软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(6):2411−2430 [doi: 10.13328/j.cnki.jos.007648] [CSTR: 32375.14.jos.007648]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

恶意敌手环境下的隐私保护目标检测*

肖欣怡[1], 柳 林[1,2], 郭 茜[1], 罗玉川[1], 王勇军[1], 陈荣茂[1,2], 黄俊杰[1], 刘天瑞[1], 付绍静[1,2]

[1](国防科技大学 计算机学院, 湖南 长沙 410073)

[2](国防科技大学 密码中心, 湖南 长沙 410073)

通信作者: 柳林, E-mail: liulin16@nudt.edu.cn

摘 要: 图像处理任务正快速向云端和多方协同环境迁移, 而云服务器上直接处理明文图像数据, 极易泄露图像中的敏感信息, 且难以抵御篡改等恶意攻击, 无法保证数据完整性和服务可靠性. 在此背景下, 提出一种面向恶意敌手环境的目标检测推理方案——MalOD, 实现针对恶意敌手环境的安全目标检测. MalOD 通过构建加密的特征金字塔网络 (secure feature pyramid network, SecFPN) 实现密文图像的多级特征提取, 并基于多层次密文特征设计安全区域提议网络 (secure region proposal network, SecRPN) 和兴趣区域安全对齐 (secure region of interest align, SecRoIA) 模块从而完成安全目标检测. 具体来说, 借助复制秘密共享 (replicated secret sharing, RSS) 技术, 设计一系列安全计算原语, 包括安全向上取整函数、安全双线性插值和安全最近邻插值, 为 SecFPN、SecRPN、SecRoIA 等模块提供底层支撑, 确保恶意敌手环境下检测流程的高效与准确. 此外, 证明 MalOD 的正确性和安全性, 并在 COCO 2017 和 Pascal VOC 2012 数据集上进行性能评估. 实验结果表明, 在满足严格安全要求的同时, MalOD 实现较高的目标检测精度. 特别地, 当目标检测的交并比阈值为 0.5 时, 其在 COCO 子集上平均精度仅比明文检测下降 0.113. 为恶意环境下的隐私保护图像处理提供了理论和实践支持, 尤其适用于不可信的云计算和多方协作场景中.

关键词: 隐私保护; 目标检测; 恶意敌手; 安全多方计算

中图法分类号: TP309

中文引用格式: 肖欣怡, 柳林, 郭茜, 罗玉川, 王勇军, 陈荣茂, 黄俊杰, 刘天瑞, 付绍静. 恶意敌手环境下的隐私保护目标检测. 软件学报, 2026, 37(6): 2411–2430. http://www.jos.org.cn/1000-9825/7648.htm

英文引用格式: Xiao XY, Liu L, Guo Q, Luo YC, Wang YJ, Chen RM, Huang JJ, Liu TR, Fu SJ. Privacy-preserving Object Detection Under Malicious Adversaries. Ruan Jian Xue Bao/Journal of Software, 2026, 37(6): 2411–2430 (in Chinese). http://www.jos.org.cn/1000-9825/7648.htm

Privacy-preserving Object Detection Under Malicious Adversaries

XIAO Xin-Yi[1], LIU Lin[1,2], GUO Qian[1], LUO Yu-Chuan[1], WANG Yong-Jun[1], CHEN Rong-Mao[1,2], HUANG Jun-Jie[1], LIU Tian-Rui[1], FU Shao-Jing[1,2]

[1](College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China)

[2](Center for Cryptologic Research, National University of Defense Technology, Changsha 410073, China)

Abstract: Image processing tasks are rapidly migrating to cloud and multi-party collaborative environments. However, directly processing plaintext image data on cloud servers easily leads to the leakage of sensitive information in images and is difficult to resist malicious attacks such as tampering, thus failing to guarantee data integrity and service reliability. To address these challenges, this study proposes MalOD, an object detection inference framework for environments with malicious adversaries. MalOD is a framework to achieve secure object detection under malicious adversaries. MalOD constructs an encrypted feature pyramid network (SecFPN) to perform multi-level feature extraction on encrypted images. Based on these multi-level cipher text features, a secure region proposal network (SecRPN) and a

* 基金项目: 国家自然科学基金 (62472431); 湖南省自然科学基金 (2023JJ30640)
收稿时间: 2025-11-12; 修改时间: 2026-01-13; 采用时间: 2026-01-30; jos 在线出版时间: 2026-04-09
CNKI 网络首发时间: 2026-04-10

secure region of interest align (SecRoIA) module are designed to achieve secure object detection. By leveraging replicated secret sharing (RSS), a series of secure computation primitives are designed, including a secure ceiling function, secure bilinear interpolation, and secure nearest-neighbor interpolation. These primitives provide the underlying support for SecFPN, SecRPN, and SecRoIA, ensuring the efficiency and accuracy of the detection process under malicious adversaries. The correctness and security of MalOD are proved, and its performance is evaluated on the COCO 2017 and Pascal VOC 2012 datasets. Experimental results show that MalOD achieves high object detection accuracy while meeting strict security requirements. In particular, when the intersection over union (*IoU*) threshold is 0.5, the average precision on the COCO subset decreases by only 0.113 compared with plaintext detection. This study provides theoretical and practical support for privacy-preserving image processing under malicious environments and is particularly suitable for untrusted cloud computing and multi-party collaboration scenarios.

Key words: privacy-preserving; object detection; malicious adversary; secure multi-party computation

随着云计算和多方协同计算技术的快速发展, 图像处理正逐渐从传统的本地处理模式向云端和多方协同等新型计算环境迁移, 用户需要将明文图像数据上传至云服务器进行处理, 如亚马逊的图像识别服务 (AWS recognition)、谷歌视觉智能平台 (Google cloud vision API) 以及医疗图像诊断系统等. 这种方式极易泄露图像内容, 存在敏感信息泄露的风险, 且在面对篡改等恶意敌手攻击时难以保证数据的完整性和服务的可靠性.

作为一项图像处理中的核心技术, 目标检测已成为众多关键应用的基础, 如视频内容分析[1−4]、自动驾驶[5−7]、工业检查[8,9]以及医疗辅助[10]等.

现有的方法[11−13]要求用户直接将原始图像上传至云服务器, 然后在服务器上对明文图像数据进行目标检测. 这种集中式处理方式存在诸多问题: 服务提供商可能会窃取敏感信息、恶意篡改数据, 或者在未经授权的情况下直接收集数据, 从而引发严重的隐私问题. 大部分的隐私保护目标检测方案[14,15]基于半诚实模型, 即假设服务器可能会试图窃取数据, 但仍然会严格遵循预定的协议并产生正确的结果. 然而, 这种模型无法抵御恶意敌手的攻击. 在更符合实际情况的恶意敌手模型中, 攻击者会主动偏离协议, 窃取或破坏训练好的模型, 并篡改检测结果, 直接危及数据的完整性和服务的可靠性. 如何在医疗影像和智能监控等敏感场景中, 在不牺牲目标检测效率和准确性的前提下有效抵御恶意攻击, 已成为一个亟待解决的技术难题.

目前的隐私保护技术主要有同态加密 (homomorphic encryption, HE)、差分隐私 (differential privacy, DP) 和安全多方计算 (secure multi-party computation, SMPC). 同态加密[16−18]允许对加密数据进行计算, 但其计算复杂度高且延迟大, 限制了其在现实系统应用中的部署. 差分隐私[19−21]通过引入噪声来保护数据, 理论上可以防止敏感信息泄露, 但噪声的添加往往会降低检测精度, 对目标检测性能产生较大的负面影响. 安全多方计算使多方能够在不泄露各自私有数据的情况下协作计算得出结果, 这使其成为在保护隐私的同时完成复杂数据处理任务的理想方法. 然而, 目前多数基于 SMPC 的机器学习方案[14,22−25]仅面向半诚实安全模型进行设计, 未能有效抵御恶意敌手的攻击威胁. 虽然已有工作[26,27]探索了恶意模型下的方案, 但其为实现更高安全性所引入的巨大计算复杂度和通信开销, 严重阻碍了实际部署. 更为严峻的是, 目标检测网络通常比标准的卷积神经网络 (convolutional neural network, CNN) 层次更深、算子更多样、结构更复杂. 这种复杂性导致在恶意敌手环境下应用 SMPC 时, 不同层次间的安全计算协议极可能出现兼容性问题或功能失配. 因此, 在恶意敌手环境下设计隐私保护目标检测模型 (确保数据保密性和计算正确性, 同时满足实时效率要求), 仍然是一个关键挑战.

针对上述挑战, 本文提出了一种基于安全多方计算的恶意敌手环境下的安全目标检测方案——MalOD. 该方案利用复制秘密共享 (replicated secret sharing, RSS) 技术和联合消息传递技术, 在恶意敌手存在的情况下保证数据保密性和检测精度. MalOD 在计算和通信开销方面进行了优化, 为在严格安全要求下实现隐私保护目标检测提供了理论基础和实际可行性.

本文的主要贡献总结如下.

● 提出一种面向恶意敌手环境的目标检测推理方案——MalOD, 实现了密文图像的多级特征提取, 并基于此完成了安全目标检测. 作为首个专门针对恶意敌手环境设计的安全目标检测方案, MalOD 引入了安全特征金字塔网络 (secure feature pyramid network, SecFPN), 并设计了安全区域提议网络 (secure region proposal network, SecRPN) 和兴趣区域安全对齐 (secure region of interest align, SecRoIA) 模块. 通过将这些安全计算模块相结合,

MalOD 在恶意敌手环境下有效地保护了目标检测各个环节数据的隐私和完整性.

● 针对恶意敌手环境, 基于 RSS 技术设计了一套安全计算原语, 包括安全向上取整 (secure ceil, *SCeil*)、安全双线性插值 (secure bilinear interpolation, *SBI*) 和安全最近邻插值 (secure nearest neighbor interpolation, *SNNI*) 函数, 以适应恶意敌手环境的需求. 这些原语能够在深度学习中实现高效且安全的多级特征提取, 实现了推理效率和安全性之间的平衡.

● 证明了 MalOD 的正确性和安全性, 并在 COCO 和 Pascal VOC 数据集上对其进行系统的功能评估和性能评估. 实验结果表明, 在确保安全性以应对恶意敌手的情况下, MalOD 性能损失可以忽略不计, 能够满足目标检测不同任务场景下的安全与性能要求.

本文第 1 节介绍隐私保护目标检测研究近期进展. 第 2 节介绍基础知识. MalOD 的系统架构、安全计算原语设计以及隐私保护目标检测模型的详细设计分别在第 3–5 节介绍. 最后, 在第 6 节展示实验评估结果, 在第 7 节进行总结. 此外, 附录 A 详细阐述了 MalOD 的正确性和安全性分析, 遵循模块化证明思路, 在文献 [28,29] 基础协议的安全性与正确性的前提下, 本文重点对新增的安全计算原语以及 MalOD 整体系统的正确性与安全性进行严格证明.

1 隐私保护目标检测相关工作

在目标检测领域, 算法的精度和效率一直是研究的核心问题. 近年来, 随着隐私保护需求的日益增长, 隐私保护目标检测逐渐成为研究热点. 本节将首先回顾传统目标检测算法的发展历程, 然后重点探讨隐私保护目标检测领域的现有方法及其不足.

● 目标检测算法. 现有的目标检测算法主要分为两阶段检测算法和单阶段检测算法两大类. 两阶段检测算法首先生成候选区域, 然后对这些区域进行分类和边界框回归, 以实现高精度检测. Fast R-CNN[30]通过引入感兴趣区域 (region of interest, RoI) 层, 统一了候选区域的特征表示, 从而提高了检测速度. Faster R-CNN[31]进一步引入区域提议网络 (region proposal network, RPN) 实现了端到端的训练和检测, 显著提升了检测效率. 此外, 特征金字塔网络 (feature pyramid network, FPN)[32]通过整合不同尺度的特征, 增强了多尺度目标的检测能力. 与从图像直接预测目标位置和类别的单阶段检测器[11−13]相比, 两阶段检测算法尽管检测速度相对较慢, 但通常能够实现更高的精度和召回率, 特别适用于对检测精度要求较高的应用场景, 如医学图像检测[33].

● 隐私保护目标检测. 当前, 目标检测算法的研究主要集中在针对明文图像的处理上, 针对隐私保护的研究相对较少. 且现有的大多数隐私保护目标检测方法[14,15]仍然基于半诚实模型 (即参与方会遵循协议, 但可能会尝试从协议中获取额外信息), 未能有效应对恶意攻击的威胁. 文献 [14] 面向半诚实敌手提出了基于两方安全计算的隐私保护 Faster R-CNN 框架, 应用于医学图像目标检测. 而文献 [15] 则在三方框架中结合秘密共享和混淆电路, 扩展了安全计算协议以实现目标检测. 这些方法在一定程度上提升了计算过程的隐私保护, 但仍无法有效抵御恶意敌手. 与目标检测任务相比, 隐私保护 CNN 的研究已取得显著进展. 文献 [34] 通过量化技术将神经网络训练任务置于 SMPC 场景中, 实现了对神经网络的安全训练. 文献 [35] 提出了一种三方计算协议, 优化了广域网高延迟环境下分布式推理性能. 文献 [36] 则针对隐私保护机器学习中的多方计算, 提出了一种面向诚实多数场景的可扩展推理框架. 这些成果为在保护隐私的前提下执行 CNN 的核心计算任务 (如训练、推理) 提供了坚实的技术基础, 证明了其可行性. 然而, 将这些成熟的隐私保护 CNN 技术迁移并适配到目标检测网络, 仍面临显著挑战: 目标检测结构更复杂, 涉及多分支、多阶段模块, 算子种类多, 计算流程差异大. 尤其在恶意敌手模型下, 如何同时保证安全性、效率与实用性, 仍是亟待攻克的关键难题.

