高技术通讯 2022 年 第 32 卷 第 10 期:991-1003

doi:10.3772/j.issn.1002-0470.2022.10.001

基于双重混沌映射算法的深度学习模型梯度安全保护研究①

林 宁② 陈晓明 夏春伟 李文星 叶 靖③ 刘自臻 李晓维

(中国科学院计算技术研究所 北京 100190)

(中国科学院大学计算机科学与技术学院 北京 101408)

摘 要 在联邦学习任务中,不同用户会上传深度学习模型的梯度到中央服务器进行梯度聚合,然而直接上传模型的原始梯度并不安全,攻击者会利用梯度攻击方法还原出用户的输入数据。当前,基于安全多方计算(SMPC)、差分隐私(DP)和同态加密(HE)来保护梯度安全的方法,存在通信开销较大、精度损失严重和加解密时延开销过大等主要问题。本文提出一种基于双重混沌映射算法的深度学习模型梯度安全保护方法,通过交换深度学习模型梯度的位置能够有效地防止恶意攻击者通过梯度攻击来偷窥用户个人隐私。为了降低时延开销,本文将深度学习模型层的映射问题转化为 0-1 整数背包问题,并利用动态规划求解出最优的保护方案。在 CIFAR-10、CIFAR-100、LFW 以及 ImageNet 数据集上的实验结果表明,本文所提方法能够防御当前最有效的两种梯度攻击,保护了深度学习模型梯度的安全性。此外,在 CPU、GPU 以及 3 款手机芯片上的实验结果表明,所提方法运行效率极高仅需要毫秒级就能完成安全保护。

关键词 深度学习;梯度安全;混沌映射;整数背包;动态规划

0 引 言

随着以深度学习模型为代表的人工智能技术的快速研究和发展,模型和数据的安全和隐私保护问题也开始受到广泛的关注。例如,在分布式训练深度学习模型的任务中(如联邦学习(federated learning, FL)[1-6]),为了保护用户的输入数据隐私,不同用户通过上传模型的梯度而不是输入数据到中央服务器进行梯度聚合。然而,最近的研究结果表明,用户直接上传模型的原始梯度并不安全。例如,文献[7]提出一种有效的梯度泄露攻击方法(deep leakage from gradients, DLG),恶意攻击者直接利用模型的梯度,通过 L-BFGS[8]优化求解器来不断优化梯度间的欧式距离损失函数,就能完全恢复出用户的输入图像。文献[9]也提出一种有效的梯度攻击方法(inverting gradients attack, IGA),通过构造初始化梯度与真实梯度之间的 Cosine 相似度距离,并利用 ADAM[10]优化算法能够在层数更深的模型上(如 ResNet-18 模型[11])恢复出用户的隐私输入图像。

现存的基于安全多方计算(secure multi-party computation,SMPC)、同态加密(homomorphic encryption, HE)以及差分隐私(differential privacy, DP)保护数据安全的方法,存在的主要问题分别是通信开销较大、时延开销过大以及精度损失严重。应用于图像领域中的混沌映射算法具有延迟开销较小、保护效果良好且映射过程无损等特点,这些都适合成为解决深度学习模型梯度安全保护的备选技术方案。

本文的主要贡献包括:(1)总结了梯度攻击方法的基本原理,提出一种基于混沌映射算法的梯度

① 国家重点研发计划(2020YFB1600201)和国家自然科学基金(U20A20202,62090024,61876173)资助项目。

② 男,1993 年生,博士生;研究方向:计算机系统结构,深度学习模型安全;E-mail: linning19b@ict.ac.cn。

③ 通信作者,E-mail: yejing@ict.ac.cn。

(收稿日期:2021-07-06)

安全保护方法。保护梯度不是通过改变梯度的值大小来实现的,而是利用混沌映射算法变换梯度的位置来实现的,避免了复杂的数学运算过程,极大地降低了时延开销。映射之后的梯度能够完全恢复原始梯度值的大小,因此不会对模型的精度造成影响。(2)提出了一种有效且时延开销较小的深度学习模型梯度映射保护方案。通过在模型各层梯度内任意选取不同的映射参数,并将模型层的映射问题转化为 0 - 1 整数背包问题,利用动态规划算法求解出最优的保护方案,进一步降低了映射整个深度学习模型的时延开销。(3)提出了一种双重混沌映射方案,能在联邦学习场景中同时防止恶意服务器和恶意用户利用梯度攻击还原出用户的输入隐私数据。本文在最新的两类梯度攻击方法 DLG 以及 IGA 上验证了所提方法的有效性,同时在多种深度学习模型上进行了实验并给出梯度保护算法在不同硬件平台上的时延开销对比。

1 研究现状

当前主要存在三类保护深度学习模型梯度安全的方法。第一类是基于同态加密的数据加密方法。同态加密方法存在的主要问题是加解密时延开销过大。例如,文献[12]使用 Paillier 算法[13-14](同态加密算法的一种),在一块 Intel Xeon CPU E5-2660 v3 CPU 上需要花费 454.8 ms 才能加密 52 650 个模型的参数值(大约 0.5 M)。当前这种加密效率无法满足实用性的需求,因为绝大多数深度学习模型的参数量远大于 0.5 M,例如 VGG16 模型[15]的参数量为 138 M,在时延开销方面至少需要花费大约二十多分钟来完成一次 VGG16 模型的加密过程。然而,在联邦学习场景中,通常需要几百次梯度上传操作,意味着需要耗费大约十几个小时在模型的梯度加密操作上。而对于计算资源受限的端设备,加密时延开销将会更大。为了降低加解密时延开销,文献[16]提出先利用矩阵分解方式来得到少量需要上传的参数值(如梯度值),之后再对分解后的参数值进行同态加密操作。然而,尽管加密少量的值能够降低时延开销,但是由于依然采用了运算过程复杂的同态加密方法,因此加解密时延依然较大。此外,由于该方法利用了矩阵分解,会对模型的精度造成影响,且方法未在大数据集上(如 ImageNet 数据集)或者层数较深的模型上进行实验验证。因此,当前基于同态加密来保护梯度安全性的方法,在算法运行效率方面依然有待进一步提升。

第二类保护梯度安全的方法是差分隐私[17-18]。差分隐私方法的基本思想是在权衡实用性和隐私性的前提下,向数据引入噪声,因此差分隐私的时延几乎可以忽略不计。然而,由于对需要保护的模型梯度添加了噪声,无法保证模型的精度不受影响。添加的噪声级越大,对模型梯度保护的安全等级就越高,然而模型精度损失就越大。例如在文献[7]中,作者在实验中说明为了防御 DLG 攻击,采用差分隐私方法对模型梯度值添加高斯噪声或拉普拉斯噪声。当添加的噪声等级为 0.001 时,模型的精度下降了大约 3%,但无法起到梯度保护作用,攻击者利用添加噪声后的梯度依然能够恢复出模型的输入图像;与之相反,当添加的噪声等级为 0.01 或者 0.1 时,能对梯度起到安全保护作用,然而模型精度下降超过了 30%。因此,基于添加噪声的差分隐私方法,在安全性与精度权衡方面需要进一步研究。

此外,还有一类保护梯度安全的方法是安全多方计算,最早由文献[19]提出,主要研究如何协同地从每一方的隐私输入中计算函数的结果,而不需要将输入展示出来。安全多方计算主要通过不经意传输(oblivious transfer, OT)、密钥共享(secret sharing, SS)和阈值同态加密(threshold homomorphic encryption, THE)来实现[4]。安全多方计算存在的主要问题是通信过程复杂以及通信轮数较多,造成的计算开销和时延开销较大。例如,文献[20]提出利用密钥共享来对不同用户上传的梯度进行安全聚合,在服务器端的通信开销为 $O(n^2 + mn)$,其中 m 代表数据量的大小,n 代表用户的个数。因此,数据量越大以及用户个数越多,带来的通信成本就越高。

