

学术博士学位论文

面向神经网络训练与推理的隐私保护 关键技术研究

**Research on Key Technologies of
Privacy Preservation for Neural Network
Training and Inference**

杨文梯

2024 年 11 月

国内图书分类号：TP399
国际图书分类号：004

学校代码：10054
密级：公开

学术博士学位论文
面向神经网络训练与推理的隐私保护
关键技术研究

博士研究生：杨文梯
导师：关志涛教授
申请学位：工学博士
学科：控制科学与工程
专业：信息安全
所在学院：控制与计算机工程学院
答辩日期：2024年11月
授予学位单位：华北电力大学

Classified Index: TP399
U.D.C: 004

Dissertation for the Academic Doctoral Degree

Research on Key Technologies of Privacy Preservation for Neural Network Training and Inference

Candidate :	Wenti Yang
Supervisor :	Prof. Zhitao Guan
Academic Degree Applied for :	Doctor of Engineering
Subject:	Control Science and Engineering
Speciality:	Information Security
School:	School of Control and Computer Engineering
Date of Defence :	November, 2024
Degree-Conferring-Institution :	North China Electric Power University

华北电力大学博士学位论文原创性声明

本人郑重声明：此处所提交的学位论文《面向神经网络训练与推理的隐私保护关键技术研究》，是本人在导师指导下，在华北电力大学攻读学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或完成的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。论文由本人撰写，未使用人工智能代写。本声明的法律结果将完全由本人承担。

作者签名：

日期： 年 月 日

华北电力大学博士学位论文使用授权书

学位论文系研究生在华北电力大学攻读学位期间在导师指导下完成的成果，知识产权归属华北电力大学所有，学位论文的研究内容不得以其它单位的名义发表。研究生完全了解华北电力大学关于保存、使用学位论文的规定，同意学校保留并向有关部门递交论文的复印件和电子版本，同意学校将学位论文的全部或部分内容编入有关数据库进行检索，允许论文被查阅和借阅，学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。本人授权华北电力大学，可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

本学位论文属于（请在以上相应方框内打“√”）：

公开 内部 秘密 机密 绝密

作者签名：

日期： 年 月 日

导师签名：

日期： 年 月 日

摘要

近年来，以神经网络为代表的深度学习技术取得了突破性进展，为满足小微企业及个人在各领域对神经网络的需求，诸如 ChatGPT，文心一言等神经网络推理服务应运而生。随着隐私泄露事件频发，国内外隐私保护立法逐步完善，人们的隐私保护意识日益增强，隐私泄露已成为大数据时代的焦点问题。为解决神经网络训练和推理中的隐私泄露风险，充分利用数据的潜在价值，并推动人工智能发展，隐私保护机器学习逐渐成为研究热点。本文研究神经网络训练与推理中的隐私保护关键技术，旨在通过安全多方计算、同态加密和零知识证明等密码学方法，解决神经网络合作训练与推理中的隐私泄露和推理不可验证等问题。具体的，1) 在神经网络训练过程中，针对合作训练存在的训练数据和模型隐私泄露问题，提出基于多密钥同态加密的隐私保护合作训练方案；2) 在神经网络推理过程中，针对模型、推理数据和推理结果的隐私泄露，以及对模型真实性和推理结果正确性的不可验证两个关键问题，提出两个能够满足不同推理场景下隐私保护与可验证需求的神经网络推理方案。其中，可验证神经网络隐私保护多方推理方案采用安全多方计算技术，相比于同态加密，能够实现更加复杂的计算，且不存在计算深度限制，能够更加灵活地应用于不同深度和结构的模型中；可验证神经网络隐私保护加密推理方案采用同态加密技术，无需多方交互，对用户更友好，且避免了多方合谋攻击。本文主要研究内容如下：