2 基础知识

本节对目标检测 Faster R-CNN 的网络结构进行介绍, 并说明本研究中使用的基本符号和基础计算协议.

2.1 Faster R-CNN

Faster R-CNN 由特征提取网络、RPN 以及分类与回归模块这 3 个模块组成.

● 特征提取网络. Faster R-CNN 利用 CNN 提取语义和空间特征. 为了更高效地进行多层次特征提取, 本文采用 ResNet50 与 FPN 相结合的主干网络. 其中, ResNet50 通过残差连接有效缓解了梯度消失问题, 而 FPN 通过融合浅层和深层特征图, 进一步提升检测性能.

● 区域提议网络. RPN 通过滑动窗口机制生成锚框, 然后利用边界框回归对锚框进行优化, 生成候选区域. 随后通过非极大值抑制 (non-maximum suppression, NMS) 筛选出高质量的区域提议.

● 分类与回归. 基于 RPN 输出的候选区域, Faster R-CNN 通过 RoI 算法提取候选区域对应的固定大小的特征, 然后执行分类和边界框回归, 最终完成目标检测任务.

2.2 基础符号表示

● 复制秘密共享. 本文采用 (4, 3) 复制秘密共享方案. 将需要共享的秘密值 x 分成 4 个随机加法秘密份额 $x^{(0)}+x^{(1)}+x^{(2)}+x^{(3)}=x$, 并分发至 4 台服务器 S_0、S_1、S_2、S_3. 对于一个秘密值 $x\in\mathbb{Z}_{2^k}$, 符号 $[x]^m$ 表示服务器在模 m 下所持有的秘密份额集合, 其中 S_i 持有 $[x]_i^m=\{x^{(j)}\}_{j\neq i}$, 任意两台服务器可以重构 x. 通常情况下有 $m=2^k$, 在没有特殊情况时, 将省略该符号. 本文采用 SMPC 中通用的定点数编码方式, 将浮点数映射为整数. 具体而言, 对于浮点数 $v\in\mathbb{R}$, 给定小数精度 f, 其在环 $\mathbb{Z}_{2^k}$ 上的整数表示为 $x=\lfloor v\cdot 2^f\rfloor \bmod 2^k$. 文中所有密文域计算均基于此定点编码进行. 本文所涉及的其他主要符号说明见表 1.

表 1 主要符号说明

符号	描述	备注
$\mathbf{F}$	特征图	加粗大写字母表示整张特征图
$\mathbf{F}(x,y)$	特征图在坐标 (x,y) 处的值	—
H,W	特征图的高与宽	—
a	锚框	中心坐标 $(x_{c,(a)},y_{c,(a)})$, 尺寸 $(w_{(a)},h_{(a)})$
$\mathcal{A}$	锚框集合	下标用于区分不同尺寸锚框
b	目标边界框	中心坐标 $(x_{c,(b)},y_{c,(b)})$, 尺寸 $(w_{(b)},h_{(b)})$
$\mathcal{B}$	边界框集合	与 $\mathcal{A}$ 相对应
$Prob$	置信度集合	—
δ	偏移量 (锚框与边界框之间的偏移)	—

● 数据重构和输入共享. 文献 [28] 中的共享输入协议通过引入冗余份额, 使得即使部分数据被恶意篡改, 仍能通过其他冗余份额恢复原始数据. 同时, 该文献提出的联合消息传递协议 (joint messaging protocol, JMP) 采用交叉验证机制, 对不同来源的数据进行相互验证, 从而及时发现恶意敌手. 本文利用 JMP 和共享输入协议, 在恶意敌手环境下实现高效的数据重构和输入共享, 满足中止安全性.

2.3 基础运算协议

本节的基础运算协议基于安全四方计算[28], 采用了 MP-SPDZ[36]框架 (https://github.com/data61/MP-SPDZ). 本研究利用 JMP 协议, 实现服务器之间的安全交互, 确保了在恶意敌手环境下的安全性.

● 安全加法协议 (secure addition protocol, Π_{SAdd}): 对于来自服务器 S_i 的输入 $([x]_i,[y]_i)$, Π_{SAdd} 输出 $[x+y]_i$, 记作 $[x]+[y]=[x+y]$, 其中 $i\in\{0,1,2,3\}$.

● 安全乘法协议 (secure multiplication protocol, Π_{SMult}): 对于各个服务器 S_i 持有的输入 $([x]_i,[y]_i)$, Π_{SMult} 输出 $[x\times y]_i$, 记作 $[x]\times[y]=[x\times y]$, 其中 $i\in\{0,1,2,3\}$.

● 安全比较协议 (secure comparison protocol, Π_{SComp}): 对于各个服务器 S_i 持有的输入 $([x]_i,[y]_i)$, 输出 $[z]_i=\{z_j\}_{j\neq i}$, 其中 $i,j\in\{0,1,2,3\}$. 若 $x>y$, 则 $z^{(0)}+z^{(1)}+z^{(2)}+z^{(3)}=1$; 否则, $z^{(0)}+z^{(1)}+z^{(2)}+z^{(3)}=0$.

● 安全选择协议 (secure selection protocol, Π_{SSelect}): 给定来自各个服务器 S_i 的输入 $([x]_i,[y]_i,[c]_i)$, 其中 $c\in\mathbb{Z}_2$, Π_{SSelect} 输出 $[z]_i=\{z^{(j)}\}_{j\neq i}$. 若 $c=0$, $z^{(0)}+z^{(1)}+z^{(2)}+z^{(3)}=x$; 否则, $z^{(0)}+z^{(1)}+z^{(2)}+z^{(3)}=y$. 结合 Π_{SSelect} 以及 Π_{SComp}, 可得到安全最大值函数 (secure maximum function, Π_{SMax}).

● 安全排序协议 (secure sorting protocol, Π_{SSort}): 设有序数组 $X=\{x_0,x_1,\ldots,x_n\}$ 和 $Y=\{y_0,y_1,\ldots,y_n\}$, 那么 $[X]=\{[x_0],[x_1],\ldots,[x_n]\}$、$[Y]=\{[y_0],[y_1],\ldots,[y_n]\}$. 各 S_i 输入 $([X]_i,[Y]_i)$, Π_{SSort} 向 S_i 输出 $[Z]_i$, 其中 Z 是按照 X 的升序对 Y 进行排列的结果.

● 安全深度学习层: 本文采用的基础的安全深度学习层, 包括安全卷积层 (secure convolutional layer, Π_{SConv})、安全最大池化层 (secure max pooling layer, Π_{SMaxpool})、安全批量归一化层 (secure batch normalization layer, Π_{SBN}) 以及安全线性层 (secure linear layer, Π_{SLinear}).

● 安全非线性函数: 主要有自然指数函数、*ReLU* 函数、*Softmax* 函数和 *Sigmoid* 函数.

$$ReLU(x)=\max(0,x),\ Softmax(x_i)=\frac{e^{x_i}}{\sum_{j=1}^{n}e^{x_j}},\ Sigmoid(x)=\frac{1}{1+e^{-x}} \tag{1}$$

对于输入 $[x]$, 利用安全指数函数 (secure exponential function, Π_{SNE})[37]、安全 *ReLU* (secure *ReLU*, Π_{SReLU})、安全 *Softmax* (secure *Softmax*, Π_{SSoftmax}) 以及安全 *Sigmoid* (secure *Sigmoid*, Π_{SSigmoid}) 进行隐私计算. 具体而言, Π_{SNE} 先将 e^x 转换为 $2^{(x\cdot\log_2 e)}$, 再将指数分解为整数与小数部分, 分别通过安全移位与多项式近似实现; Π_{SReLU} 采用比较协议判断符号位并在 SMPC 中实现安全乘法; Π_{SSoftmax} 和 Π_{SSigmoid} 均基于 Π_{SNE} 近似实现.

● 安全向下取整函数 (secure floor function, Π_{SFloor}): 对于各个服务器 S_i 的输入 $[x]_i$, Π_{SFloor} 输出 $[z]_i$, 满足 $z^{(0)}+z^{(1)}+z^{(2)}+z^{(3)}=\lfloor x\rfloor$.

3 MalOD 系统架构

本节从系统模型、威胁模型和设计目标这 3 个方面对 MalOD 的系统架构进行介绍.

3.1 系统模型

MalOD 的系统模型如图 1 所示, 涉及的 4 个实体包括客户端、服务器、模型提供方和可信第三方.

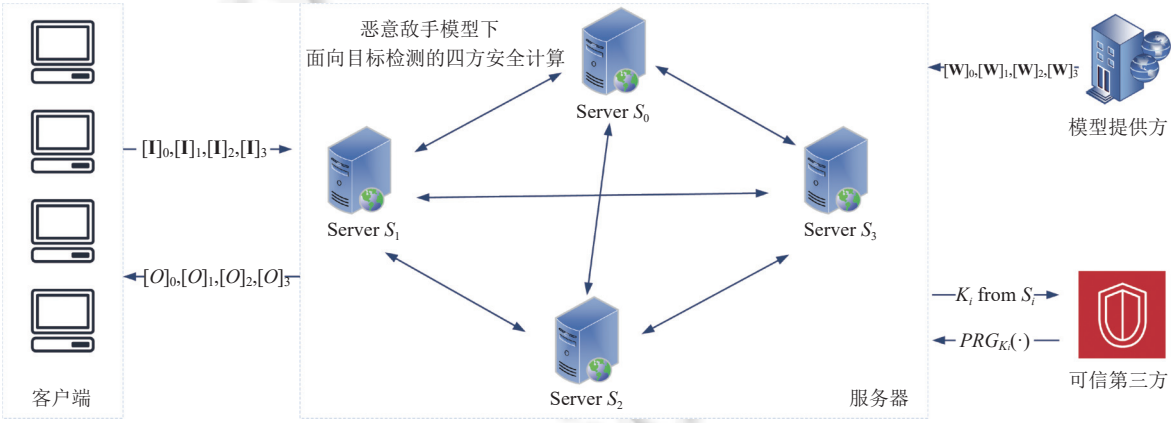

图 1 MalOD 系统模型

● 可信第三方 (trusted third party, $\mathcal{T}$) 为 MalOD 的服务器提供伪随机数生成器 $PRG(\cdot)$. 根据服务器提供的预共享的密钥 K,$\mathcal{T}$ 通过 $PRG_K(\cdot)$ 生成关联随机数, 以保证计算的安全性.

● 模型提供方 (model provider, $\mathcal{M}$) 提供模型权重 $\mathbf{W}$, 并将其分割为 4 个随机份额 $\mathbf{W}=\sum_{i=0}^{3}\mathbf{W}^{(i)}$, 然后将其分发给云服务器, 服务器 S_i 持有 $[\mathbf{W}]_i=\{\mathbf{W}^{(j)}\}_{j\neq i}$.

● 客户端 (client, $\mathcal{C}$) 按照 (4, 3) 的 RSS 方案, $\mathcal{C}$ 将其输入的图像 $\mathbf{I}$ 分割为 4 个随机份额 $\mathbf{I}=\sum_{i=0}^{3}\mathbf{I}^{(i)}$. 每个份额被发送至对应的云服务器, 服务器 S_i 持有 $[\mathbf{I}]_i=\{\mathbf{I}^{(j)}\}_{j\neq i}$.

● 云服务器 (S_0,S_1,S_2,S_3) 根据 $[\mathbf{I}]$ 和 $[\mathbf{W}]$ 进行 MalOD 的隐私保护目标检测安全计算. 计算完成后, 服务器分

别将输出 $[O]_0$, $[O]_1$, $[O]_2$, $[O]_3$ 发送回 $\mathcal{C}$, $\mathcal{C}$ 通过 $O=O^{(0)}+O^{(1)}+O^{(2)}+O^{(3)}$ 对输出进行重构获取检测结果.

MalOD 系统整体可分为离线设置与在线安全推理两个阶段. 在离线设置阶段, 系统首先完成协议初始化及预计算, 服务器集群与可信第三方交互以建立安全参数 (如公私钥及随机数种子等); 随后, 模型提供方将预训练权重映射为秘密份额并分发至各计算节点, 完成模型的隐私保护部署. 在在线安全推理阶段, 客户端生成待检测图像的秘密份额并发送至服务器, 4 台服务器在接收到数据后协同执行隐私保护目标检测协议. 最终, 加密的检测结果返回至客户端进行重构. 在实际部署中, 可信第三方通常由政府监管机构、公证处或具有公信力的非营利组织担任, 仅负责系统初始化而不参与实时计算; 服务器节点可部署在不同的信任域或跨多云环境中, 以物理隔离保障非合谋假设; 模型提供方则通过份额化权重, 实现了模型即服务 (MaaS) 的安全范式.