2 背景及研究动机

2.1 梯度攻击

文献[7]提出 DLG 梯度攻击方法,主要通过构

造随机初始化输入图像 x' 对应的模型梯度 $\nabla W'$ 与真实输入图像对应的梯度 ∇W 之间的欧式距离,利用 L-BFGS[5] 来求解出满足梯度间距离最小的输入图像 x', 梯度间欧式距离函数关系如式(1)所示。

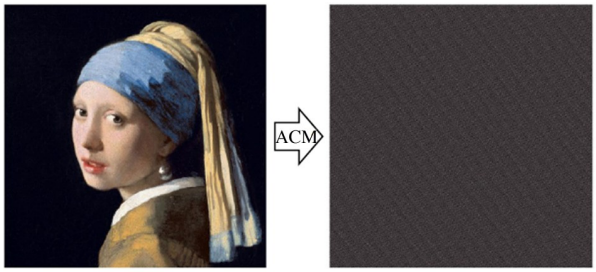
$$\arg\min_{x'} \| \nabla W' - \nabla W \|^2 = \arg\min_{x'} \| \frac{\partial L(F(x', W), y')}{\partial W} - \nabla W \|^2 \tag{1}$$

式中, F 代表深度学习模型, L 代表损失函数, y' 代表随机初始化输入图像对应的输出类别。因此,一旦恶意攻击者获得了模型的原始梯度 ∇W, 通过求解式(1)能恢复出模型的输入图像。另外,文献[9]通过将梯度间的欧式距离替换为 Cosine 相似度,并利用 ADAM 优化器来求解出最优的输入图像,实现了 IGA 梯度攻击。

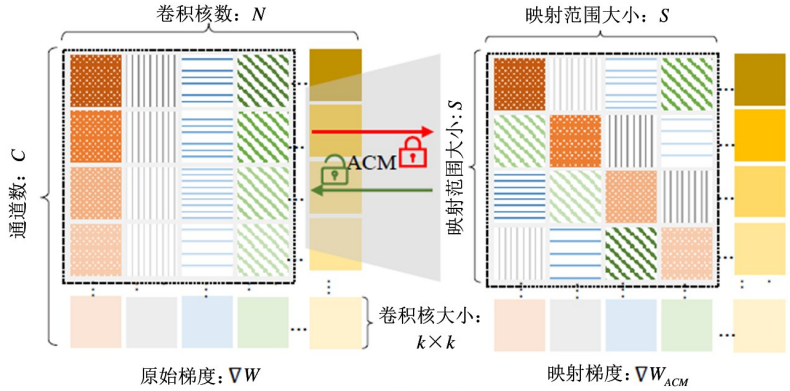
通过以上分析可知,如果对原始梯度 ∇W 施加某种数据保护措施,例如使用差分隐私方法对 ∇W 添加等级较大的噪声值,那么利用式(1)进行优化求解,最终会得到错误的输入图像,实现了模型梯度的安全保护。然而,添加噪声等级较大的值会带来严重的精度损失。本文希望能够完全不损失模型精度的前提下,高效快速地完成对模型原始梯度 ∇W 的保护。

2.2　ACM 混沌映射算法

通过分析梯度攻击原理可知,除了对模型梯度 ∇W 直接进行数值操作能够对梯度进行保护之外,还可以通过变换梯度内部值的位置来实现梯度保护。变换位置后的梯度与原始梯度的差异性越大,对梯度的安全保护效果就越好,且变换位置运算过程通常不需要数据加密算法中复杂的数学运算操作(如同态加密),能够节省运算的时延开销。基于上述考虑,本文希望利用图像隐私保护领域中的 Arnold's Cat Map (ACM)算法[21-22] 来保护模型的梯度。ACM 属于混沌映射理论的一种,最早由 Vladimir Arnold 提出并主要用于图像安全隐私保护。图 1(a)展示了使用 ACM 算法对图像映射之后的效果图。ACM 通过以特定方式交换图像中像素位置来实现安全保护,由于映射变换破坏了相邻像素之间的相关性,映射后的图像没有任何语义信息,可以达到与传统数值加密算法同样的保护效果,且映射过程运行效率较高、映射前后完全可逆,因此不会对模型造成精度损失。下文将详细介绍 ACM 映射算法的基本原理。



(a) 图像映射



(b) 梯度映射

图 1　ACM 映射示例图

映射 本文利用 ACM 算法通过变换深度学习模型中卷积层或全连接层的梯度位置来完成映射。如图 1(b)所示,以单层梯度 ∇W 为例,设深度学习模型第 l 层梯度的形状为 $C \times N \times k \times k$,其中 C 是通道数目、N 是卷积核的个数、k 是卷积核的大小,其值通常取为 1、3、5 和 7。当 $k = 1$ 时,可将 ACM 算法应用于全连接层。为了降低计算量同时节省映射时延开销,本文只对前两维,即 $C \times N$, 进行位置映射变换操作,实验结果证实了对梯度前两维进行映射变换是有效的。映射过程如式(2)所示。

$$\begin{bmatrix} i_e \\ j_e \end{bmatrix} = A^{\tau} \begin{bmatrix} i \\ j \end{bmatrix} (\mathrm{mod} S),\ A = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \tag{2}$$

其中,参数 p、q 和 τ 为正整数值,为 ACM 的映射参数。(i, j) 以及 $(i_{\mathrm{ACM}}, j_{\mathrm{ACM}})$ 分别为原始梯度的位置以及 ACM 映射后梯度的位置。S 代表映射范围的大小(映射范围须为正方形,即 $S \times S$),映射范围的大小须同时满足 $S \leqslant C$ 和 $S \leqslant N$, 且 ACM 映射能够在单层梯度的任意范围内实施,假设为 $[\alpha_1, \alpha_2] \times [\beta_1, \beta_2]$ $(0 \leqslant \alpha_1 < \alpha_2 \leqslant N$ 以及 $0 \leqslant \beta_1 < \beta_2 \leqslant C)$,则需要进行位置映射的集合 $\mathbb{I}$,如式(3)所示。

$$\mathbb{I} = \{(i, j) \mid \alpha_1 \leqslant i \leqslant \alpha_2; \beta_1 \leqslant j \leqslant \beta_2\} \tag{3}$$

其中,映射范围的大小 $S = (\alpha_2 - \alpha_1) = (\beta_2 - \beta_1)$。此外,ACM 能够对深度学习模型的任意层进行位置映射,因此映射层集合 $\mathbb{L}$、ACM 映射参数(即 p、q 和 τ)以及映射位置集合 $\mathbb{I}$ 构成 ACM 算法的映射因子,映射因子 P_{ACM} 的具体表达如式(4)所示。

$$P_{\mathrm{ACM}} = [\mathbb{L}, \{\tau^l, S^l, p^l, q^l \mid l \in \mathbb{L}\}] \tag{4}$$

其中,上角标 l 代表映射层的序号,映射后的梯度 ∇W_{ACM} 如图 1(b)右子图所示。

逆映射 当已知映射因子以及映射后对应的梯度位置 $(i_{\mathrm{ACM}}, j_{\mathrm{ACM}})$,原始梯度的位置 (i, j) 可以通过式(5)求得。

$$\begin{bmatrix} i \\ j \end{bmatrix} = A^{-\tau} \begin{bmatrix} i_{\mathrm{ACM}} \\ j_{\mathrm{ACM}} \end{bmatrix} (\mathrm{mod} S) \tag{5}$$

