$T_{Mul_{Z_n^*}}$	Z_n^*下的乘法操作	0.0127
$T_{Exp_{Z_n^*}}$	Z_n^*下的指数操作	9.6022
$T_{Exp^n_{Z_n^*}}$	Z_n^*下的n次指数操作	53.2424
$T_{Mul_{Z^*_{n^2}}}$	$Z^*_{n^2}$下的乘法操作	0.0188
$T_{Exp_{Z^*_{n^2}}}$	$Z^*_{n^2}$下的指数操作	35.8323
$T_{Mod_{\widehat{q}}}$	模$\widehat{q}$运算	0.0009

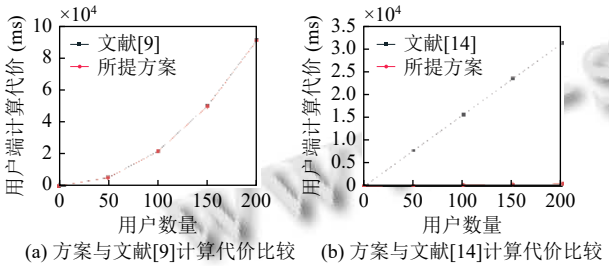
图 2 和图 3 分别给出了随用户数量的变化, 在用户端与云服务器端所提方案与文献 [9,14] 的计算代价比较. 由图 2(a) 可知, 用户数量为 50、100、150、200

时, 所提方案的计算代价相比于文献 [9] 分别降低了 110.28 ms、220.56 ms、330.84 ms、441.12 ms. 由此推断随着用户数量的增加, 所提方案更优于文献 [9]. 由图 2(b) 可知, 所提方案的运行时间与文献 [14] 相比较低. 经计算, 所提方案相比于文献 [14] 降低了 98.10%. 由图 3(a) 可知, 所提方案与文献 [9] 的运行时间几乎相同. 由图 3(b) 可知, 所提方案与文献 [14] 相比, 具有明显的优势.

表 3 计算代价比较

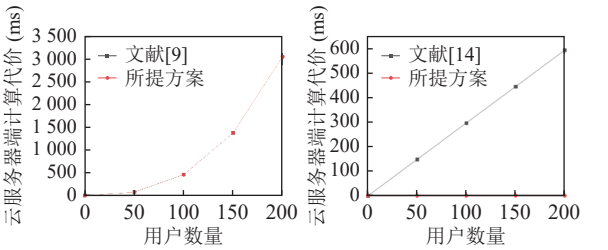
方案	用户端的计算代价	云服务器端计算代价
文献[9]	$0.0012n^3+2.0800n^2+1.4211n$	$0.0003n^3+0.0043n^2+0.0035n+0.7325$
所提方案[1]	$0.0012n^3+2.0800n^2-0.7845n$	$0.0003n^3+0.0043n^2-0.0003n$
文献[14]	$155.9072n$	$0.0354n+39.6836$
所提方案[2]	$2.9625n$	$0.0019n-0.0003$

注: 所提方案[1]表示所提方案与文献[9]实现的功能相同. 所提方案[2]表示所提方案与文献[14]实现的功能相同



(a) 方案与文献[9]计算代价比较 (b) 方案与文献[14]计算代价比较

图 2 用户端计算代价比较



(a) 方案与文献[9]计算代价比较 (b) 方案与文献[14]计算代价比较

图 3 云服务器端计算代价比较

3.4 通信代价

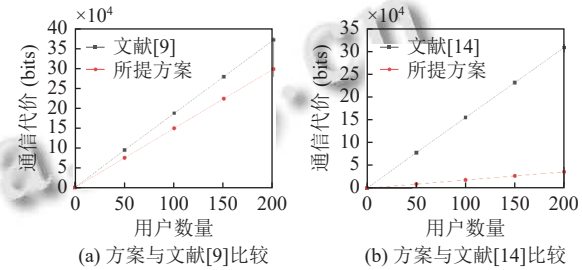
本节给出所提方案的通信代价及在实现相同功能的前提下, 与文献 [9,14] 的通信代价对比. 基于 128 bits 的安全性, 定义群$\mathbb{G}_1$中的元素大小为 512 bits; $\mathbb{G}_2$群中的元素大小为 1 024 bits; 双线性映射群$\mathbb{G}_T$中的元素是 3 072 bits; N的大小为 1 024 bits; Z_n^*的大小为 6 144 bits; $Z_{n^2}^*$的大小为 12 288 bits, Z_p的大小为 256 bits, 用户身份长度 ID 为 32 bits. 表 4 为所提方案与文献 [9,14] 的通信代价对比. 其中, 设定相关方案的用户数量与掉线用户数量分别为 n、$0.6n$.

图 4 给出了随用户数量的变化, 所提方案与文献 [9,14] 通信代价比较. 所提方案中单个用户与云服务器端的通信代价均低于文献 [9,14]. 当用户数量$n=200$ 时, 相比于文献 [9,14] 分别降低了 19.75%、88.34%. 对于资源受限的用户端来说是十分有利的, 同时服务器能有效地节省自身资源.