3.2 威胁模型

MalOD 系统采用诚实多数的恶意敌手模型, 假设 4 台服务器中至多存在一个半诚实或恶意节点, 且节点间无合谋行为. 通过冗余参与方设计的交叉验证机制 (详见第 2.2 节), 系统可实现对于恶意敌手的检测, 并在该安全假设下实现可中止安全性. 与文献 [14,15] 的半诚实模型 (假设服务器严格遵循协议) 不同, 本文通过增加冗余参与方, 考虑了更强的威胁, 即参与方可能通过伪造消息、篡改中间结果或偏离协议逻辑来攻击系统, 从而导致隐私泄露或结果失真. 此外, 本文假设客户端、模型提供者和可信第三方均为诚实实体, 且各实体之间通过安全信道完成通信, 从而避免通信链路攻击. 在实际部署中, 4 台服务器可由不同信任域的独立组织或第三方机构托管, 如可分别托管于不同地区的 Google Cloud 平台中 [38], 从组织和硬件层面降低合谋概率, 从而满足诚实多数的恶意敌手模型假设.

3.3 设计目标

MalOD 的设计目标是在保护图像和模型数据安全的同时, 高效地实现目标检测. 具体而言, 其目标包括: 正确性, 即 MalOD 的目标检测结果应与非隐私保护下的目标检测结果一致; 安全性, 即在整个 MalOD 的数据交互和计算过程中, 各服务器仅能获取自身持有的秘密份额, 无法推断出中间结果, 从而保护客户端图像、模型权重和检测结果的隐私, 防止数据泄露和篡改; 高效率, 即在确保正确性和安全性的前提下, 尽量降低协议的计算开销和通信开销.

4 安全计算原语设计

本节设计了若干新的安全计算原语, 具体包括安全向上取整函数、安全双线性插值函数、安全最近邻插值函数以及安全边界框解码操作 (secure bounding box decoding, *SBBDec*), 这些原语将应用于第 5 节中.

4.1 安全向上取整函数

向上取整函数用于计算不小于其输入 x 的最小整数, 即 $\lceil x\rceil=\min\{n\in Z\mid n\geqslant x\}$. 相反, 向下取整函数表示为 $\lfloor x\rfloor=\max\{n\in Z\mid n\leqslant x\}$. 考虑到 $\lceil x\rceil=-\lfloor -x\rfloor$, *SCeil* 可以基于 Π_{SFloor} 进行构建.

在 *SCeil* 中, 每个服务器会在 Π_{SFloor} 计算的前后各执行一次本地的符号取反操作. 整个过程中, 除一次 Π_{SFloor} 外不需要额外的通信, 如公式 (2) 所示. *SCeil* 算法如算法 1 所示.

$$\lceil x\rceil=\lceil x^{(0)}+x^{(1)}+x^{(2)}+x^{(3)}\rceil=-\lfloor -(x^{(0)}+x^{(1)}+x^{(2)}+x^{(3)})\rfloor=-\lfloor -x^{(0)}-x^{(1)}-x^{(2)}-x^{(3)}\rfloor \tag{2}$$

算法 1. 安全向上取整函数: *SCeil* 算法.

输入: S_0 持有 $[x]_0$, S_1 持有 $[x]_1$, S_2 持有 $[x]_2$, S_3 持有 $[x]_3$;
输出: S_0 获取 $[z]_0$, S_1 获取 $[z]_1$, S_2 获取 $[z]_2$, S_3 获取 $[z]_3$.

1. S_0、S_1、S_2 和 S_3 在本地计算 $[-x]_0=-1\times[x]_0$, $[-x]_1=-1\times[x]_1$, $[-x]_2=-1\times[x]_2$ 和 $[-x]_3=-1\times[x]_3$
2. $[y]_0,[y]_1,[y]_2,[y]_3\leftarrow\Pi_{\mathrm{SFloor}}([-x]_0,[-x]_1,[-x]_2,[-x]_3)$, 其中 $y=\lfloor -x\rfloor$
3. $[z]_0=-1\times[y]_0$, $[z]_1=-1\times[y]_1$, $[z]_2=-1\times[y]_2$, $[z]_3=-1\times[y]_3$, 即 $z=-y$

4.2 安全最近邻插值算法

SNNI 算法通过将目标特征图中每个像素点的特征值设置为与其最近的源密文特征图像素点的值, 实现了特征图的放缩, 为采样提供了一种简单快速的近似方法. 设尺寸为 $(dstH, dstW)$ 的目标特征图中的像素点的坐标为 (x', y'); 其在大小为 $(srcH, srcW)$ 的源密文特征图中, 对应的采样点坐标记作 (x, y). 则其对应关系如下:

$$x = x' \times (srcW / dstW),\ y = y' \times (srcH / dstH) \tag{3}$$

在 *SNNI* 算法中, 每个服务器输入源密文特征图的秘密份额, 即 $[\mathbf{F}]_0$、$[\mathbf{F}]_1$、$[\mathbf{F}]_2$ 和 $[\mathbf{F}]_3$. 由于目标特征图的尺寸 $(dstH, dstW)$ 是公开的, 所有服务器均可根据公式 (3) 独立计算采样点的映射索引, 并输出放缩后的密文特征图, 其尺寸为 $(dstH, dstW)$. *SNNI* 算法如算法 2 所示. 该算法本质是对本地秘密份额的重排与复制, 不涉及任何需多方交互的计算 (如安全乘法或比较). 因此, *SNNI* 属于完全的本地操作, 其通信开销为 0.

算法 2. 安全最近邻插值: *SNNI* 算法.

输入: S_0 持有 $[\mathbf{F}]_0$, S_1 持有 $[\mathbf{F}]_1$, S_2 持有 $[\mathbf{F}]_2$, S_3 持有 $[\mathbf{F}]_3$, 目标特征图大小 $(dstH, dstW)$;
输出: S_0 获取 $[\mathbf{F}']_0$, S_1 获取 $[\mathbf{F}']_1$, S_2 获取 $[\mathbf{F}']_2$, S_3 获取 $[\mathbf{F}']_3$.

1. 获取 $[\mathbf{F}]$ 的尺寸 $(srcH, srcW)$
2. $r_x = srcW / dstW$, $r_y = srcH / dstH$
3. **for** $x' = 1$ to $dstW$ **do**
4. **for** $y' = 1$ to $dstH$ **do**
5. $x = x' \times r_x$, $y = y' \times r_y$
6. S_0、S_1、S_2、S_3分别计算 $[\mathbf{F}']$: $\mathbf{F}'^{(0)}(x', y') = \mathbf{F}^{(0)}(x, y), \mathbf{F}'^{(1)}(x', y') = \mathbf{F}^{(1)}(x, y), \mathbf{F}'^{(2)}(x', y') = \mathbf{F}^{(2)}(x, y), \mathbf{F}'^{(3)}(x', y') = \mathbf{F}^{(3)}(x, y)$
7. **end for**
8. **end for**

4.3 安全双线性插值

双线性插值算法通过计算目标点周围 4 个像素的加权平均值来放缩图像, 从而确保平滑过渡. 设目标点坐标为 (x, y), 给定其 4 个相邻点的像素值 $f(x_1, y_1)$、$f(x_2, y_1)$、$f(x_1, y_2)$ 和 $f(x_2, y_2)$, 其中 $x_1 \leqslant x \leqslant x_2$ 且 $y_1 \leqslant y \leqslant y_2$, 计算公式如下:

$$f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} f(x_1, y_1) + \frac{x - x_1}{x_2 - x_1} f(x_2, y_1) \tag{4}$$

$$f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} f(x_1, y_2) + \frac{x - x_1}{x_2 - x_1} f(x_2, y_2) \tag{5}$$

$$f(x, y) = \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \tag{6}$$

在 *SBI* 算法中, 各个服务器 S_i 输入特征图 $\mathbf{F}$ 和目标点坐标 (x, y) 的秘密份额 $([\mathbf{F}]_i, [x]_i, [y]_i)$, 调用两次 Π_{SFloor} 和两次 *SCeil* 来计算相邻像素坐标值的秘密份额; 同时, 为了索引特征图, 必须对坐标进行重构从而引入了公开通信. 随后, 根据公式 (4)–(6), 通过 Π_{SMult} 计算 $[f(x, y)]$, 得到 (x, y) 处插值计算结果的输出份额. 特别地, 公式涉及定点数密文乘法, 每执行一次 Π_{SMult} 均需进行通信以完成数值截断. 当采样频率为 r 时, *SBI* 算法计算 $r \times r$ 个采样点的平均像素值作为目标像素点的插值计算结果. 采样率为 r 的 *SBI* 算法详见算法 3.

算法 3. 安全双线性插值: *SBI* 算法.

输入: S_0 持有 $([\mathbf{F}]_0, [x]_0, [y]_0)$, S_1 持有 $([\mathbf{F}]_1, [x]_1, [y]_1)$, S_2 持有 $([\mathbf{F}]_2, [x]_2, [y]_2)$, S_3 持有 $([\mathbf{F}]_3, [x]_3, [y]_3)$, 采样频率 r;
输出: S_0 获取 $[f]_0$, S_1 获取 $[f]_1$, S_2 获取 $[f]_2$, S_3 获取 $[f]_3$.

1. 初始化: $f^{(0)} = 0$, $f^{(1)} = 0$, $f^{(2)} = 0$, $f^{(3)} = 0$, $total\_r = r \times r$

2. **for** $i=0$ to $r-1$ **do**
3. **for** $j=0$ to $r-1$ **do**
4. $[x']=[x]+(i-r/2)/r$, $[y']=[y]+(j-r/2)/r$
5. $[x_1]\leftarrow \Pi_{\mathrm{SFloor}}([x']),[y_1]\leftarrow \Pi_{\mathrm{SFloor}}([y']),[x_2]\leftarrow SCeil([x']),[y_2]\leftarrow SCeil([y'])$
6. S_0、S_1、S_2 和 S_3 通过 JMP 收到 x_1、x_2、y_1、y_2
7. $([k_1]_0,[k_1]_1,[k_1]_2,[k_1]_3)\leftarrow \dfrac{[x_2-x']\times[y_2-y']}{(x_2-x_1)(y_2-y_1)},([k_2]_0,[k_2]_1,[k_2]_2,[k_2]_3)\leftarrow \dfrac{[x'-x_1]\times[y_2-y']}{(x_2-x_1)(y_2-y_1)}$,
$([k_3]_0,[k_3]_1,[k_3]_2,[k_3]_3)\leftarrow \dfrac{[x_2-x']\times[y'-y_1]}{(x_2-x_1)(y_2-y_1)},([k_4]_0,[k_4]_1,[k_4]_2,[k_4]_3)\leftarrow \dfrac{[x'-x_1]\times[y'-y_2]}{(x_2-x_1)(y_2-y_1)}$
8. S_0、S_1、S_2、S_3 本地查询 $[f_1]=[\mathbf{F}(x_1,y_1)]$, $[f_2]=[\mathbf{F}(x_2,y_1)]$, $[f_3]=[\mathbf{F}(x_1,x_2)]$, $[f_4]=[\mathbf{F}(x_2,y_2)]$
9. $[f]_0,[f]_1,[f]_2,[f]_3\leftarrow([k_1]\times[f_1]+[k_2]\times[f_2]+[k_3]\times[f_3]+[k_4]\times[f_4])/total\_r+[f]$
10. **end for**
11. **end for**

4.4 安全边界框解码操作

在边界框预测中, *SBBDec* 操作通过将锚框偏移量解码为实际的边界框坐标, 从而确定目标在图像中的具体位置. 每个服务器 S_i 分别持有锚框 a 和偏移量 δ 的秘密份额 $([a]_i,[\delta]_i)$, 其中 $[a]_i=\{a^{(j)}\}_{j\neq i}$, $[\delta]_i=\{\delta^{(j)}\}_{j\neq i}$, 且 $a=\sum_{i=0}^{3}a^{(i)}$, $\delta=\sum_{i=0}^{3}\delta^{(i)}$. 解码权重 (wx,wy,ww,wh) 是预先定义的明文常数, 用于将偏移量转换为实际坐标.

$$x_{c,(b)}=\frac{\delta[0]}{wx}\times w_{(a)}+x_{c,(a)}\Rightarrow\sum_{i=0}^{3}x_{c,(b)}^{(i)}=\frac{\sum_{i=0}^{3}\delta[0]^{(i)}}{wx}\times\sum_{i=0}^{3}w_{(a)}^{(i)}+\sum_{i=0}^{3}x_{c,(a)}^{(i)} \tag{7}$$

$$y_{c,(b)}=\frac{\delta[1]}{wy}\times h_{(a)}+y_{c,(a)}\Rightarrow\sum_{i=0}^{3}y_{c,(b)}^{(i)}=\frac{\sum_{i=0}^{3}\delta[1]^{(i)}}{wy}\times\sum_{i=0}^{3}h_{(a)}^{(i)}+\sum_{i=0}^{3}y_{c,(a)}^{(i)} \tag{8}$$

$$w_{(b)}=\mathrm{e}^{\frac{\delta[2]}{ww}}\times w_{(a)}\Rightarrow\sum_{i=0}^{3}w_{(b)}^{(i)}=\mathrm{e}^{\sum_{i=0}^{3}\frac{\delta[2]^{(i)}}{ww}}\times\sum_{i=0}^{3}w_{(a)}^{(i)} \tag{9}$$

$$h_{(b)}=\mathrm{e}^{\frac{\delta[3]}{wh}}\times h_{(a)}\Rightarrow\sum_{i=0}^{3}h_{(b)}^{(i)}=\mathrm{e}^{\sum_{i=0}^{3}\frac{\delta[3]^{(i)}}{wh}}\times\sum_{i=0}^{3}h_{(a)}^{(i)} \tag{10}$$

4 个服务器根据公式 (7)–(10) 执行解码操作, 计算得到边界框的中心坐标 $([x_{c,(b)}],[y_{c,(b)}])$ 以及尺寸 $([w_{(b)}],[h_{(b)}])$, 其中 $x_{c,(a)}$、$y_{c,(a)}$、$w_{(a)}$ 和 $h_{(a)}$ 分别表示锚框的中心坐标和尺寸参数. 在计算过程中, $([x_{c,(b)}],[y_{c,(b)}])$ 除在本地的线性运算外, 仅需执行两次 Π_{SMult}; $[w_{(b)}]$ 和 $[h_{(b)}]$ 的计算各需要一次 Π_{SNE} 和一次 Π_{SMult}. 最终, *SBBDec* 输出安全解码后的边界框 $[b]_i=\{b^{(j)}\}_{j\neq i}$, 满足 $b=\sum_{i=0}^{3}b^{(i)}$. *SBBDec* 操作的详细算法见算法 4.