在式(2)和式(5)中,参数 τ 较大时,计算 A^{τ} 以及 $A^{-\tau}$ 的过程较为耗时,为了降低时延开销,快速求解出 A^{τ} 以及 $A^{-\tau}$, 本文将参数 p、q 值的大小设置为 1。此时, A^{τ} 的表达式可以表示为

$$A^{\tau} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix}^{\tau} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^{\tau} = \begin{bmatrix} f_{2\tau-1} & f_{2\tau} \\ f_{2\tau} & f_{2\tau+1} \end{bmatrix} \tag{6}$$

式(6)中,f_{τ} 为斐波那契数,即,

$$f_1 = 1, f_2 = 1, f_{\tau+2} = f_{\tau+1} + f_{\tau}, \tau = 1, 2, \cdots \tag{7}$$

利用斐波那契数列的递推公式,可以提前求得 A^{τ}, 因此节省了计算的时延开销。同理,逆映射参数 $A^{-\tau}$ 的表达式为

$$A^{-\tau} = \begin{bmatrix} f_{2\tau+1} & -f_{2\tau} \\ -f_{2\tau} & f_{2\tau-1} \end{bmatrix} \tag{8}$$

3 梯度安全保护算法

3.1 威胁模型及场景

为了验证所提算法的有效性,本文以联邦学习场景为例,展示了如何保护深度学习模型梯度的安全。图 2 显示了联邦学习的参与方,其中包括一个中央服务器和多个联邦用户。在联邦学习场景训练深度学习模型时,服务器方会接收到来自不同联邦用户的梯度,这些梯度是用户在本地端设备利用各自的隐私输入数据求得的。在此场景中,如果梯度没有进行安全保护操作,一旦被恶意服务器接收或者被第三方恶意用户通过中间人攻击截获,利用 2.1节提到的梯度攻击方法(例如,DLG 攻击以及 IGA 攻击)能完全恢复出用户的隐私输入数据,因此给深度学习模型梯度的安全带来了严重的威胁。本文将利用 ACM 算法对梯度进行安全保护,同时防止联邦学习场景中恶意服务器和恶意用户利用梯度还原用户的输入数据。

3.2 最优映射方案

利用 ACM 映射,用户可以对深度学习模型所有层的梯度值进行位置交换从而实现较好的保护效果。然而,对模型所有层的梯度值进行 ACM 映射会导致较高的时延开销,尤其是对计算资源受限的智能终端设备(如手机等移动端设备),时延开销会随着映射层数的增加而增大。为了能够获得具有良好的保护效果且能够快速高效地运行,本文通过最大化映射梯度 ∇W_{ACM} 与原始梯度 ∇W 之间差值,构

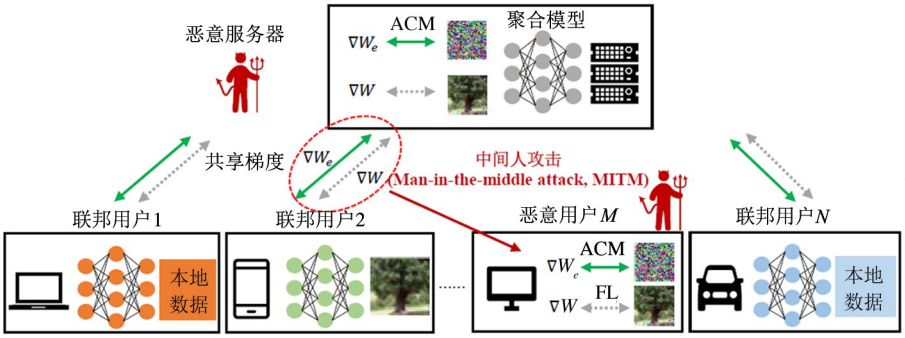


图 2　联邦学习中的梯度泄露

造了梯度差值最大化目标函数,并使得映射模型梯度总的时延开销 $T(P_{\mathrm{ACM}})$ 小于预先指定的最大时延 $t_{\max}$ 开销,具体如式(9)和式(10)所示。

$$P_{\mathrm{ACM}} = \operatorname{argmax} \| \nabla W_{\mathrm{ACM}}(P_{\mathrm{ACM}}) - \nabla W \|_1 \tag{9}$$

$$\text{s.t.} \quad T(P_{\mathrm{ACM}}) \leqslant t_{\max} \tag{10}$$

通过分析上述目标函数可知,最大化映射梯度 $\nabla W_{\mathrm{ACM}}(P_{\mathrm{ACM}})$ 与原始梯度 ∇W 的差值能够有效防御 DLG 及 IGA 攻击利用原始梯度直接还原用户的输入图像,这是因为恶意攻击者获得的梯度 $\nabla W_{\mathrm{ACM}}(P_{\mathrm{ACM}})$ 与原始梯度差值较大,使用式(1)破解时,会得到完全错误的优化结果,因此也就无法恢复用户的输入图像。直接求解式(9)会得到最优且唯一的映射因子 P_{ACM}。然而,唯一映射因子会增加恶意攻击者破解的可能性。

为了增大攻击者破解的难度,本文通过在优化目标式(9)中增加更多 ACM 映射因子选取的随机性。首先,对于需要映射的深度学习梯度层集合$\mathbb{L}$,在每一层中选取任意的映射范围。然后,利用这些层对模型进行 ACM 映射。单层梯度内大的映射范围 S 以及更多的映射层数能够产生良好的保护效果,但也会带来较大时延开销。为了使得映射后的模型能够取得较好的保护效果,且映射时延开销小于预期最大的时延开销 $t_{\max}$,将式(9)中的优化目标改进为如下优化目标:

$$\max \quad \sum\nolimits_l x^l \times \| \nabla W^l_{\mathrm{ACM}} - \nabla W^l \|_1 \tag{11}$$

$$\text{s.t.} \quad \sum\nolimits_l x^l \times t^l \leqslant t_{\max} \tag{12}$$

其中,$x^l \in \{0,1\}, l \in \mathbb{L}$。$x^l = 1$ 表示深度学习模型的第 l 层梯度需要进行 ACM 映射,相反 $x^l = 0$ 表示该层不进行 ACM 映射。t^l 表示第 l 层梯度的映射对应的时延开销。

通过分析上述公式可知,式(11)中的优化问题可以被看作经典的 0 - 1 整数背包问题,本文利用动态规划方法求解该整数背包问题。在算法 1 及算法 2中详细展示了求解式(11)中的优化目标的具体过程。在算法 1 中,首先,计算深度学习模型第 l 层的映射梯度和原始梯度之间距离的 L1 范数 d_l,其中 $d_l = \| \nabla w^l_{\mathrm{ACM}} - \nabla W^l \|_1$,$d_l$ 越大代表 ACM 映射后的梯度与原始梯度的差异性越大。之后,在硬件平台测量出每一层 ACM 映射对应的时延开销 t_l。最后,利用算法 2 中的动态规划算法求解出最优的保护方案,即需要使用 ACM 映射的层数。需要说明的是,整个算法的求解过程不会耗费大量的时延开销,原因在于:(1)在硬件平台测量映射时延开销 t_l 是非常快的,仅需要 L 次测量就可以完成,其中 L 是深度学习模型的层数。(2)计算深度学习模型的第 l 层映射梯度和原始梯度距离的 L1 范数也仅仅需要 L 次就能够完成。因此,通过算法 1 能求出每一层的映射因子 $\{\tau^l, S^l\}$,利用算法 2 可以求得需要映射的层集合$\mathbb{L}$,因此可得最终的映射因子 P_{ACM}。通过以上分析可以得出,本文提出的算法能够确保映射因子是有效且不唯一的,同时映射时延开销小于预期最大的时延开销。