表 4 通信代价比较

方案	用户端计算代价	云服务器端计算代价	总体通信代价
文献[9]	$768n+2048$	$1075.2n$	$1843.2n+2048$
所提方案[1]	$768n+512$	$716.8n$	$1484.8n+512$
文献[14]	13 312	$15364n$	$15364n+13312$
所提方案[2]	1 536	$1792n$	$1792n+1536$

注: 所提方案[1]表示所提方案与文献[9]实现的功能相同. 所提方案[2]表示所提方案与文献[14]实现的功能相同



(a) 方案与文献[9]比较 (b) 方案与文献[14]比较

图 4 总体通信代价比较

综上所述, 所提方案需较少的通信代价, 更加适用于资源受限的联邦学习系统.

4 结论

本文利用对称同态加密技术结合双掩码协议提出了联邦学习下保护全局模型与支持用户掉线的可验证的安全聚合协议. 该方案利用对称同态加密技术解决现有文献利用公钥体系下同态加密代价较高的问题. 同时利用 Pedersen 承诺避免了利用双线性聚合签名技术不能抵抗用户合谋攻击的缺陷且能高效地实现对聚

合结果的验证. 安全性证明表明, 所提方案满足机密性、认证性、完整性且能抵抗用户间与用户与云服务器间的合谋攻击, 提供了较为全面的功能性保证. 性能分析表明, 与其他文献相比所提方案系统的效率更高, 能更好地满足实际需求.

参考文献

1 McMahan HB, Moore E, Ramage D, *et al*. Federated learning of deep networks using model averaging. arXiv:1602.05629v1, 2016.

2 黄倩怡, 李志洋, 谢文涛, 等. 智能家居中的边缘计算. 计算机研究与发展, 2020, 57(9): 1800–1809. [doi: 10.7544/issn1000-1239.2020.20200253]

3 莫梓嘉, 高志鹏, 杨杨, 等. 面向车联网数据隐私保护的高效分布式模型共享策略. 通信学报, 2022, 43(4): 83–94. [doi: 10.11959/j.issn.1000-436x.2022074]

4 Zhu LG, Liu ZJ, Han S. Deep leakage from gradients. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 1323.

5 Bonawitz KA, Ivanov V, Kreuter B, *et al*. Practical secure aggregation for privacy-preserving machine learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas: ACM, 2017. 1175–1191.

6 Liu ZY, Guo JL, Lam KY, *et al*. Efficient dropout-resilient aggregation for privacy-preserving machine learning. IEEE Transactions on Information Forensics and Security, 2023. 1839–1854. [doi: 10.1109/TIFS.2022.3163592]

7 Xu GW, Li HW, Liu S, *et al*. Verifynet: Secure and verifiable federated learning. IEEE Transactions on Information Forensics and Security, 2020, 15: 911–926. [doi: 10.1109/TIFS.2019.2929409]

8 Guo XJ, Liu ZL, Li J, *et al*. VeriFl: Communication-efficient and fast verifiable aggregation for federated learning. IEEE Transactions on Information Forensics and Security, 2021, 16: 1736–1751. [doi: 10.1109/TIFS.2020.3043139]

9 Zhang L, Xu JB, Vijayakumar P, *et al*. Homomorphic encryption-based privacy-preserving federated learning in IoT-enabled healthcare system. IEEE Transactions on Network Science and Engineering, 2022: 1–17. [doi: 10.1109/TNSE.2022.3185327]

10 Diffie W, Hellman M. New directions in cryptography. IEEE Transactions on Information Theory, 1976, 22(6): 644–654. [doi: 10.1109/TIT.1976.1055638]

11 Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613. [doi: 10.1145/359168.359176]

12 Tramèr F, Zhang F, Juels A, *et al*. Stealing machine learning models via prediction APIs. Proceedings of the 25th USENIX Conference on Security Symposium. Austin: USENIX Association, 2016. 601–618.

13 Phong LT, Aono Y, Hayashi T, *et al*. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Transactions on Information Forensics and Security, 2018, 13(5): 1333–1345. [doi: 10.1109/TIFS.2017.2787987]

14 Zhang XL, Fu AM, Wang HQ, *et al*. A privacy-preserving and verifiable federated learning scheme. Proceedings of the 2020 IEEE International Conference on Communications. Dublin: IEEE, 2020. 1–6. [doi: 10.1109/ICC40277.2020.9148628]

15 Li LC, Lu RX, Choo KKR, *et al*. Privacy-preserving-outsourced association rule mining on vertically partitioned databases. IEEE Transactions on Information Forensics and Security, 2016, 11(8): 1847–1861. [doi: 10.1109/TIFS.2016.2561241]

16 Boneh D, Gentry C, Lynn B, *et al*. Aggregate and verifiably encrypted signatures from bilinear maps. Proceedings of the 2003 International Conference on the Theory and Application of Cryptographic Techniques. Warsaw: Springer, 2003. 416–432.

17 Pedersen TP. Non-interactive and information-theoretic secure verifiable secret sharing. Proceedings of the 2001 Annual International Cryptology Conference. Berlin: Springer, 2001. 129–140.

18 Akinyele JA, Garman C, Miers I, *et al*. Charm: A framework for rapidly prototyping cryptosystems. Journal of Cryptographic Engineering, 2013, 3(2): 111–128. [doi: 10.1007/s13389-013-0057-3]

(校对责编: 孙君艳)