算法 4. 安全边界框解码: *SBBDec* 操作.

输入: S_0 持有 $([a]_0,[\delta]_0)$, S_1 持有 $([a]_1,[\delta]_1)$, S_2 持有 $([a]_2,[\delta]_2)$, S_3 持有 $([a]_3,[\delta]_3)$, 解码权重 (wx,wy,ww,wh);
输出: S_0 获取 $[b]_0$, S_1 获取 $[b]_1$, S_2 获取 $[b]_2$, S_3 获取 $[b]_3$.

1. S_0、S_1、S_2 和 S_3 根据 $[a]$ 在本地计算参数 $[x_{c,(a)}]$、$[y_{c,(a)}]$、$[w_{(a)}]$、$[h_{(a)}]$
2. $[x_{c,(b)}]\leftarrow\dfrac{[\delta[0]]}{wx}\times[w_{(a)}]+[x_{c,(a)}]$, $[y_{c,(b)}]\leftarrow\dfrac{[\delta[1]]}{wy}\times[h_{(a)}]+[y_{c,(a)}]$
3. $[w_{(b)}]\leftarrow\Pi_{\mathrm{SNE}}\left(\dfrac{[\delta[2]]}{ww}\right)\times[w_{(a)}]$, $[h_{(b)}]\leftarrow\Pi_{\mathrm{SNE}}\left(\dfrac{[\delta[3]]}{wh}\right)\times[h_{(a)}]$
4. S_0、S_1、S_2 和 S_3 根据参数 $[x_{c,(b)}]$、$[y_{c,(b)}]$、$[w_{(b)}]$、$[h_{(b)}]$ 在本地构建 $[b]$

对于目标检测过程中涉及的关键计算算子, 本节给出了相应的安全实现, 包括安全向上取整、安全双线性插值、安全最近邻插值以及安全边界框解码. 这些安全原语与目标检测模型的各个核心模块紧密耦合: 其中, 安全最近邻插值用于 SecFPN 中的特征上采样以实现多尺度特征融合; 安全双线性插值构成了 SecRoIA 模块的核心, 用于特征对齐; 而安全边界框解码则被应用于 SecRPN 和最终的分类回归网络中, 以完成坐标的恢复. 由此, Faster R-CNN 的推理流程能够在恶意敌手环境下, 通过安全多方计算方式完成执行.

5 隐私保护目标检测模型

本节在前述基础运算协议与安全计算原语的基础上, 构建 MalOD 方案的隐私保护目标检测模型. 该模型基于 Faster R-CNN 的推理结构, 由安全主干网络 (secure backbone, SecBackbone)、安全区域提议网络 (SecRPN)、RoI 安全对齐 (SecRoIA) 和分类模块 (secure classification, SecClassification) 组成. 各模块基于安全多方计算在 4 个服务器之间协同工作, 实现密文图像输入到检测结果输出的端到端安全推理, 其整体结构如图 2 所示, 水平箭头表示密文数据在各处理阶段间的正向流动, 垂直双向箭头表示 4 台服务器在计算过程中进行的 JMP 安全通信与交叉验证.

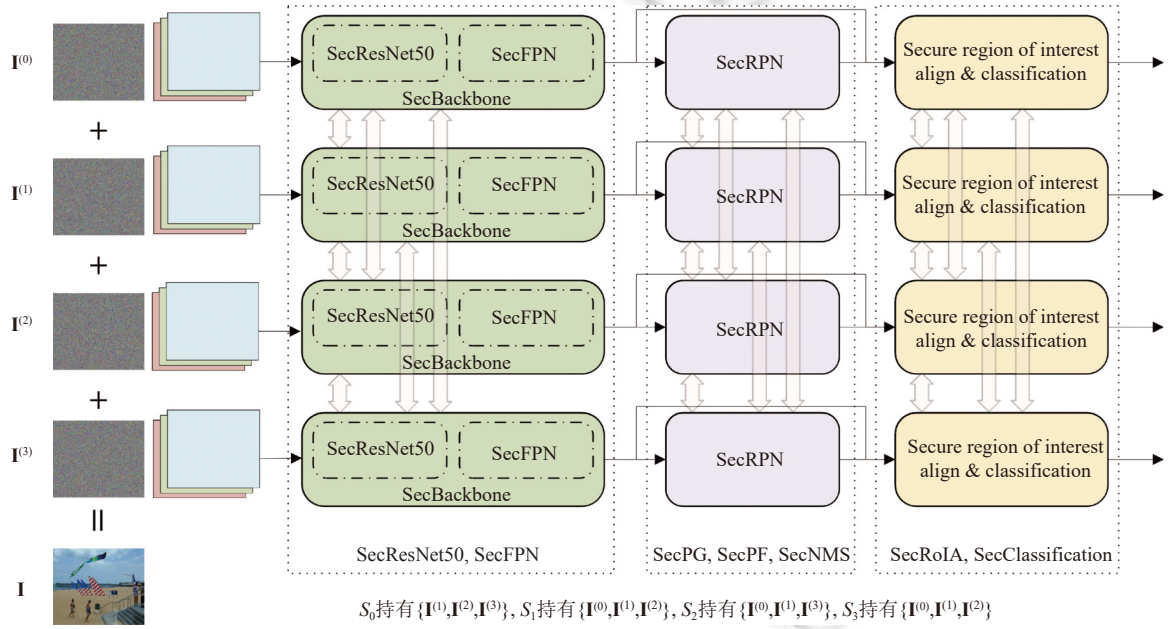

图 2 面向恶意敌手的隐私保护目标检测模型

5.1 安全主干网络

SecBackbone 由安全 ResNet50 (secure ResNet50, SecResNet50) 和 SecFPN 组成, 用于多级安全特征提取. 在 SecResNet50 中, 服务器 S_i 将 $[\mathbf{I}]_i$ 处理为不同尺度的特征图份额 $[\mathcal{C}]=\{[\mathbf{C}_1],[\mathbf{C}_2],[\mathbf{C}_3],[\mathbf{C}_4]\}$, 特征图的分辨率从 $\mathbf{C}_1$ 到 $\mathbf{C}_4$ 逐渐降低.

SecFPN 包含 4 个特征融合层, 分别对应 SecResNet50 的输出 $[\mathbf{C}_1]$、$[\mathbf{C}_2]$、$[\mathbf{C}_3]$ 和 $[\mathbf{C}_4]$, 用于生成融合后的特征图 $[\mathbf{P}_1]$、$[\mathbf{P}_2]$、$[\mathbf{P}_3]$ 和 $[\mathbf{P}_4]$, 如图 3 所示. 其中, $[\mathbf{P}_4]$ 保留了最丰富的语义信息且具有最低分辨率. 此外, 通过 Π_{SMaxpool} 生成特征图 $[\mathbf{P}_5]$. SecFPN 通过 *SNNI* 融合不同层次的细节和语义特征, 使 SecBackbone 能够提取层次化的多尺度特征图集合 $[\mathcal{P}]=\{[\mathbf{P}_1],[\mathbf{P}_2],[\mathbf{P}_3],[\mathbf{P}_4],[\mathbf{P}_5]\}$.

对于 $[\mathbf{P}_N]$ 的计算, 每个服务器输入 $([\mathbf{C}_N],[\mathbf{C}_M])$, 其中 $N=1,2,3$ 且 $M=N+1$, 如公式 (11) 所示, 将安全深度学习层处理后的 $[\mathbf{C}'_N]$ 与 *SNNI* 采样后的 $[\mathbf{C}_M]$ 进行特征融合得到 $[\mathbf{P}_N]$.

$$[\mathbf{P}_N]_0,[\mathbf{P}_N]_1,[\mathbf{P}_N]_2,[\mathbf{P}_N]_3 \leftarrow SNNI([\mathbf{C}_M]_0,[\mathbf{C}_M]_1,[\mathbf{C}_M]_2,[\mathbf{C}_M]_3)+\sum_{i=0}^{3}\mathbf{C}_N'^{(i)} \tag{11}$$

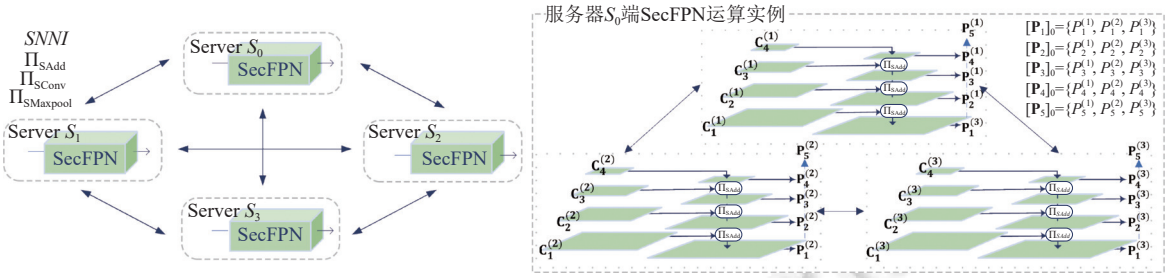

图 3 SecFPN 多层次特征提取

5.2 安全区域提议网络

SecRPN 基于多层次特征图生成提议区域, 主要由 3 个组件构成: 锚框生成器 (anchor box generator, ABG)、安全提议生成器 (secure proposal generator, SecPG) 和安全提议过滤器 (secure proposal filter, SecPF). ABG 负责生成初始的锚框, SecPG 根据特征图和锚框生成提议区域, 而 SecPF 则对生成的提议区域进行筛选和优化.

5.2.1 锚框生成器

ABG 生成了 5 组基础锚框集合, 尺寸分别为 32、64、128、256 和 512, 记作 $A_{(s)}$, 其中 $s\in\{32,64,128,256,512\}$. 每组 $A_{(s)}$ 的锚框包含 3 种长宽比 $AR=\{0.5,1.0,2.0\}$. 每个特征图 $[\mathbf{F}]\in[\mathcal{P}]$ 都被分配一个对应的基础锚框集合, 其中分辨率较低的特征图对应尺寸较大的锚框. 具体来说, 特征图 $[\mathbf{P}_1]$、$[\mathbf{P}_2]$、$[\mathbf{P}_3]$、$[\mathbf{P}_4]$ 和 $[\mathbf{P}_5]$ 分别对应锚框集合 $A_{(32)}$、$A_{(64)}$、$A_{(128)}$、$A_{(256)}$ 和 $A_{(512)}$, 从而实现不同尺寸目标的检测. 每个服务器在特征图 $[\mathbf{F}]$ 上执行本地滑动窗口操作, 生成对应的锚框组 $[A'_{(s)}]=\left\{[a'_{(s),1}],[a'_{(s),2}],\ldots,[a'_{(s),n}]\right\}$, 其中 $n=H\times W\times 3$, (H,W) 为 $[\mathbf{F}]$ 的尺寸. 最终, 得到所有特征图上的锚框集合 $[\mathcal{A}']=\left\{[A'_{(32)}],[A'_{(64)}],[A'_{(128)}],[A'_{(256)}],[A'_{(512)}]\right\}$.

5.2.2 安全提议生成器

SecPG 在密文上预测目标区域边界框和对应的置信度 (每个边界框包含目标对象的概率). 它通过 Π_{SConv} 和 Π_{SReLU} 层增强 SecBackbone 提取的特征图 $[\mathbf{F}]\in[\mathcal{P}]$, 得到 $[\mathbf{F}']$, 然后通过安全分类分支和安全回归分支进行处理. 安全分类分支计算每个锚框的置信度. 安全回归分支预测锚框到真实目标的偏移量. 最终, SecPG 使用 *SBBDec* 解码边界框, 输出所有特征图生成的边界框集合 $[\mathcal{B}]$ 及其置信度 $[Prob]$, 如算法 5 所示. 边界框总数 $N=\sum_{i=1}^{5}H_i\times W_i$, 其中 H_i 和 W_i 分别是特征图 $[P_i]\in[\mathcal{P}]$ 的高和宽.

算法 5. 安全提议生成器 SecPG.

输入: S_0 持有 $([\mathcal{P}]_0,[\mathcal{A}]_0)$, S_1 持有 $([\mathcal{P}]_1,[\mathcal{A}]_1)$, S_2 持有 $([\mathcal{P}]_2,[\mathcal{A}]_2)$, S_3 持有 $([\mathcal{P}]_3,[\mathcal{A}]_3)$;
输出: S_0 获取 $([\mathcal{B}]_0,[Prob]_0)$, S_1 获取 $([\mathcal{B}]_1,[Prob]_1)$, S_2 获取 $([\mathcal{B}]_2,[Prob]_2)$, S_3 获取 $([\mathcal{B}]_3,[Prob]_3)$.