算法 1　求解各层梯度间距以及映射时延

输入:原始梯度 ∇W; ACM 映射算法超参数 $\tau_{\max}$; 深度学习模型层数 L;每一层的卷积核个数 N^l 以及通道数 C^l

输出:梯度间距集合$\mathbb{D} = \{d_1, d_2, d_3, \cdots, d_L\}$; 时延集合 $\mathbb{T} = \{t_1, t_2, t_3, \cdots, t_L\}$; 映射因子集合$\mathbb{P} = \{\{\tau^l, S^l\} \mid l = 1, \cdots, L\}$

初始化：$\mathbb{D}\leftarrow\varnothing,\mathbb{T}\leftarrow\varnothing,\mathbb{P}\leftarrow\varnothing$

while $l \leqslant L$ do

　$\tau^l = \text{randint}(0, \tau_{\max})$

　$S^l = \text{randint}(0, \min\{C^l, N^l\})$

　$\nabla W^l_{ACM} = \text{ACM}(\nabla W^l, \tau^l, S^l)$ // $p=1, q=1$

　$d_l = \| \nabla W^l_e - \nabla W^l \|_1$

　$t_l = latency(\nabla W^l \rightarrow \nabla W^l_{ACM})$ //时延测量

　$\mathbb{D}\leftarrow d_l, \mathbb{T}\leftarrow t_l, \mathbb{P}\leftarrow[l, \{\tau^l, S^l\}]$

end while

算法 2　动态规划求解 0－1 背包问题

输入：梯度间距集合 $\mathbb{D}=\{d_1,d_2,d_3,\cdots,d_L\}$；映射时延集合 $\mathbb{T}=\{t_1,t_2,t_3,\cdots,t_L\}$；预期最大时延开销 $t_{\max}$；深度学习模型层数 L

输出：映射层集合 $\mathbb{L}$

初始化：二维数组 $dp[L, t_{\max}]$, $Keep[L, t_{\max}]$, $\mathbb{L}\leftarrow\varnothing$

for l in range $(1,2,\cdots,L)$

　for t in range $(0,1,\cdots,t_{\max})$

　　$t_l = \mathbb{T}[l-1]$, $d_l = \mathbb{D}[l-1]$

　if $t_l \leqslant t$ and $(d_l + dp[l-1, t-t_l] > dp[l-1, t])$

　　$dp[l,t] = d_l + dp[l-1, t-t_l]$

　$Keep[l, t] = 1$

　else

　$dp[l, t] = dp[l-1, t]$

$K = t_{\max}$

for l in range $(L, 0, -1)$

　if $Keep[l, K] == 1$

　　$\mathbb{L}\leftarrow l$

　　$K -= \mathbb{T}[l-1]$

return $\mathbb{L}$

3.3　双重映射机制

为了能够同时防止恶意服务器以及恶意用户利用深度学习模型的梯度来偷窥用户输入，本文提出对需要上传的模型梯度进行双重 ACM 映射，来实现保护模型梯度的安全性。图 3 展示了双重映射机制上传梯度的流程。其中，第一轮映射是为了防止恶意服务器通过梯度来偷窥用户的输入，此时所有用户共享同一个映射因子来保护模型梯度，本文将其称为用户-用户映射。第二轮映射是为了防止恶意用户通过中间人攻击窃取梯度来偷窥其他用户的输入。因此，每个用户对梯度进行第二轮映射时，每个用户的映射因子只与服务器进行共享，且各用户之间的映射因子不同，因此本文将其称为用户-服务器映射。映射因子使用 3.2 节提出的方法生成。

为了在服务器端进行梯度聚合操作，当服务器收到不同用户传来的双重映射梯度后，首先根据用户-服务器映射因子对双重映射的梯度进行一次逆映射操作。逆映射一次后的梯度依然处于映射状态（即第一轮映射梯度状态），此时服务器没有用户-用户之间的映射因子，因此恶意服务器要想偷窥用户的输入，就需要破解映射后的梯度。由于不同用户上传的第一次映射后梯度具有相同的映射因子，表明所有用户都对梯度进行了相同方式的位置变换，而联邦学习中常见的聚合操作是线性的求均值操作，因此映射后的梯度能够在第一轮映射状态进行聚合操作。最后，服务器将保持映射状态的聚合梯度返回给不同的用户，不同用户根据用户-用户映射因子对梯度进行一次逆映射操作，并利用完全逆映射后的梯度（即原始梯度）在本地更新模型的参数，并进行下一轮的学习。

恶意用户只有用户-用户映射因子以及该恶意用户与服务器之间的映射因子，没有其他用户与服务器之间的映射因子，因此无法直接对双重映射的梯度进行两次逆映射来偷窥其他用户的输入。同

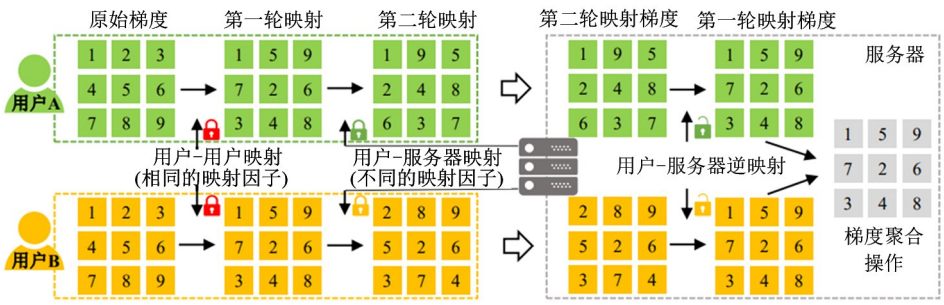


图 3　双重映射示意图

理,恶意服务器只有不同用户与服务器之间的映射因子,但没有用户之间的映射因子,因此也无法对双重映射的梯度进行二次逆映射,也无法直接利用梯度攻击方法来偷窥用户的输入。综上,本文提出的双重映射机制不但能阻止恶意服务器通过梯度偷窥用户的输入隐私,同时还能防止恶意用户偷窥用户的输入,因此确保了模型梯度的安全性。

3.4　安全性分析

ACM 映射算法主要应用于图像隐私保护领域,本文利用其效率高且安全保护效果好等特点将其应用于深度学习模型的梯度安全保护。对于 ACM 在图像领域的保护,存在两种已知的攻击方式,分别为已知明文攻击(known-plaintext attack)和选择明文攻击(chosen-plaintext attack)[23-24]。一方面,已知明文攻击假设攻击者已知明文与密文对,然而在梯度安全保护场景下,恶意服务器只知道映射梯度而无法得知原始梯度。同理,恶意用户只能获得本地原始梯度与映射后的梯度对,而无法获得其他用户原始的梯度。因此,已知明文攻击对于本文的梯度保护场景是无法实施的。另一方面,对于选择明文攻击,假设攻击者能够访问映射的内部机制。得益于双重映射协议,本文提出的方法能有效防止此类攻击。原因在于恶意服务器只能访问自身与不同用户的映射机制,而无法得知用户之间的映射方式。同理,恶意用户只能访问本地和服务器之间的映射机制,而无法得知其他用户与服务器之间的映射机制。因此,应用于图像领域的已知明文攻击和选择明文攻击在深度学习模型梯度保护领域是不起作用的。

另一种可能的攻击方法是暴力搜索,根据上文分析可知,本文所提方法能映射深度学习模型的任意层以及任意范围,因此暴力搜索的攻击复杂度较大,接近 $O(2^L\prod_{l=1}^{L}\tau^l\min\{(N^l)^2C^l,(C^l)^2N^l\})$。以下文实验中使用的 ConvNet64 模型为例(模型结构详见文献[9]),该模型具有 8 层网络结构(即 $L=8$),由于第一层可映射范围较小,因此只考虑映射后 7 层梯度,假设 ACM 映射参数 τ 的取值范围为 10,则暴力搜索的攻击复杂度接近 4.7×10^{56}。即使攻击者使用算力强大的计算机来破解,假设能够每秒验证 1000 次,破解本文所提梯度保护方法需要花费 4.7×10^{53} s,约等于 1.5×10^{46} a。