1) 针对现有纵向合作学习方案面临的模型逆向攻击和标签推理攻击等问题，提出一种基于多密钥同态加密的隐私保护合作训练方案-SecureSL。首先，本方案基于拆分学习实现纵向合作训练框架，以实现持有不同特征数据的多个参与方之间的神经网络合作训练；其次，本方案采用多密钥同态加密技术，以解决多方合作训练过程中训练数据、标签和模型的隐私泄露问题，以及多方合谋攻击的问题；此外，为了提高加密计算效率，采用单指令多数据流（Single Instruction Multiple Data, SIMD）操作并行处理加密计算，并提出两种 SIMD 友好的点积计算优化方法，同时对训练的计算过程进行适应性的修改以兼容 SIMD 操作。实验结果表明，相比于已有方案，本方案在对准确率不造成明显影响的前提下，可以实现更好隐私保护效果。在效率方面，相比于原始计算方式，本方案所提优化方法一将明文矩阵与密文向量点积计算的同态加密旋转操作由 $O(n^2)$ 降低至 $O(n_1 + n_2)$ ，其中 $n = n_1 \cdot n_2$ 。优化方法二将密文矩阵间乘法运算的加密、同态乘法、旋转开销分别由 $O(n)$, $O(n^2)$, $O(n^2)$ 降低至 $O(1)$, $O(n)$, $O(n)$ 。

2) 针对现有基于安全多方计算的神经网络推理方案中，存在的难以保证模型真实性以及多方返回结果正确性的问题，提出一种可验证神经网络隐私保护多方推理方案-VSecNN。首先，本方案结合零知识简洁非交互式知识论证（Zero-Knowledge Succinct Non-Interactive Argument of Knowledge，zk-SNARKs）与安全多方计算协议，构造一种多方证明生成方法，并通过在推理和证明生成过程切换基于不同代数结构的安全多方计算协议，提高多方证明生成效率；其次，将所构造的多方证明生成方法与神经网络推理过程相结合，并解决结合过程中的计算不兼容问题，实现可验证的神经网络安全多方推理方案。对推理结果的可验证性使得本方案可以抵抗主动敌手攻击，相比于已有安全多方推理方案具有更高的安全性。本方案在不同的模型结构下进行了实验评估，实验结果表明，以单方无隐私保护的可验证神经网络推理方案作为基准方案，本方案在多方执行证明生成过程中，对于全连接模型推理的证明生成的时间相较于基准方案降低了 2-9 倍，验证时间基本保持持平。

3) 在现有基于同态加密的神经网络加密推理方案中，存在恶意服务器或服务器受到密钥恢复攻击等问题，从而破坏推理结果的正确性，导致安全威胁。针对上述问题，提出一种可验证神经网络隐私保护加密推理方案-VHENN。首先，本方案基于环上电路的 zk-SNARKs 协议，设计了环多项式乘法、比特分解等同态加密中重要运算到二次环程序的转换方法，以实现基于环多项式构造的同态加密可验证方案；随后，将可验证同态加密方案与神经网络推理相结合，并对神经网络推理中的非线性运算进行适应性的调整，以构造可验证的加密推理方案。VHENN 不仅可以满足模型、推理数据、推理结果隐私保护以及模型真实性和推理正确性的可验证性，且无需多方交互，满足计算和通信资源受限的用户需求。在实验评估中，本方案采用 SIMD 操作，降低 zk-SNARKs 证明系统的约束数量，从而提升可信设置、证明生成和验证的效率。在实验示例中，相比于未采用 SIMD 操作时的计算，本文方案约束数量降低了 1 至 3 个数量级；相比于对比方案，本文方案在各环节的计算时间降低超过 4 个数量级。