1. 初始化: $\mathcal{B}^{(i)}=\varnothing$ 和 $Prob^{(i)}=\varnothing$ for $i\in\{0,1,2,3\}$
2. **for** $[\mathbf{F}]\in[\mathcal{P}]$ 和与之对应的 $[A]\in[\mathcal{A}]$ **do**
3. 初始化与 $[A]$ 对应的边界框集合 $B^{(i)}=\varnothing$ for $i\in\{0,1,2,3\}$
4. $[\mathbf{F}']_0,[\mathbf{F}']_1,[\mathbf{F}']_2,[\mathbf{F}']_3\leftarrow\Pi_{\text{SReLU}}(\Pi_{\text{SConv}}([\mathbf{F}]_0,[\mathbf{F}]_1,[\mathbf{F}]_2,[\mathbf{F}]_3))$

//安全分类分支: 利用 S_0、S_1、S_2 和 S_3 计算

5. $[Prob]_0,[Prob]_1,[Prob]_2,[Prob]_3\leftarrow\Pi_{\text{SConv}}([\mathbf{F}']_0,[\mathbf{F}']_1,[\mathbf{F}']_2,[\mathbf{F}']_3)$
6. $[Prob]_0,[Prob]_1,[Prob]_2,[Prob]_3\leftarrow\Pi_{\text{SSigmoid}}([Prob]_0,[Prob]_1,[Prob]_2,[Prob]_3)$
7. S_0、S_1、S_2 和 S_3 在本地将 $[Prob]$ 的形状从 $(H,W,3)$ 重塑为 $(H\times W\times 3)$

//安全回归分支: 利用 S_0、S_1、S_2 和 S_3 计算

8. $[\Delta]_0,[\Delta]_1,[\Delta]_2,[\Delta]_3 \leftarrow \Pi_{\text{SConv}}([\mathbf{F}']_0,[\mathbf{F}']_1,[\mathbf{F}']_2,[\mathbf{F}']_3)$
9. S_0、S_1、S_2 和 S_3 在本地将 $[\Delta]$ 的形状从 $(H,W,12)$ 重塑为 $(H\times W\times 3,4)$
10. **for** $[a]\in[A]$ 和与之对应的 $[\delta]\in[\Delta]$ **do**
11. S_0、S_1、S_2 和 S_3 计算 $[b]_0,[b]_1,[b]_2,[b]_3 \leftarrow SBBDec([a],[\delta])$
12. $B^{(0)} \leftarrow B^{(0)}+\{b^{(0)}\}, B^{(1)} \leftarrow B^{(1)}+\{b^{(1)}\}, B^{(2)} \leftarrow B^{(2)}+\{b^{(2)}\}$ and $B^{(3)} \leftarrow B^{(3)}+\{b^{(3)}\}$
13. **end for**
14. $\mathcal{B}^{(0)} \leftarrow \mathcal{B}^{(0)}+\{B^{(0)}\}, \mathcal{B}^{(1)} \leftarrow \mathcal{B}^{(1)}+\{B^{(1)}\}, \mathcal{B}^{(2)} \leftarrow \mathcal{B}^{(2)}+\{B^{(2)}\}$ and $\mathcal{B}^{(3)} \leftarrow \mathcal{B}^{(3)}+\{B^{(3)}\}$
15. **end for**

5.2.3 安全提议过滤器

SecPF 根据边界框及置信度对提议区域进行筛选. 通过 Π_{SComp} 和 Π_{SSelect} 将每个边界框的坐标限制在对应特征图的尺寸范围内, 并移除置信度小于 0 的边界框; 然后, 通过安全非极大值抑制 (secure non-maximum suppression, SecNMS) 算法, 根据边界框交并比 (intersection over union, *IoU*) 保留置信度较高的候选框, 并抑制与之重叠程度超过阈值的其他边界框, 进一步减少冗余提议, 如算法 6 所示.

算法 6. 安全 NMS 算法 SecNMS.

输入: S_0 持有 $([\mathcal{B}]_0,[Prob]_0)$, S_1 持有 $([\mathcal{B}]_1,[Prob]_1)$, S_2 持有 $([\mathcal{B}]_2,[Prob]_2)$, S_3 持有 $([\mathcal{B}]_3,[Prob]_3)$, SecNMS 阈值为 Θ;
输出: S_0 获取 $([\mathcal{B}]_0,[Prob]_0)$, S_1 获取 $([\mathcal{B}]_1,[Prob]_1)$, S_2 获取 $([\mathcal{B}]_2,[Prob]_2)$, S_3 获取 $([\mathcal{B}]_3,[Prob]_3)$.

1. S_0、S_1、S_2 和 S_3 通过 Π_{SMax} 计算 $[\mathcal{B}]$ 中最大的坐标 $max_{\text{coordinate}}$
2. 初始化: $Tag^{(i)}=\varnothing$ for $i\in\{0,1,2,3\}$, $[\mathcal{B}']=[\mathcal{B}]$, $j=0$
3. **for** $[B']\in[\mathcal{B}']$ **do**
4. $offset=(max_{\text{coordinate}}+1)\cdot j$, $j=j+1$
5. **for** $[b']\in[B']$ **do**
6. $Tag \leftarrow Tag+\{1\}$;
7. S_0、S_1、S_2 和 S_3 本地更新 $\{b'^{(0)},b'^{(1)},b'^{(2)},b'^{(3)}\} \leftarrow \{b'^{(0)},b'^{(1)},b'^{(2)},b'^{(3)}\}+offset$
8. **end for**
9. **end for**
10. $[\mathcal{B}] \leftarrow \Pi_{\text{SSort}}([Prob],[\mathcal{B}])$, $[\mathcal{B}'] \leftarrow \Pi_{\text{SSort}}([Prob],[\mathcal{B}'])$, 随后 $[Prob] \leftarrow \Pi_{\text{SSort}}([Prob],[Prob])$
11. S_0、S_1、S_2 和 S_3 通过 JMP 获取 B'
12. **while** $\mathcal{B}'\neq\varnothing$ **do**
13. $b=\mathcal{B}'[0]$, 从 $\mathcal{B}'$ 中删除 b
14. **for** b' in $\mathcal{B}'$ **do**
15. **if** $IoU(b,b')>\Theta$ **then** b' 对应的 $[Tag]$ 设为 0
16. **end if**
17. **end for**
18. **end while**
19. 根据 $[Tag]$ 更新 $([\mathcal{B}]_0,[Prob]_0),([\mathcal{B}]_1,[Prob]_1),([\mathcal{B}]_2,[Prob]_2),([\mathcal{B}]_3,[Prob]_3)$

在 SecNMS 中, 本文通过为不同层次的边界框添加特定的偏移量实现在空间中区分不同大小的候选框, 防止它们相互抑制, 然后进行 *IoU* 计算, 公式如下:

$$IoU(b_i,b_j)=\frac{Area(b_i\cap b_j)}{Area(b_i\cup b_j)} \tag{12}$$

设 SecNMS 的阈值为 Θ, 若 $IoU(b_i,b_j)>\Theta$, 则移除 b_i 和 b_j 中置信度较低的边界框. 但在 SMPC 中, 直接移除边界框在程序上并不可行, 因此, 对于 N 个边界框的 SecNMS 中的 IoU 计算共需进行 $6N(N-1)/2$ 次比较, 而在加密计算上将会带来巨大的计算和通信开销. 为了平衡安全性和算法效率, 在服务器本地进行 IoU 计算. 为确保恶意敌手无法偏离协议进行计算, 系统在批量 IoU 计算前后调用 JMP 协议[28], 实现对恶意行为的检测与防护, 从而保障计算结果的完整性和正确性. 在 IoU 计算过程中, 当敌手无先验知识且无法进行多轮攻击时, 所有参与方仅获得未经筛选的边界框位置信息 (不含图像特征). 这些边界框包含大量噪声和冗余候选目标, 且坐标精度粗糙. 在此威胁模型下, 由于真实有效目标尚未暴露, 难以直接定位敏感目标, 整体信息泄露风险可控.

本文为每个边界框 $[b]\in[\mathcal{B}]$ 设置一个密文标签, 初始值设为“1”, 表示保留边界框. 如果边界框被移除, 则将标签更新为“0”; 否则, 保持不变, 以此来标记删除操作.

5.3 RoI 安全对齐与分类

SecRoIA 算法基于 SecRPN 生成的预测框, 从特征图集合 $[\mathcal{P}]$ 中提取对应区域的特征图. 由于目标边界框的大小各异, 其对应的特征图区域的尺寸也各不相同. 为了确保这些特征图能够被统一输入到神经网络中进行分类预测, 必须先将其调整至相同的尺寸. 为此, SecRoIA 采用 SBI 算法对特征图进行下采样, 将其标准化为 7×7 的固定尺寸. 特别地, 得益于 SBI 主要依赖算术友好的安全乘法运算, 规避了一些非线性插值算子带来的高昂计算与通信开销, 从而在保证对齐精度的同时, 实现了计算效率与准确性的平衡. SecRoIA 算法的详细流程见算法 7. 随后, SecClassification 算法通过两个 Π_{SLinear} 层和一个 Π_{SReLU} 层将调整后的特征图转换为高维特征向量. 与 SecRPN 类似, SecClassification 算法利用安全分类分支和安全回归分支对该高维向量进行分类和回归预测, 并使用 $SBBDec$ 进行解码, 得到最终的检测结果.

算法 7. RoI 安全对齐算法 SecRoIA.

输入: S_0 持有 $([\mathcal{P}]_0,[\mathcal{B}]_0)$, S_1 持有 $([\mathcal{P}]_1,[\mathcal{B}]_1)$, S_2 持有 $([\mathcal{P}]_2,[\mathcal{B}]_2)$, S_3 持有 $([\mathcal{P}]_3,[\mathcal{B}]_3)$, 采样频率 r, 输出特征图尺寸 (h,w);
输出: S_0 获取 $[\mathcal{F}]_0$, S_1 获取 $[\mathcal{F}]_1$, S_2 获取 $[\mathcal{F}]_2$, S_3 获取 $[\mathcal{F}]_3$.

1. 初始化 RoI 特征图集合 $\mathcal{F}^{(i)}=\varnothing$ for $i\in\{0,1,2,3\}$
2. **for** $[b]\in[\mathcal{B}]$ **do**
3. 初始化尺寸为 h×w 的特征图 $[\mathbf{F}]$
4. S_0、S_1、S_2 和 S_3 将 $[b]$ 放缩到其对应的 $[\mathbf{P}_N]\in[\mathcal{P}]$ 中的区域中, 得到 $[b']$, $b'=(x_1,y_1,x_2,y_2)$
5. S_0、S_1、S_2 和 S_3 计算 $[bin_w]\leftarrow([x_2]-[x_1])/w$ 和 $[bin_h]\leftarrow([y_2]-[y_1])/h$
6. **for** $i=0$ to $h-1$ **do**
7. **for** $j=0$ to $w-1$ **do**
8. $[x_c]\leftarrow[x_1]+(j+0.5)\times[bin_w]$, $[y_c]\leftarrow[y_1]+(i+0.5)\times[bin_h]$
9. $[\mathbf{F}(i,j)]\leftarrow SBI([\mathbf{P}_N],[x_c],[y_c],r)$
10. **end for**
11. **end for**
12. Update $[\mathcal{F}]$: $\mathcal{F}^{(0)}\leftarrow\mathcal{F}^{(0)}+\{\mathbf{F}^{(0)}\}$, $\mathcal{F}^{(1)}\leftarrow\mathcal{F}^{(1)}+\{\mathbf{F}^{(1)}\}$, $\mathcal{F}^{(2)}\leftarrow\mathcal{F}^{(2)}+\{\mathbf{F}^{(2)}\}$, $\mathcal{F}^{(3)}\leftarrow\mathcal{F}^{(3)}+\{\mathbf{F}^{(3)}\}$
13. **end for**

6 实验分析

实验在一台配备 Intel(R) Xeon(R) W5-3425 CPU 和 128 GB 内存的工作站上进行, 操作系统为 WSL Ubuntu 22.04.5 LTS. 为了模拟面向恶意敌手的安全四方计算环境, 本文基于 PyTorch (2.3.1+cu121) 和 MP-SPDZ[29]库构建了实验框架. 在底层算术协议中, 模数参数 m 取 2^{64}, 即所有安全计算均在环 $\mathbb{Z}_{2^{64}}$ 上执行; 为了适配整数环上的运算, 所有浮点数形式的数据 (包括图像像素和模型权重) 均采用定点数表示, 小数位精度设为 $f=16$, 具体细节请参

见第 2 节. 在模型设置中, SecNMS 的交并比阈值 Θ 设为 0.5, SecRoIA 的采样参数 r 设为 7. 实验中采用标准的数据预处理策略, 即客户端在本地对原始待检测图像执行归一化操作, 随后将其转换为定点数并生成秘密份额上传至服务器. 模型权重加载自对应训练集 (COCO 2017 和 Pascal VOC 2012) 上的预训练模型, 并在系统初始化阶段转换为定点数份额分发至 4 台服务器.

基于上述设置, 本节首先在 COCO 2017 数据集上进行系统功能评估, 随后扩展至 COCO 2017 和 Pascal VOC 2012 数据集进行综合性能评估.