4　实验仿真

4.1　实验设置及数据集

本文在两类最新的攻击方法上验证了对梯度的安全保护效果,分别为 DLG 梯度攻击[7]以及 IGA 梯度攻击[9]。本文提出的梯度安全保护方法适用于各类深度学习模型,包括浅层模型以及深层模型。其中,使用文献[7]中提到的浅层模型在 CIFAR-100[25]数据集上,评估防御 DLG 攻击方法的有效性。使用在文献[9]中提到的深层 ConvNet-64 模型以及 ResNet-18 模型[11]在 CIFAR-10[25]、LFW[26]以及 ImageNet[27]数据集上来评估防御 IGA 攻击方法的有效性。

对于防御 DLG 攻击,采用原始文献中的添加噪声的 DP 防御方法作为基准进行对比。由于基于同态加密的 LWE 防御方法[12]与 DLG 攻击方法使用的浅层模型参数量接近,因此本文将其作为实验基准进行分析说明。对于防御 IGA 攻击,由于模型的层数较深且参数量较大,在原始文献中没有给出相关防御策略,本文提出将 ACM 方法直接应用于深层模型的所有卷积层作为实验基准进行对比并分析其时延开销。

实验中使用的评价指标包括:(1)峰值信噪比(peak signal to noise ratio, PSNR),用来评估梯度攻击之后的图像恢复效果。PSNR 越小,表示恢复的图像质量越差,代表本文提出的梯度安全保护方法效果越好。(2)时延开销,用来评价算法在硬件平台上的运行效率。时延开销越小,代表算法运行效率越高。表 1 展示了本文使用的硬件实验平台。在时延开销评估方面,对于算力较弱的硬件平台,本文使用 3 款移动手机芯片 Qualcomm Snapdragon 450、-625 以及 -835 来进行实验,并使用C ++ 来实现本文所提算法,同时使用安卓 NDK 库对程序进行编译。对于算力较强的硬件平台,本文使用 Intel E5-2650 V4 CPU 和 NVIDIA TITAN V GPU 进行实验,分别使用C ++ 和 CUDA 实现本文所提算法。

表 1　实验中使用的硬件平台

	Snapdragon			Intel E5-2650	TITAN V
	450	625	835	V4 CPU	GPU
核数	8	8	4	12	5120
面积/mm^2	-	-	72.3	-	815
功率/W	3	5	9	105	250
内存/GB	4	3	6	64	12
工艺/nm	14	14	10	14	12
频率/GHz	1.8	2	2.45	2.9	1.45

4.2　实验结果及分析

4.2.1　映射因子分析

图 4 以及图 5 分别展示了不同映射因子参数对 ConvNet 64 模型的保护效果以及时延开销的影响。其中，保护效果利用映射后的梯度与原始梯度之间差值的 L1 范数（$\| \nabla W^l_{ACM} - \nabla W^l \|_1$）来表示。为了降低映射时延开销，在映射因子式(4)中设置 $p = 1$ 以及 $q = 1$。因此，模型梯度的映射时延开销以及

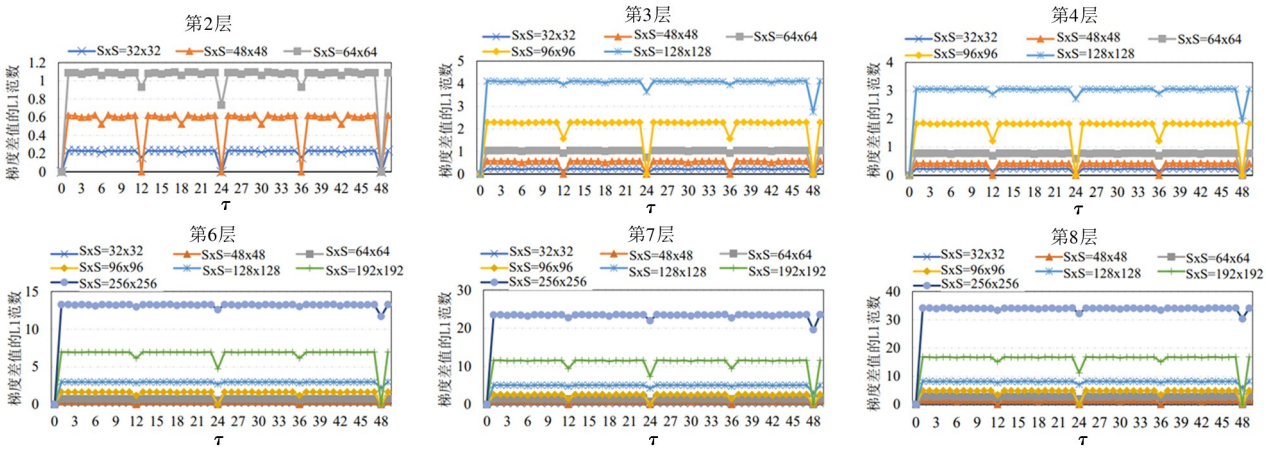


图 4　映射因子对 ConvNet 64 模型梯度保护效果的影响

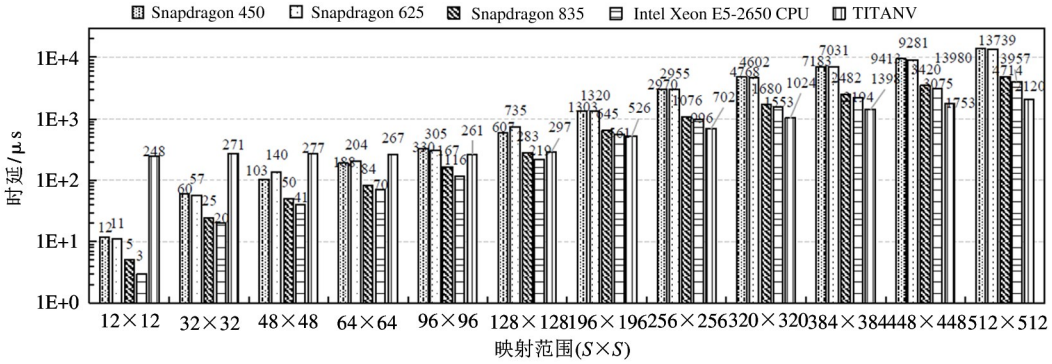


图 5　映射范围大小对时延开销的影响

保护效果主要取决于层集合$\mathbb{L}$、映射参数 τ 以及映射范围大小 S。

映射层集合$\mathbb{L}$的影响。图 4 展示了不同映射因子对 ConvNet 64 模型映射效果的影响。从图中能够得出，模型靠后层的保护效果要好于靠前层的保护效果。例如，当映射范围 $S \times S = 128 \times 128$ 以及 $\tau = 5$ 时，模型第 8 层对应的梯度差值 L1 范数大小为8.21，第 7 层对应梯度的差值 L1 范数大小为 5.06，而第 6 层对应的梯度差值 L1 范数大小为 2.99。梯度间的差值 L1 范数越大表明映射后的梯度与原始梯度差异越大，此时梯度攻击方法利用映射后的梯度就越难以恢复出原始的输入图像。

映射参数 τ 的影响。从图 4 中能够得出，τ 在大部分取值情况下，不会对保护效果有较大差别的影响。举例来说，当映射 ConvNet64 的第 8 层梯度及 $S \times S = 96 \times 96$ 时，随着 τ 的不断增大，除了 τ 为 12 的倍数对应的梯度差值距离较小之外，大部分梯度差值 L1 范数距离为 4.9，因此参数 τ 具有周期