6.1 功能评估

● 图像秘密共享. 本节在示例图像上展示了图像的复制秘密共享, 如图 4 所示. 图像 $\mathbf{I}^{(0)}$、$\mathbf{I}^{(1)}$、$\mathbf{I}^{(2)}$ 和 $\mathbf{I}^{(3)}$ 表示加性秘密份额, 服务器 S_i 持有 $\{\mathbf{I}^{(j)}\}_{j\neq i}$. 在每个原始图像及其秘密份额的下方, 本文生成了相应图像的 RGB 颜色直方图. 从视觉上看, 仅凭单个服务器所持有的份额, 重建出原始图像是不可行的. 从 RGB 颜色直方图可以看出, 生成的秘密份额图像具有高度随机性, 类似于噪声, 无法从中推断出原始图像的内容.

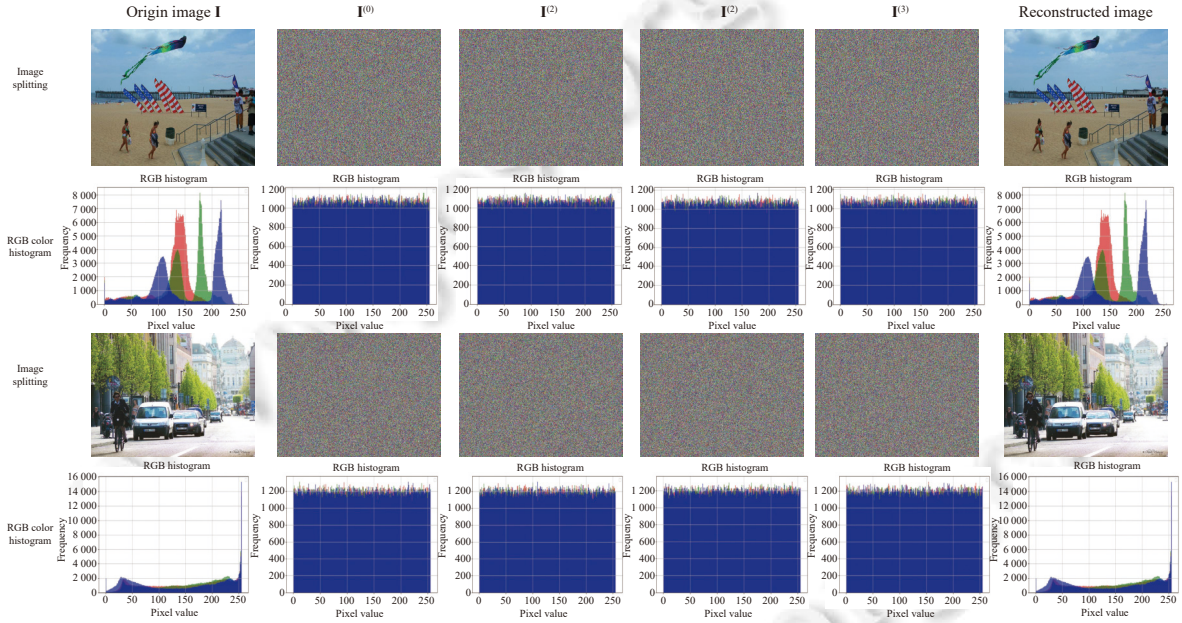

图 4 图像秘密共享

● MalOD 检测结果可视化. 将秘密份额分发给服务器后, 服务器执行 MalOD. 对于目标置信度分数高于 0.5 的检测结果, 进行可视化, 如图 5 所示. 可以发现, MalOD 的检测结果和明文 Faster R-CNN 几乎完全一致. 在 6 个样例图像中, 从左到右, MalOD 检测到的目标总数量分别是 4、2、2、5、18 和 13, 而 Faster R-CNN 总共检测到 4、2、2、6、14 和 14 个目标. 二者检测到的目标数量几乎相同, 且相应的边界框位置、分类结果和置信度表现出高度一致. 这表明, 与 Faster R-CNN 相比, MalOD 在保证强大的安全性的同时, 为单张图像的目标检测提供了与明文几乎相同的检测精度. 即 MalOD 在保护数据免受恶意敌手攻击的同时, 有效地保留了目标检测模型的可用性.

● 运行时间和通信量. 本节在 RGB 图像上评估了输入不同分辨率图像的 MalOD 的运行时间和通信性能. 进行目标检测之前, 同一张图像分别被预处理为 128×96 和 256×192 的分辨率. 表 2 详细列出了 MalOD 每个模块的运行时间以及总的通信开销. 与预期一致, 图像分辨率的降低使得运行时间和全局通信成本也随之减少, 这归因于计算量和数据传输量的降低, 也证明了 MalOD 能够随着图像大小的变化高效地扩展. 值得注意的是, MalOD 整体通信量相对较大, 主要由于在恶意安全模型下, (4, 3) 复制秘密共享和 JMP 协议引入了冗余份额和交叉验证机制

来提供更强的安全保障, 这些机制虽然带来了额外通信成本, 但为了在恶意敌手环境下保障数据的安全性, 这是必要的合理开销.

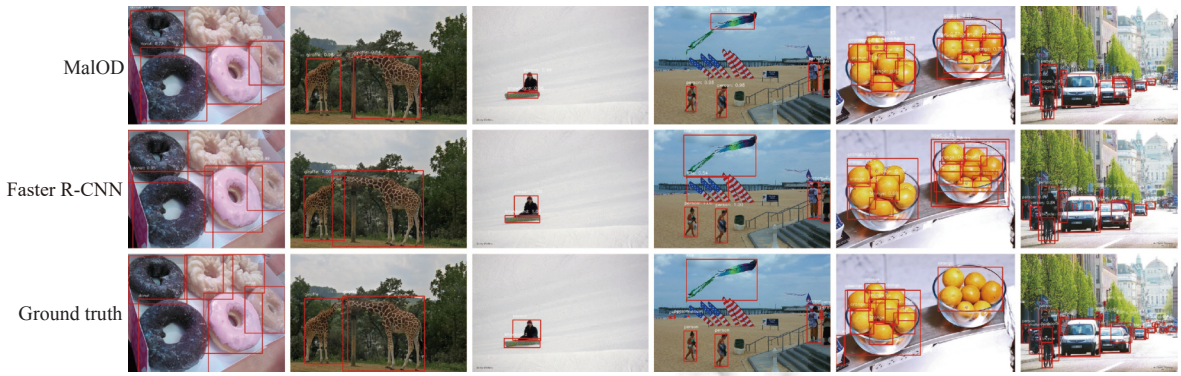

图 5 检测结果可视化对比

表 2 运行时间和通信性能

Image size (resolution)	子模块运行时间 (s)			MalOD总开销	
	SecBackbone	SecRPN	SecRoIA & SecClassification	运行时间 (s)	通信量 (MB)
128×96	46.15	44.8072	891.0869	982.044	37231.3
256×192	138.585	81.914	894.451	1114.95	49470.5

观察不同模块的运行时间可以发现, SecBackbone 和 SecRPN 的运行时间以及总的通信开销随着图像分辨率的提高而增加. 这是由于分辨率的增加, 实现目标检测的计算量更大, 数据传输需求也更高. 而 SecRoIA 和 SecClassification 是最耗时的阶段, 且其执行时间在不同分辨率下保持稳定. 这是因为最终阶段涉及对每个候选区域进行分类和回归, SecRoIA 和 SecClassification 的运行时间与提议的数量成正比增长. 而对于同一张图像, 不同分辨率的输入, SecRPN 生成的候选提议数量几乎保持不变, 使其成为 MalOD 整体性能的主要瓶颈. 可以通过限制 SecRPN 阶段生成的候选区域数量, 从而提高 SecRoIA 和 SecClassification 的效率.

为评估效率相对水平, 本文与现有最新隐私保护目标检测方案[15]进行对比: 在性能接近的 CPU 平台上, 文献 [15] 处理 500×319×3 分辨率图像耗时 1992.8529 s, 而 MalOD 在更高分辨率的图像输入 (512×352×3) 下仅需 1634 s. 在 CPU 执行推理条件下, MalOD 于更高分辨率、防御恶意敌手的更强威胁模型下仍实现 18% 加速, 具体对比见表 3, 表中用加粗表示最优结果.

表 3 性能对比

Model	Image size (resolution)	子模块运行时间 (s)			MalOD时间开销 (s)
		SecBackbone	SecRPN	SecRoIA & SecClassification	
文献[15]	500×319	570.9962	1210.9652	**210.8915**	1992.8529
MalOD	512×352	**474.747**	**289.101**	870.152	**1634**

6.2 性能评估

考虑到安全推理引入的显著计算和通信开销, 在隐私保护机器学习的研究中, 通常采用数据集的较小子集来平衡评估成本和性能评估开销. 本实验分别在 COCO 2017 和 Pascal VOC 2012 验证集中随机选择 200 张图像, 组成两个评估子集, 以增强结果的可靠性和泛化能力. 性能评估采用官方 COCO 评估工具包 (pycocotools), 计算平均精度均值 (mean average precision, mAP) 与平均召回率均值 (mean average recall, mAR), 以全面衡量模型性能. 实验采用 COCO 数据集评估标准, 通过 *IoU* 阈值 0.50–0.95 (以 0.05 为步长) 计算综合 mAP (mAP@0.5:0.95) 和 mAR, 全面衡量模型在不同定位精度下的性能. 同时, 依据 Pascal VOC 等基准的通用规范, 计算 *IoU* 阈值为 0.50 的标准 mAP (mAP@0.5). 在召回率评估中, mAR 在 *IoU* 阈值为 0.50–0.95 范围内, 并分别针对每张图像中最大检

测数 (maxDets) 为 1、10 和 100 的情况进行计算 (分别记为 mAR@1、mAR@10 和 mAR@100), 以综合反映模型在不同检测数量约束下的召回能力. 评估结果如表 4 所示, 其中每张图像最多保留 100 个检测结果, 且评估所有大小的目标.

表 4 MalOD 性能评估

数据集	模型	mAP@0.5	mAP@0.5:0.95	mAR@1	mAR@10	mAR@100
COCO 2017	Faster R-CNN (plaintext)	0.656	0.427	0.336	0.503	0.515
	MalOD (encrypted)	0.543	0.258	0.242	0.332	0.340
	Decrease	0.113	0.169	0.094	0.171	0.175
Pascal VOC 2012	Faster R-CNN (plaintext)	0.717	0.501	0.487	0.568	0.568
	MalOD (encrypted)	0.526	0.313	0.371	0.392	0.392
	Decrease	0.191	0.189	0.116	0.176	0.176

与明文目标检测相比, 在采用四方安全计算和复制秘密共享协议的恶意敌手模型下, MalOD 满足更严格安全性约束, 同时保持了可用的检测性能. 实验表明, MalOD 在多项指标上与明文模型相比仅略有下降: 在 COCO 2017 上, mAP@0.5:0.95 下降 0.169, mAP@0.5 下降 0.113; 在 Pascal VOC 2012 上, 相应降幅分别为 0.189 和 0.191. 召回率方面, maxDets=100 时 MalOD 在两类数据集上的 mAR 下降均不足 0.18. 这些性能变化处于可接受区间, 尤其考虑到其背后安全机制带来的隐私提升. 性能差异主要源于安全多方计算框架的固有特性, 包括: (i) 非线性运算 (如 Π_{SNE}、Π_{SReLU}、Π_{SSoftmax}) 需通过分段线性化以及安全近似实现, 导致特征提取过程中数值分布特性改变, 进而影响边界框回归精度; (ii) 复制秘密共享依赖定点运算, 编码精度和截断操作引入量化误差, 影响预测精度; (iii) 四方计算协议需要将如 RoIAlign、FPN 融合等复杂算子拆解为安全子协议, 这个过程会降低主干网络中特征聚合的精度. 尽管如此, MalOD 在 *IoU*@0.5 条件下在 COCO 2017 和 Pascal VOC 2012 的评估子集上仍分别达到 0.543 和 0.526 的 mAP, 证明其核心检测功能未受显著影响. 相较于仅防御半诚实敌手的现有方案, MalOD 在恶意模型下实现了可比较的精度, 并额外提供可中止安全性与恶意行为检测机制. 在医疗影像、监控等隐私敏感场景中, 该类有限性能代价可视为合理权衡, 也为高隐私要求下的实用目标检测提供了新途径.

值得注意的是, 随着 *IoU* 阈值从 0.5 增加到 0.95, MalOD 的 mAP 的下降与明文检测相当, 这表明加密对高精度目标检测的负面影响得到了很好的控制.

7 总 结

本文提出了一种恶意敌手下隐私保护目标检测框架 MalOD. 该框架基于 (4, 3) 复制秘密共享方案, 能够在不泄露用户输入和模型参数的情况下, 实现安全的目标检测, 并且具有抵御篡改等恶意攻击的能力. 本文在 COCO 2017 和 Pascal VOC 2012 数据集上进行充分的实验, 实验结果表明, 尽管与明文检测相比, MalOD 有更高的计算和通信开销, 但其在多种场景下具有更高的实用性, 适用于对隐私保护要求较高的场景, 如医疗成像和智能监控等. 未来的工作将致力于进一步优化核心算子, 以降低系统的通信与计算开销; 同时, 也将拓展框架的普适性, 研究其在单阶段检测模型及更多计算场景下的适配方案. 此外, 也将研究如何在检测到恶意节点后, 实现协议从四方计算到三方或两方的安全降级, 确保计算任务的鲁棒性和连续性, 从而提升系统在长期在线服务中的实用价值.