性,文献[28]中给出了其周期性取决于映射范围以及映射参数 p 和 q。因此,在随机选取映射参数 τ 时,应该剔除那些导致保护效果较差的值。

映射范围大小 S 的影响。映射范围的大小不仅会影响保护效果,同时还会影响时延开销。越大的映射范围,保护效果就越好,但随之也带来了较大的时延开销。例如,如图4所示,对于ConvNet 64模型的第8层梯度以及 τ 取值为6时,映射范围 $S\times S=256\times256$、196×196 以及 128×128 对应的梯度间距的L1范数分别为33.8、16.5以及7.9。然而,图5展示了不同映射范围在不同硬件平台的时延开销对比,当映射范围为 $S\times S=256\times256$ 时,在Intel Xeon E5-2650 v4 CPU上对应的时延开销为996 μs,比 $S\times S=196\times196$ 以及 $S\times S=128\times128$ 对应的时延开销分别快435 μs以及777 μs。

4.2.2　防御DLG攻击

图6以及表2展示了本文所提方法防御DLG攻击方法的效果,实验中对比了基于同态加密的LWE方法[12]的加密时延开销,还比较了对梯度添加高斯噪声(Gaussian)以及拉普拉斯(Laplacian)噪声的差分隐私方法(differential privacy, DP)。其中,差分隐私方法在文献[7]进行了实验验证,本文利用文献中的实验数据在表2中进行分析说明。

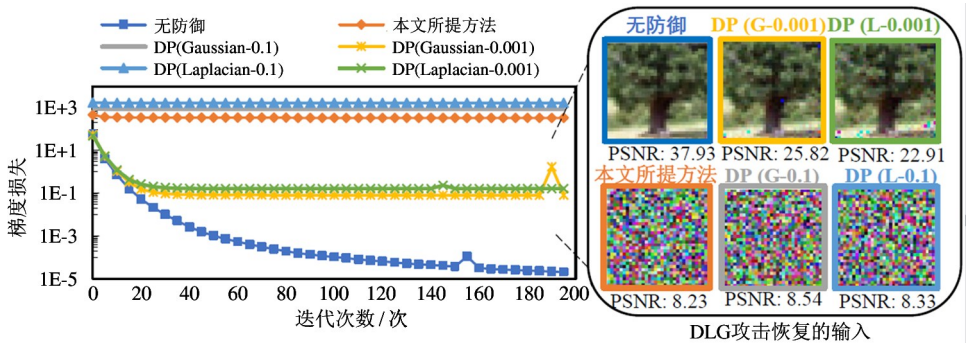


图6　防御DLG攻击的效果

表2　不同防御方法性能对比

	DP				LWE	本文所提方法	
	G-10^{-3}	G-10^{-1}	L-10^{-3}	L-10^{-1}	CPU	CPU	GPU
精度损失	3%	75.3%	3%	75.3%	0%	0%	0%
峰值信噪比	25.82	8.54	22.91	8.33	-	8.23	8.23
时延开销	~0	~0	~0	~0	>454 ms	6 μs	496 μs
防御能力	×	√	×	√	√	√	√

从图6能够得出,如果不对梯度增加任何保护措施(即图中的无防御),DLG攻击方法能够持续不断地优化梯度损失,直到损失接近为0,此时峰值信噪比(PSNR)达到最大值37.93,从图中看出输入图像已经几乎完全被还原出来。差分隐私方法添加较大级别的噪声时能够阻止DLG攻击,但对模型在原始任务上的精度造成了严重的影响。例如,DP(L-0.1)(拉普拉斯噪声为0.1级别)对应的梯度匹配损失几乎没有下降,且利用DLG攻击方法恢复出来的输入图像的峰值信噪比(PSNR)为8.33。由于恢复之后的图像没有任何语义内容信息,因此实现了对梯度安全保护的目标,然而模型的精度损失为75.3%,如表2所示。而差分隐私方法添加较小级别的噪声无法对梯度起到安全保护作用,因此无法阻止攻击者使用DLG方法还原输入数据。例如,DP(L-0.001)(拉普拉斯噪声为0.001级别)对应的峰值信噪比为22.91,从图能够看出输入图像的内容信息。

与差分隐私相反，基于同态加密的 LWE 方法以及本文所提方法都能够有效防止 DLG 攻击还原输入图像，且能够保证精度不下降。文献[7]的 4 层的深度学习模型大约包含 85 000 个梯度，使用 LWE 方法需要至少花费大约 450 ms 才能在一块 Intel Xeon E5-2660 v3 CPU 进行加密。本文所提方法仅需要花费大约 6 μs 就能对该模型在一块 Intel Xeon E5-2650 v4 CPU 上实现保护，因此能够节省大量的时延开销。所提方法时延开销较小的原因在于，没有利用复杂的加密运算对 85 000 个梯度值进行逐个加密（例如，同态加密），而是直接对梯度中的整个卷积核进行 ACM 映射，且能够选择部分映射范围，因此大幅降低了时延开销。此外，本文也在一块 TITAN V GPU 实现了对该模型的保护，实验数据显示需要花费大约 496 μs 能完成映射过程。GPU 上时延开销比 CPU 大的原因是调用 CUDA 核函数本身会花费几百微秒的时延。

4.2.3　防御 IGA 攻击

图 7 展示了在不同的梯度防御措施下，恶意攻击者利用 IGA 攻击通过模型梯度恢复的输入图像。图中 T-US 中的 T 代表期望的最大时延开销（即 t_{max}）为 Tus，时延开销在一块 Intel Xeon E5-2650 v4 CPU 上进行测试，此外 US 表示联邦学习中的恶意服务器仅有服务器（Server）与客户（User）之间的映射因子，而没有客户与客户之间的映射因子。举例来说，在 1000-US 设定参数条件下，本文提出的方法大约需要 999 μs 就能对 ConvNet64 模型完成安全保护。利用 IGA 攻击对 1000-US 设置下的梯度进行攻击，恢复的输入图像峰值信噪比（PSNR）为 0.93，远小于无保护措施下的峰值信噪比 11.42。对于更深层的 ResNet-18 模型，在 5000-US 设定条件下，本文所提方法需要 4627 μs 来保护梯度，恢复的输入图像峰值信噪比为 -0.82，几乎没有任何语义信息。对于 ConvNet64 模型，在 500-US 设定条件下，对于梯度保护的效果均较差，这是由于设定的最大时延开销过小（t_{max} 为 500 μs）而导致的。此外，在图 7 中还展示了利用 ACM 方法对模型所有层梯度映射保护后的 IGA 梯度攻击防御效果。由于对模型所有层梯度都进行了映射操作，因此 ACM 方法能有效保护模型梯度的安全性，但时延开销比本文所提方法要高。因此，本文在实现良好梯度保护效果的同时，能够保持较小的时延开销。

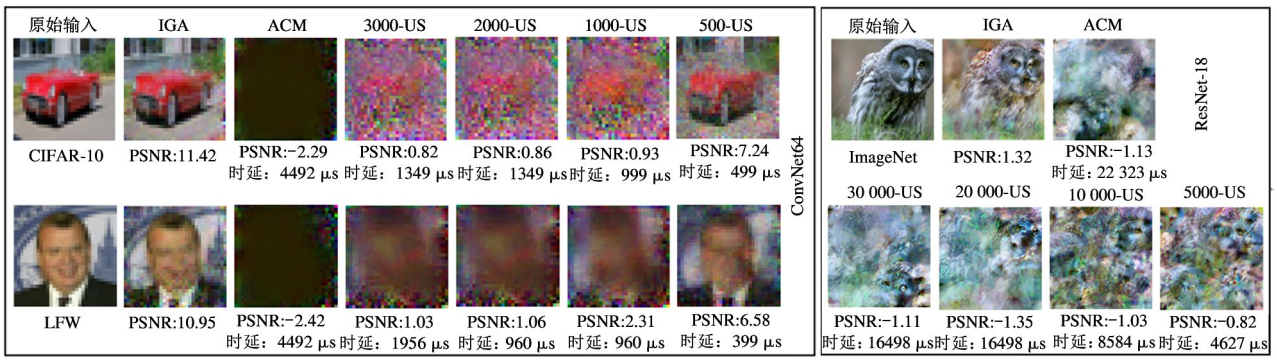


图 7　防御 IGA 攻击的效果

除了需要防止恶意服务器利用模型的梯度偷窥用户隐私之外，还要防止恶意用户利用中间人攻击通过梯度偷窥用户的输入。如图 8 所示，一旦攻击者同时拥有用户之间的映射因子（UU 映射因子）及其他用户和服务器之间的映射因子（US 映射因子），此时利用 IGA 攻击能够完全恢复用户的输入图像，如子图 3000-US&UU 所示。然而在实际情况中，恶意用户只有用户之间的映射因子（UU 映射因子），没有其他用户与服务器之间的映射因子（US 映射因子）。因此，恶意用户无法利用 IGA 攻击通过梯度来还原其他用户的输入图像，如子图 3000-UU 所示恶意用户利用中间人攻击无法获取。如果攻击者既没有用户之间的映射因子，又没有用户与服务器之间的映射因子，则完全无法利用 IGA 攻击来偷窥用户的输入数据。此外，在 500-UU 设定参数条件下，由于最大时延开销过小（t_{max} 为 500 μs），

因此防御能力较差。

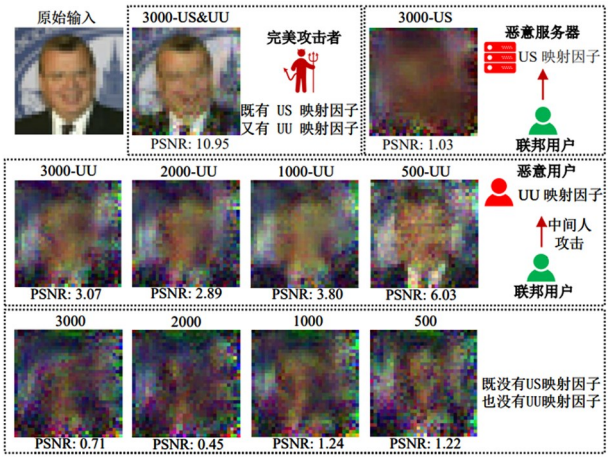


图 8　防御 IGA 中恶意用户攻击

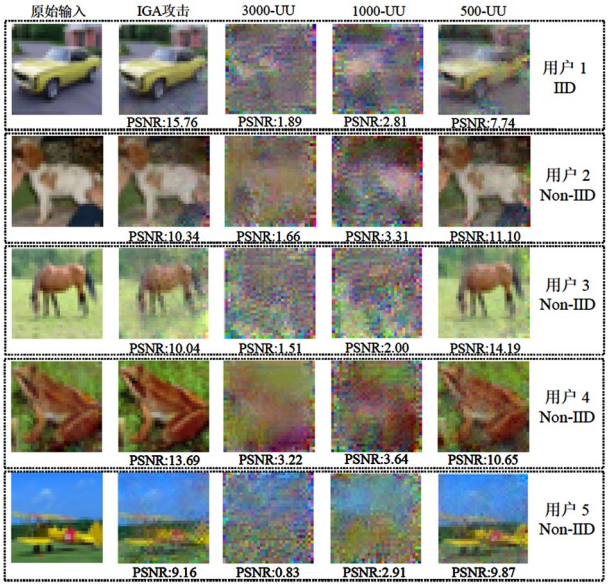


图 9　用户间映射因子的适用性

4.2.4　用户间的映射因子适用性

第一次对梯度进行 ACM 映射时所有用户使用的是某个用户生成的映射因子来对梯度进行位置交换。在联邦学习任务中,不同用户会具有不同类型的输入数据,这些输入数据可能是类似的,服从独立同分布(independent and identically distributed,IID),对模型求得的梯度可能是类似的。也可能不是类似的,服从非独立同分布的(Non-IID),求出的梯度不是类似的。在图 9 中,所有用户使用的是同一映射因子进行安全保护,该映射因子为 ACM 映射算法在图 7 的小汽车数据所产生的梯度上的优化结果。从映射后防御 IGA 攻击的效果能够得出,当期望的最大时延开销 $t_{\max}$ 为3000 μs或1000 μs时,由于图 9 中的小汽车具有与图 7 中的小汽车相同的属性,属于独立同分布(IID)数据,因此图 9 中的小汽车产生的梯度值能够使用图7中的小汽车产生的梯度对应的映射因子进行混沌映射保护,且具有良好的防御效果。此外,对于其他用户拥有比如狗、马、青蛙和飞机等输入对应的模型梯度,使用小汽车对应梯度的映射因子也能对其进行有效保护。这是因为当最大时延 $t_{\max}$ 设置较大时,能够对深度学习模型中较多层的梯度进行混沌映射,并取得较好的保护效果。

4.2.5　移动端平台性能

为了评估本文提出方法在计算资源受限的端设备上的性能表现,图 10 展示了 3 款手机芯片 Qualcomm Snapdragon 835、－625 和 －450 上梯度保护效果以及对应的时延开销,同时也对比了性能较强的 CPU 和 GPU 上的实验结果。在 3 款手机芯片上,性能较好的 Qualcomm Snapdragon 835 芯片在不同的实验配置下都能产生较好的梯度保护效果。性能较差的 Snapdragon 625 以及 Snapdragon 450 芯片,

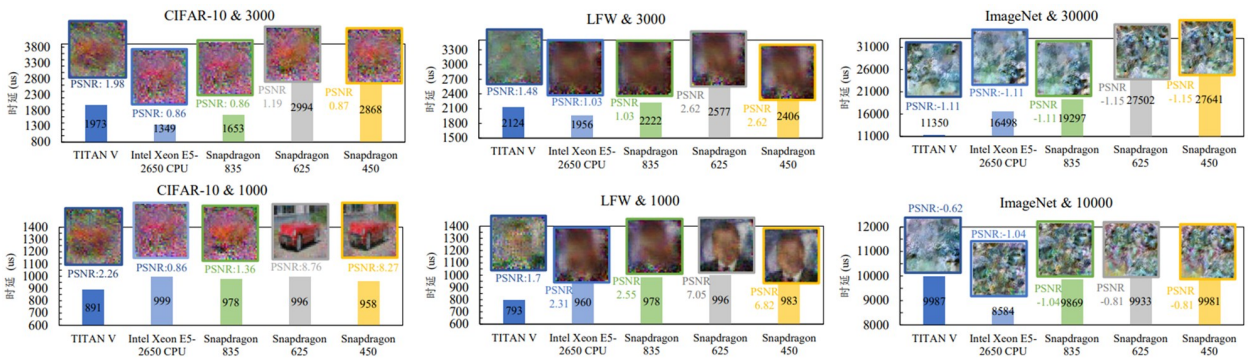


图 10　不同平台的保护效果

保护效果较差,原因是混沌映射时延限制要求较高而芯片性能较差。例如,在 1000(即 t_{max} 为1000 μs)实验设置下,对梯度的安全保护效果较差。而当实验配置设置为 3000 时(即 t_{max} 为 3000 μs),Snapdragon 625 以及 Snapdragon 450 芯片的保护效果几乎与 Snapdragon 835 相同。综上,本文所提方法能够保护模型梯度的安全且时延开销较小,在毫秒级别就能完成梯度安全保护。