References

[1] Cho M, Kim T, Shim M, Wee D, Lee S. Towards multi-domain learning for generalizable video anomaly detection. In: Proc. of the 38th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2024. 1591. [doi: 10.52202/079017-1591]

[2] Liu X, Nejadasl FK, van Gemert JC, Booij O, Pintea SL. Objects do not disappear: Video object detection by single-frame object location anticipation. In: Proc. of the 2023 IEEE/CVF Int'l Conf. on Computer Vision. Paris: IEEE, 2023. 6927–6938. [doi: 10.1109/ICCV51070.2023.00640]

[3] Shi YH, Wang NY, Guo XJ. YOLOV: Making still image object detectors great at video object detection. In: Proc. of the 37th AAAI Conf. on Artificial Intelligence. Washington: AAAI, 2023. 2254–2262. [doi: 10.1609/aaai.v37i2.25320]

[4] Liu TR, Meng QJ, Huang JJ, Vlontzos A, Rueckert D, Kainz B. Video summarization through reinforcement learning with a 3D spatio-

temporal U-Net. IEEE Trans. on Image Processing, 2022, 31: 1573–1586. [doi: 10.1109/TIP.2022.3143699]

[5] Hu YH, Yang JZ, Chen L, Li KY, Sima CH, Zhu XZ, Chai SQ, Du SY, Lin TW, Wang WH, Lu LW, Jia XS, Liu Q, Dai JF, Qiao Y, Li HY. Planning-oriented autonomous driving. In: Proc. of the 2023 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Vancouver: IEEE, 2023. 17853–17862. [doi: 10.1109/CVPR52729.2023.01712]

[6] Cai JH, Zhang L, Dong J, Guo JC, Wang YA, Liao MS. Automatic identification of active landslides over wide areas from time-series InSAR measurements using Faster RCNN. Int'l Journal of Applied Earth Observation and Geoinformation, 2023, 124: 103516. [doi: 10.1016/j.jag.2023.103516]

[7] Liu TR, Luo WH, Ma L, Huang JJ, Stathaki T, Dai TH. Coupled network for robust pedestrian detection with gated multi-layer feature extraction and deformable occlusion handling. IEEE Trans. on Image Processing, 2021, 30: 754–766. [doi: 10.1109/TIP.2020.3038371]

[8] Chen QB, Jin WZ, Ge JY, Liu MD, Yan YC, Jiang J, Yu L, Guo XJ, Li SC, Chen JZ. CP-DETR: Concept prompt guide DETR toward stronger universal object detection. In: Proc. of the 39th AAAI Conf. on Artificial Intelligence. Philadelphia: AAAI, 2025. 2141–2149. [doi: 10.1609/aaai.v39i2.32212]

[9] Liao YR, Wang HN, Lin CB, Li Y, Fang YQ, Ni SY. Research progress of deep learning-based object detection of optical remote sensing image. Journal of Communications, 2022, 43(5): 190–203 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000−436x.2022071]

[10] Huang SZ, Sirejiding S, Lu YX, Ding Y, Liu LH, Zhou H, Lu HT. YOLO-Med: Multi-task interaction network for biomedical images. In: Proc. of the 2024 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP). Seoul: IEEE, 2024. 2175–2179. [doi: 10.1109/ICASSP48485.2024.10446165]

[11] Wang CY, Yeh IH, Mark Liao HY. YOLOv9: Learning what you want to learn using programmable gradient information. In: Proc. of the 18th European Conf. on Computer Vision. Milan: Springer, 2024. 1–21. [doi: 10.1007/978-3-031-72751-1_1]

[12] Wang A, Chen H, Liu LH, Chen K, Lin ZJ, Han JG, Ding GG. YOLOv10: Real-time end-to-end object detection. In: Proc. of the 38th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2024. 3429.

[13] Tian YJ, Ye QX, Doermann D. YOLOv12: Attention-centric real-time object detectors. In: Proc. of the 39th Int'l Conf. on Neural Information Processing Systems. San Diego: Curran Associates Inc., 2025.

[14] Liu Y, Ma Z, Liu XM, Ma SQ, Ren K. Privacy-preserving object detection for medical images with Faster R-CNN. IEEE Trans. on Information Forensics and Security, 2022, 17: 69–84. [doi: 10.1109/TIFS.2019.2946476]

[15] Wang RN, Luo M, Feng Q, Peng C, He DB. Multi-party privacy-preserving Faster R-CNN framework for object detection. IEEE Trans. on Emerging Topics in Computational Intelligence, 2024, 8(1): 956–967. [doi: 10.1109/TETCI.2023.3296502]

[16] Akavia A, Gentry C, Halevi S, Vald M. Achievable CCA2 relaxation for homomorphic encryption. Journal of Cryptology, 2025, 38(1): 5. [doi: 10.1007/s00145-024-09526-1]

[17] Wu LQ, Fu SJ, Luo YC, Yan HY, Shi HY, Xu M. A robust and lightweight privacy-preserving data aggregation scheme for smart grid. IEEE Trans. on Dependable and Secure Computing, 2024, 21(1): 270–283. [doi: 10.1109/TDSC.2023.3252593]

[18] Bai LF, Zhu YF, Li YJ, Wang S, Yang XQ. Research progress of fully homomorphic encryption. Journal of Computer Research and Development, 2024, 61(12): 3069–3087 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.202221052]

[19] Zhao Y, Zhang K, Gao LX, Chen JJ. Privacy and fairness analysis in the post-processed differential privacy framework. IEEE Trans. on Information Forensics and Security, 2025, 20: 2412–2423. [doi: 10.1109/TIFS.2025.3528222]

[20] Li MQ, Tian YL, Zhang JP, Zhao DM. Incentive scheme for federated learning based on zero-concentrated differential privacy. Journal of Communications, 2025, 46(1): 79–92 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2025008]

[21] Zhang PF, Zhu YB, Cheng X, Zhang ZK, Liu XM, Sun L, Fang XJ, Zhang J. Locally differentially private truth discovery algorithm via adaptive pruning. Ruan Jian Xue Bao/Journal of Software, 2025, 36(7): 3405–3428 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7287.htm [doi: 10.13328/j.cnki.jos.007287]

[22] Yang Y, Mu K, Deng RH. Lightweight privacy-preserving GAN framework for model training and image synthesis. IEEE Trans. on Information Forensics and Security, 2022, 17: 1083–1098. [doi: 10.1109/TIFS.2022.3156818]

[23] Ren LQ, Liu ZY, Li FJ, Liang KT, Li Z, Luo B. PrivDNN: A secure multi-party computation framework for deep learning using partial DNN encryption. Proc. on Privacy Enhancing Technologies, 2024, 2024(3): 477–494. [doi: 10.56553/popets-2024-0089]

[24] Patra A, Schneider T, Suresh A, Yalame H. ABY2.0: Improved mixed-protocol secure two-party computation. In: Proc. of the 30th USENIX Security Symp. Vancouver: USENIX Association, 2021. 2165–2182.

[25] Liu WX, Guan YW, Huo JR, Ding YC, Guo H, Li B. A fast and secure Transformer inference scheme with secure multi-party computation. Journal of Computer Research and Development, 2024, 61(5): 1218–1229 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.202330966]

[26] Byali M, Chaudhari H, Patra A, Suresh A. FLASH: Fast and robust framework for privacy-preserving machine learning. Proc. on Privacy

Enhancing Technologies, 2020, 2020(2): 459–480. [doi: 10.2478/popets-2020-0036]
[27] Wagh S, Tople S, Benhamouda F, Kushilevitz E, Mittal P, Rabin T. Falcon: Honest-majority maliciously secure framework for private deep learning. Proc. on Privacy Enhancing Technologies, 2021, 2021(1): 188–208. [doi: 10.2478/popets-2021-0011]
[28] Dalskov A, Escudero D, Keller M. Fantastic four: Honest-majority four-party secure computation with malicious security. In: Proc. of the 30th USENIX Security Symp. Vancouver: USENIX Association, 2021. 2183–2200.
[29] Keller M. MP-SPDZ: A versatile framework for multi-party computation. In: Proc. of the 2020 ACM SIGSAC Conf. on Computer and Communications Security. ACM, 2020. 1575–1590. [doi: 10.1145/3372297.3417872]
[30] Girshick R. Fast R-CNN. In: Proc. of the 2015 IEEE Int'l Conf. on Computer Vision. Santiago: IEEE, 2015. 1440–1448. [doi: 10.1109/ICCV.2015.169]
[31] Ren SQ, He KM, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149. [doi: 10.1109/tpami.2016.2577031]
[32] Lin TY, Dollár P, Girshick R, He KM, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 936–944. [doi: 10.1109/CVPR.2017.106]
[33] Chen MQ, Yu LJ, Zhi C, Sun RJ, Zhu SW, Gao ZY, Ke ZX, Zhu MQ, Zhang YM. Improved Faster R-CNN for fabric defect detection based on Gabor filter with genetic algorithm optimization. Computers in Industry, 2022, 134: 103551. [doi: 10.1016/j.compind.2021.103551]
[34] Keller M, Sun K. Secure quantized training for deep learning. In: Proc. of the 39th Int'l Conf. on Machine Learning. Baltimore: PMLR, 2022. 10912–10938.
[35] Faraji S, Kerschbaum F. Trifecta: Faster high-throughput three-party computation over WAN using multi-fan-in logic gates. Proc. on Privacy Enhancing Technologies, 2023, 2023(4): 224–237. [doi: 10.56553/popets-2023-0107]
[36] Liu FR, Xie X, Yu Y. Scalable multi-party computation protocols for machine learning in the honest-majority setting. In: Proc. of the 33rd USENIX Security Symp. Philadelphia: USENIX Association, 2024. 109.
[37] Aly A, Smart NP. Benchmarking privacy preserving scientific operations. In: Proc. of the 17th Int'l Conf. on Applied Cryptography and Network Security. Bogota: Springer, 2019. 509–529. [doi: 10.1007/978-3-030-21568-2_25]
[38] Koti N, Pancholi M, Patra A, Suresh A. SWIFT: Super-fast and robust privacy-preserving machine learning. In: Proc. of the 30th USENIX Security Symp. Vancouver: USENIX Association, 2021. 2651–2668.
[39] Ma Z, Liu Y, Liu XM, Ma JF, Li FF. Privacy-preserving outsourced speech recognition for smart IoT devices. IEEE Internet of Things Journal, 2019, 6(5): 8406–8420. [doi: 10.1109/JIOT.2019.2917933]
[40] Liu L, Fu SJ, Huang XL, Luo YC, Zhang XY, Choo KKR. SecDM: A secure and lossless human mobility prediction system. IEEE Trans. on Services Computing, 2024, 17(4): 1793–1805. [doi: 10.1109/TSC.2024.3358292]

附中文参考文献

[9] 廖育荣, 王海宁, 林存宝, 李阳, 方宇强, 倪淑燕. 基于深度学习的光学遥感图像目标检测研究进展. 通信学报, 2022, 43(5): 190–203. [doi: 10.11959/j.issn.1000−436x.2022071]
[18] 白利芳, 祝跃飞, 李勇军, 王帅, 杨晓琪. 全同态加密研究进展. 计算机研究与发展, 2024, 61(12): 3069–3087. [doi: 10.7544/issn1000-1239.202221052]
[20] 李梦倩, 田有亮, 张军鹏, 赵冬梅. 基于零集中差分隐私的联邦学习激励方案. 通信学报, 2025, 46(1): 79–92. [doi: 10.11959/j.issn.1000-436x.2025008]
[21] 张朋飞, 朱伊波, 程祥, 张治坤, 刘西蒙, 孙笠, 方贤进, 张吉. 基于自适应剪枝的满足本地差分隐私的真值发现算法. 软件学报, 2025, 36(7): 3405–3428. http://www.jos.org.cn/1000-9825/7287.htm [doi: 10.13328/j.cnki.jos.007287]
[25] 刘伟欣, 管晔玮, 霍嘉荣, 丁元朝, 郭华, 李博. 一种基于安全多方计算的快速 Transformer 安全推理方案. 计算机研究与发展, 2024, 61(5): 1218–1229. [doi: 10.7544/issn1000-1239.202330966]

附录 A 正确性与安全性分析

本附录将对 MalOD 的正确性和安全性进行证明.

A.1 MalOD 的正确性

在 MalOD 中, 每个预处理后的图像被分为 4 个加法份额: $\mathbf{I}^{(0)}$、$\mathbf{I}^{(1)}$、$\mathbf{I}^{(2)}$ 和 $\mathbf{I}^{(3)}$, 然后依次通过 SecBackbone、

SecRPN、SecRoIA 和 SecClassification 模块进行检测. MalOD 的正确性可以通过证明其模块的正确性来证明. 接下来, 本文进行 MalOD 的正确性证明.

定理 A1. MalOD 的组成模块 SecBackbone、SecRPN、SecRoIA 和 SecClassification 是正确的.