5 结 论

本文提出一种基于双重混沌映射算法的深度学习模型梯度安全保护方法,在联邦学习场景中能够同时阻止恶意用户和服务器利用梯度攻击方法恢复用户的输入隐私数据。与安全多方计算、同态加密和差分隐私方法不同,本文所提方法利用 ACM 映射算法通过交换深度学习模型梯度的位置来实现梯度保护。由于不需要对深度学习模型所有梯度值进行映射,而是将深度学习模型梯度层的混沌映射问题转化为背包问题,并利用动态规划求解出最优的保护方案,因此节省了梯度保护的时延开销。此外,对每一层梯度进行混沌映射时,任意映射范围增加了映射因子生成的随机性,也增大了攻击者破解的复杂度。此外,与差分隐私方法相比较,本文提出的方法完全不损失精度;与安全多方计算方法相比,本文所提方法通信开销可以忽略不计。所提方法的有效性在当前最新两类梯度攻击方法——DLG 和 IGA 上得到了验证。最后,在计算能力较低的移动端芯片平台上验证了混沌映射的时延开销,实验结果表明所提方法能在毫秒级别完成计算。

人工智能算法已经在日常生活的各种场景得到了大量的应用,本文希望该项研究工作能够促使人工智能算法的研究和应用人员更多地关注算法的安全性,同时能够探索更高效的安全保护方法。

参考文献

[1] MCMAHAN B, MOOREE, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[C]// Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, USA, 2017:1-10

[2] SHOKRI R, SHMATIKOV V. Privacy-preserving deep learning[C]// Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, USA, 2015:1-12

[3] YANG Q, LIU Y, CHEN T, et al. Federated machine learning: concept and applications[J]. *ACM Transactions on Intelligent Systems and Technology*, 2019, 10(2): 1-19

[4] 杨强,刘洋,程勇,等. 联邦学习 = Federated Learning[M]. 北京: 电子工业出版社, 2020

[5] 杨强,黄安埠,刘洋,等. 联邦学习实战[M]. 北京: 电子工业出版社, 2021

[6] 杨强. AI 与数据隐私保护: 联邦学习的破解之道[J]. 信息安全研究, 2019, 5(11): 961

[7] ZHU L, LIU Z, HAN S. Deep leakage from gradients[EB/OL]. https://arxiv.org/pdf/1906.08935.pdf: arXiv, (2019-12-19), [2021-07-01]

[8] LIU D C, NOCEDAL J. On the limited memory BFGS method for large scale optimization[J]. *Mathematical Programming*, 1989, 45(1): 503-28

[9] GEIPING J, BAUERMEISTER H, DRÖGE H, et al. Inverting Gradients—How easy is it to break privacy in federated learning[EB/OL]. https://arxiv.org/pdf/2003.14053v1.pdf: arXiv, (2020-03-31), [2021-07-01]

[10] KINGMA D P, BA J. Adam: a method for stochastic optimization[EB/OL]. https://arxiv.org/pdf/1412.6980v1.pdf: arXiv, (2014-12-22), [2021-07-01]

[11] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[EB/OL]. https://arxiv.org/pdf/1512.03385.pdf: arXiv, (2015-12-10), [2021-07-01]

[12] PHONG H T, AONO Y, HAYASHI T, et al. Privacy-preserving deep learning via additively homomorphic encryption[J]. *IEEE Transactions on Information Forensics and Security* 2017, 13(5): 1333-1345

[13] FONTAINE C, GALAND F. A survey of homomorphic encryption for nonspecialists[J]. *EURASIP Journal on Information Security*, 2007(2007): 1-10

[14] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[C]// Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Berlin, Germany, 1999:223-238

[15] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. https://arxiv.org/pdf/1409.1556.pdf: arXiv, (2015-03-10), [2021-07-01]

[16] YONETANI R, NARESH BODDETI V, KITANI K M, et al. Privacy-preserving visual learning using doubly permuted homomorphic encryption[C]// Proceedings of the IEEE

International Conference on Computer Vision, Venice, Italy, 2017: 2059-2069

[17] DWORK C, MCSHERRY F, NISSIM K, et al. Calibrating noise to sensitivity in private data analysis[C]// Proceedings of the Theory of Cryptography Conference, New York, USA, 2006: 265-284

[18] LI W, MILLETARilletarÌ F, XU D, et al. Privacy-preserving federated brain tumour segmentation[EB/OL]. https://arxiv.org/pdf/1910.00962.pdf: arXiv, (2019-10-02), [2021-07-01]

[19] YAO A C. Protocols for secure computations[C]// Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, Chicago, USA, 1982: 160-164

[20] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning[C]// Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas,USA, 2017:1175-1191

[21] ARNOL′D V I, AVEZ A. Ergodic Problems of Classical Mechanics[M]. New York:Benjamin Inc,1968

[22] PETERSON G. Arnold's cat map[J]. *Math* 45-*Linear AlgebraFall*, 1997, 45: 1-7

[23] HANOUTI I E, FADILI H E, ZENKOUAR K. Breaking an image encryption scheme based on Arnold map and Lucas series[EB/OL]. https://arxiv.org/ftp/arxiv/papers/1910/1910.11678.pdf: arXiv, (2019-10-19), [2021-07-01]

[24] COKAL C, SOLAK E J P L A. Cryptanalysis of a chaos-based image encryption algorithm[J]. *Physics Letters A*, 2009, 373(15): 1357-1360

[25] KRIZHEVSKY A, HINTON G. Learning Multiple Layers of Features From Tiny Images, TR2009[R]. Toronto: University of Toronto, 2009

[26] HUANG G B, MATTAR M, BERG T, et al. Labeled Faces in the Wild: A Database For studying Face Recognition in Unconstrained Environments, Technical Report 07-49[R]. Amherst: University of Massachusetts, 2008

[27] DENG J, DONG W, SOCHER R, et al. ImageNet: a large-scale hierarchical image database[C]// Proceedings of the 2009 IEEE Conference On Computer Vision And Pattern Recognition, Miami, USA, 2009: 248-255

[28] BAO J, YANG Q J N D. Period of the discrete Arnold cat map and general cat map [J]. *Nonlinear Dynamics*, 2012, 70(2): 1365-1375

The gradients protection for deep learning models based on dual chaotic map

LIN Ning, CHEN Xiaoming, XIA Chunwei, LI Wenxing, YE Jing, LIU Zizhen, LI Xiaowei

(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101408)

Abstract

In the federated learning task, different users will upload the gradients of deep learning models to a central server for gradients aggregation. However, recent studies show that directly uploading the original gradients is not secure, and attackers can utilize gradient attack methods to restore user's input data. Currently, methods based on secure multi-party computation (SMPC), differential privacy (DP) and homomorphic encryption (HE) to protect gradient security have major problems of large communication overhead, serious loss of accuracy and excessive latency overhead. This paper proposes a gradient security protection method based on a dual chaotic map algorithm, which can prevent malicious users and malicious servers from peeping users' personal privacy through gradients by exchanging its positions. To reduce the latency overhead, the proposed method transforms the map problem of the layers into a 0 – 1 integer knapsack problem, and utilizes dynamic programming to obtain the optimal encryption scheme. Experimental results on CIFAR-10, CIFAR-100, LFW and ImageNet datasets show that the proposed method can effectively defend the two latest gradient attack methods, and effectively protect the security of the gradients. In addition, the experimental results on CPU, GPU and three mobile phone chips show that the proposed method runs extremely efficiently and only requires several milliseconds to achieve security protection.

Key words: deep learning, gradients security, chaotic map, knapsack problem, dynamic programming