证明: SecBackbone 由 Π_{SAdd}、SConv、Π_{SMaxpool}、Π_{SBN} 层、Π_{SLinear} 层、Π_{SReLU} 层和 *SNNI* 组成. 已知 Π_{SAdd}、Π_{SConv}、Π_{SMaxpool}、Π_{SBN}、Π_{SLinear} 和 Π_{SReLU} 协议是正确的[35,36], 因此只需证明 *SNNI* 的正确性. 由于 *SNNI* 仅在本地执行坐标计算, 其正确性显而易见. 因此, SecBackbone 是正确的. SecRPN 模块由 Π_{SConv}、Π_{SSigmoid}、Π_{SAdd}、Π_{SMult}、Π_{SSort}、Π_{SSelect} 等已证明正确性的协议以及本文提出的 *SBBDec* 算子组成. *SBBDec* 按照其计算过程 (正文中公式 (4)–(9)) 调用 Π_{SMult}、Π_{SAdd} 和 Π_{SNE} 构成, 因此其是正确的. 由此可证得 SecRPN 的正确性. MalOD 的第 3 个模块由 SecRoIA、Π_{SSoftmax}、Π_{SConv}、Π_{SLinear}、*SBBDec* 和 Π_{SReLU} 组成. 只需证明 SecRoIA 协议的正确性即可证明该模块的正确性. 除本地计算和 Π_{SAdd} 算法, SecRoIA 还调用了 *SBI* 算法. *SBI* 由 Π_{SMult}、*SCeil* 和 Π_{SFloor} 组成. 根据公式 (2), Π_{SFloor} 是正确的, 因此 *SCeil* 是正确的, 可得 *SBI* 是正确的. 因此, SecRoIA 是正确的, 进而验证了 SecRoIA 和 SecClassification 模块的正确性.

综上所述, 由 SecBackbone, SecRPN, SecRoIA 和 SecClassification 模块构成的 MalOD 是正确的.

A.2 MalOD 的安全性

本节中, 本文根据定义 A1、引理 A1、引理 A2 和引理 A3 对恶意敌手下的 MalOD 进行安全性分析.

定义 A1. 给定一个协议 π, 如果对于真实世界中的任何恶意敌手 $\mathcal{A}$, 存在一个模拟器能够为 $\mathcal{A}$ 生成一个在计算上与真实情况不可区分的视图, 那么就可以说, 该协议在存在恶意敌手的情况下安全地实现了功能 $\mathcal{F}$.

引理 A1[39]. 如果协议 π 中的所有子协议是可模拟的, 那么 π 也是可模拟的.

引理 A2[40]. 如果 r 是一个与值 x 无关且敌手 $\mathcal{A}$ 未知的均匀分布随机数, 那么 $x \pm r$ 也是一个对 $\mathcal{A}$ 未知的均匀分布随机数.

引理 A3[35,36]. 在四方计算的诚实多数恶意敌手模型中, Π_{SAdd}、Π_{SMult}、Π_{SComp}、Π_{SSelect}、Π_{SSort}、Π_{SConv}、Π_{SMaxpool}、Π_{SBN}、Π_{SLinear}、Π_{SReLU}、Π_{SNE}、Π_{SSoftmax}、Π_{SSigmoid} 和 Π_{SFloor} 是安全的.

本文将根据上述定义和引理证明所构建协议的安全性.

定理 A2. 在四方计算的诚实多数恶意敌手模型中, *SCeil* 和 *SNNI* 协议是安全的.

证明: 在 *SCeil* 中, 每个服务器 S_i 的视图包括 $[x]_i$, $[\lfloor -x \rfloor]_i$ 和 $[\lceil x \rceil]_i$. 由于 Π_{SFloor} 是安全的, $[x]_i$ 和 $[\lfloor -x \rfloor]_i$ 是均匀分布随机数. 根据引理 A2 和公式 (2), 可以推导出 $[\lceil x \rceil]_i$ 也是均匀分布随机数. 在 *SNNI* 中, 每个服务器 S_i 的视图由 $[\mathbf{F}]_i$ 和 $[\mathbf{F}']_i$ 构成, 且各方仅在本地进行计算, 不涉及数据交互. 由于 $[\mathbf{F}]_i$ 是由均匀分布的随机数组成, 因此各方根据 $[\mathbf{F}]_i$ 在本地计算得到的 $[\mathbf{F}']_i$ 也由均匀分布的随机数组成. 综上所述, *SCeil* 和 *SNNI* 的所有视图均可模拟且与真实视图不可区分, 因此 *SCeil* 和 *SNNI* 在恶意敌手模型下是安全的.

定理 A3. 在四方计算诚实多数的恶意敌手模型中, *SBI* 和 *SBBDec* 协议是安全的.

证明: 假设恶意敌手 $\mathcal{A}$ 窃听 S_0、S_1、S_2 和 S_3 之间的信道, 并记录每个计算子协议的输入和输出视图, 分别记作 $view_{\text{in}}$ 和 $view_{\text{out}}$. 对于 *SBI* 协议, 有:

输入视图: $view_{\text{in}} = view_{\Pi_{\text{SAdd}}} \cup view_{\Pi_{\text{SFloor}}} \cup view_{SCeil} \cup view_{\Pi_{\text{SMult}}}$;

输出视图: $view_{\text{out}} = output_{\Pi_{\text{SAdd}}} \cup output_{\Pi_{\text{SFloor}}} \cup output_{SCeil} \cup output_{\Pi_{\text{SMult}}}$.

同理, 对于 *SBBDec*, $view_{\text{in}} = view_{\Pi_{\text{SAdd}}} \cup view_{\Pi_{\text{SNE}}} \cup view_{\Pi_{\text{SMult}}}$; $view_{\text{out}} = output_{\Pi_{\text{SAdd}}} \cup output_{\Pi_{\text{SNE}}} \cup output_{\Pi_{\text{SMult}}}$. 根据定理 A2、引理 A1 和引理 A3, 由于 *SBI* 和 *SBBDec* 中 $view_{\text{in}}$ 和 $view_{\text{out}}$ 的所有元素均可模拟, 因此 *SBI* 和 *SBBDec* 的 $view_{\text{in}}$ 和 $view_{\text{out}}$ 均与模拟视图不可区分. 因此 *SBI* 和 *SBBDec* 协议在恶意敌手模型中是安全的.

接下来证明 MalOD 中 SecBackbone、SecRPN、SecRoIA 和 SecClassification 的安全性.

定理 A4. SecBackbone、SecRPN、SecRoIA 和 SecClassification 在四方计算的诚实多数恶意敌手模型中是安全的.

证明: 该定理的证明方法与定理 3 的证明类似.

对于 SecBackbone 模块, 有:

$view_{\text{in}} = view_{\Pi_{\text{SAdd}}} \cup view_{\Pi_{\text{SConv}}} \cup view_{\Pi_{\text{SMaxpool}}} \cup view_{\Pi_{\text{SBN}}} \cup view_{\Pi_{\text{SLinear}}} \cup view_{\Pi_{\text{SReLU}}} \cup view_{SNNI}$;

$view_{\text{out}} = output_{\Pi_{\text{SAdd}}} \cup output_{\Pi_{\text{SConv}}} \cup output_{\Pi_{\text{SMaxpool}}} \cup output_{\Pi_{\text{SBN}}} \cup output_{\Pi_{\text{SLinear}}} \cup output_{\Pi_{\text{SReLU}}} \cup output_{SNNI}$.

对于 SecRPN 模块, 有:

$view_{\text{in}} = view_{\Pi_{\text{SConv}}} \cup view_{\Pi_{\text{SSigmoid}}} \cup view_{\Pi_{\text{SAdd}}} \cup view_{\Pi_{\text{SMult}}} \cup view_{\Pi_{\text{SSort}}} \cup view_{\Pi_{\text{SSelect}}} \cup view_{SBBDec}$;

$view_{\text{out}} = output_{\Pi_{\text{SConv}}} \cup output_{\Pi_{\text{SSigmoid}}} \cup output_{\Pi_{\text{SAdd}}} \cup output_{\Pi_{\text{SMult}}} \cup output_{\Pi_{\text{SSort}}} \cup output_{\Pi_{\text{SSelect}}} \cup output_{SBBDec}$.

对于 SecRoIA 和 SecClassification 模块, 有:

$view_{\text{in}} = view_{\Pi_{\text{SSoftmax}}} \cup view_{\Pi_{\text{SConv}}} \cup view_{\Pi_{\text{SLinear}}} \cup view_{SBBDec} \cup view_{\Pi_{\text{SReLU}}} \cup view_{SBI}$;

$view_{\text{out}} = output_{\Pi_{\text{SSoftmax}}} \cup output_{\Pi_{\text{SConv}}} \cup output_{\Pi_{\text{SLinear}}} \cup output_{SBBDec} \cup output_{\Pi_{\text{SReLU}}} \cup output_{SBI}$.

根据定理 A2、定理 A3、引理 A1 和引理 A3, 以上所有模块的输入视图 $view_{\text{in}}$ 和输出视图 $view_{\text{out}}$ 均可模拟且与真实视图不可区分. 因此在恶意敌手模型下, MalOD 是安全的.

以上证明了 MalOD 在四方计算的诚实多数恶意敌手模型中的隐私性. 下面分析 MalOD 在该威胁模型下的完整性和可验证性.

MalOD 的完整性和可验证性依赖于 (4, 3) 复制秘密共享的冗余设计及 JMP 协议[35]. 具体而言: (1) 冗余架构: 4 个服务器中至多一个恶意敌手, 确保至少三方计算正确; (4, 3) 方案使每方持有三份份额, 任意两方共享两个相同份额. (2) JMP 验证机制: 利用哈希函数验证冗余份额的一致性识别恶意敌手, 在服务器间通信以及重构时进行重叠份额的一致性检测. (3) 协议集成: MalOD 在服务器通信 (如安全卷积协议) 及结果重构阶段均调用 JMP 协议, 确保全流程完整性与可验证性. 第 2 节的基础协议均基于 JMP 通信框架, 其安全性已在文献 [35,36] 中证明. 下文对核心模块进行论证.

● SecBackbone 计算的完整性和可验证性分析: 该模块通过安全深度学习层和 *SNNI* 协议提取多尺度特征图, 需服务器间通信完成计算. 假设存在恶意敌手: (1) 安全深度学习层运算中若检测到偏离行为, 依文献 [35,36] 识别恶意敌手并实现中止安全性; (2) 若无异常, 则执行 *SNNI* 协议和 Π_{SMaxpool} 协议 (见第 5.1 节) 并输出 SecBackbone 结果. *SNNI* 协议为本地计算, 其输出需通过 JMP 协议验证; Π_{SMaxpool} 的通信过程内置 JMP 协议检测篡改行为. 故 SecBackbone 的输出可通过 JMP 协议验证, 满足完整性和可验证性. MalOD 将 SecBackbone 的验证过程与 SecPG 的 Π_{SConv} 通信合并 (算法 5 第 5 行), 在减少通信量的同时完成验证.

● SecRPN 计算的完整性和可验证性分析: 该模块由 ABG, SecPG 和 SecPF 组成. 其中, ABG 由模型提供方计算, 并将通过 (4, 3) 复制秘密共享的形式分发计算结果给服务器, 在本文的威胁模型中, 模型提供方是可信的, 因此各个服务器收到的 ABG 的结果是可信的. 服务器基于 SecBackbone 生成的特征图和 ABG 生成的锚框执行 SecPG: 首先 SecPG 进行 Π_{SConv} 计算, Π_{SConv} 需要调用 JMP 协议进行通信, 此时可以验证 SecBackbone 的输出结果和 Π_{SConv} 的计算完整性; 继而执行安全深度学习层和安全非线性函数, 均可实现对于恶意敌手的检测, 保证计算的完整性和可验证性; 最后进行 *SBBDec* 运算, 其内置 JMP 协议保障指数/乘法运算安全. SecPF 通过子协议调用 JMP 通信确保完整性, 而其输出也可通过 JMP 协议进行验证. 即 SecRPN 输出通过多层次 JMP 验证机制确保安全属性. MalOD 将 SecRPN 输出的验证过程与 SecRoIA 的 *SBI* 通信合并 (算法 7 第 9 行), 优化通信效率.

● SecRoIA 和 SecClassification 计算的完整性和可验证性分析: SecRoIA 在本地进行秘密份额的加法并调用 *SBI* 实现安全对齐. 若 *SBI* 的最终步骤 Π_{Mult} (算法 3 的第 9 行) 检测到偏离行为, 触发可中止安全性; 若不存在偏离协议执行的行为, SecRoIA 的结果输出到 SecClassification 模块进行运算. 因此, SecRoIA 输出的结果具备完整性和可验证性. SecClassification 将 SecRoIA 的输出作为输入, 其实现逻辑与 SecRPN 中的 SecPG 算法一致, 可实现恶意敌手的检测, 其输出具备完整性和可验证性.

在推理完成后, 4 个服务器将持有的秘密份额发送至客户端进行数据重构. 根据 (4, 3) 复制秘密共享方案: 各服务器发送 3 份秘密份额至客户端; 任意两组份额间存在两个重叠份额. 在至多一个恶意敌手的威胁模型下, 客户端通过执行 JMP 协议验证重叠份额的一致性, 识别恶意服务器 (至多一个); 并基于诚实方 (至少三方) 的完整份额

集重构明文检测结果. 该机制通过重叠份额一致性验证确保: 完整性, 即检测过程免受恶意篡改; 可验证性, 即客户端可交叉验证所有输入来源. 从而为 MalOD 目标检测输出提供端到端的安全保障.

作者简介

肖欣怡, 博士生, 主要研究领域为隐私计算.
柳林, 博士, 副教授, CCF 专业会员, 主要研究领域为隐私计算, 流量分析.
郭茜, 博士生, 主要研究领域为隐私计算.
罗玉川, 博士, 副教授, CCF 专业会员, 主要研究领域为隐私计算, 数据与智能安全.
王勇军, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为系统安全, 网络安全及应用安全.
陈荣茂, 博士, 研究员, 博士生导师, 主要研究领域为公钥密码算法设计与应用.
黄俊杰, 博士, 副研究员, 主要研究领域为机器学习, 计算机视觉.
刘天瑞, 博士, 副教授, 主要研究领域为机器学习, 计算机视觉.
付绍静, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为隐私计算, 智能安全.