关键词：隐私保护机器学习；安全多方计算；同态加密；零知识证明

Abstract

In recent years, deep learning techniques, particularly neural networks, have made significant advancements. To meet the growing demands for neural network applications from small and medium-sized enterprises as well as individuals across various domains, inference services such as ChatGPT and ERNIE Bot have emerged. However, with the increasing incidence of privacy breaches, privacy preservation legislation has been progressively strengthened both domestically and internationally, and public awareness of privacy issues has been steadily rising. Consequently, privacy leakage has become a critical concern in the era of big data. To mitigate the risks of privacy breaches in neural network training and inference, maximize the potential value of data, and foster the advancement of artificial intelligence, privacy-preserving machine learning has emerged as a prominent research topic. This dissertation explores key technologies for privacy preservation in neural network training and inference, focusing on addressing issues of privacy leakage and inference verifiability in collaborative neural network training and inference through cryptographic methods such as Secure Multi-party Computation (MPC), Homomorphic Encryption (HE), and Zero-Knowledge Proofs (ZKPs). Specifically, 1) In the context of neural network training, a privacy-preserving collaborative training scheme based on multi-key homomorphic encryption is proposed to address the issue of privacy leakage concerning training data and models in collaborative settings; 2) In the context of neural network inference, to address the two key issues of privacy leakage concerning the model, inference data, and inference results, as well as the unverifiability of the model's authenticity and the correctness of inference results, two neural network inference schemes are proposed to meet the privacy preservation and verifiability needs of different inference scenarios. The verifiable privacy-preserving multi-party inference scheme for neural networks utilizes secure multi-party computation technology, which allows for more complex computations compared to homomorphic encryption and imposes no depth limitations. This feature enables more flexible application for models of varying depths and structures. Conversely, the verifiable privacy-preserving encrypted inference scheme for neural networks employs homomorphic encryption technology, which requires no multi-party interaction, making it more user-friendly and avoiding multi-party collusion attacks. The main research contributions are as follows:

- 1) To address model inversion and label inference attacks in existing vertical federated learning schemes based on split learning (SL), a privacy-preserving collaborative training scheme—SecureSL—is proposed based on multi-key

homomorphic encryption. This scheme builds upon the split learning framework to enable vertical collaborative training between participants holding different feature sets. Additionally, it employs multi-key homomorphic encryption to prevent privacy leakage of training data, labels, and models in multi-party collaborative training, while also mitigating the risk of collusion attacks. To further improve the efficiency of encrypted computation, Single Instruction Multiple Data (SIMD) operation is used for parallel processing. Two SIMD-friendly optimization methods for dot product computations are proposed, along with adaptive modifications to the training process to make it SIMD-compatible. Experimental results demonstrate that, compared to existing approaches, this scheme provides stronger privacy preservation without significantly compromising accuracy. In terms of efficiency, compared to the original method, the first optimization reduces the homomorphic encryption rotation operation in plaintext-matrix and ciphertext-vector dot products from $O(n^2)$ to $O(n_1 + n_2)$, where $n = n_1 \cdot n_2$. The second optimization reduces the encryption, homomorphic multiplication, and rotation costs in ciphertext matrix multiplication from $O(n)$, $O(n^2)$, and $O(n^2)$ to $O(1)$, $O(n)$, and $O(n)$, respectively.

2) To address the challenges of ensuring model authenticity and result correctness in existing secure multi-party computation (MPC)-based neural network inference schemes, a verifiable privacy-preserving multi-party inference scheme for neural networks—VSecNN—is proposed. First, this scheme integrates Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs) with MPC protocols to construct a multi-party proof generation method. It optimizes the efficiency of multi-party proof generation by switching MPC protocols based on different algebraic structures during inference and proof generation. Additionally, this proof generation method is integrated with the neural network inference process, resolving computational incompatibilities to enable verifiable secure multi-party inference. The verifiability of the inference results makes the proposed scheme resilient to active adversarial attacks, offering higher security compared to existing secure multi-party inference schemes. The scheme is experimentally evaluated on various neural network models, and results show that, compared to a baseline verifiable neural network inference scheme without privacy protection, our scheme reduces the proof generation time for fully connected models by 2-9 times while maintaining similar verification times.

3) In existing homomorphic encryption-based neural network inference schemes, malicious servers or key-recovery attacks on servers pose a threat to the correctness of inference results, leading to security vulnerabilities. To address these issues, a verifiable privacy-preserving encrypted inference scheme for neural networks—VHENN—is proposed. First, the scheme leverages a zk-SNARKs

protocol over ring circuits, designing conversion methods for essential operations in homomorphic encryption, such as ring polynomial multiplication and bit decomposition, into quadratic ring programs. This enables a verifiable homomorphic encryption scheme based on ring polynomials. Next, the verifiable homomorphic encryption scheme is integrated with neural network inference, where the non-linear operations in the neural network are adaptively adjusted to construct the verifiable encrypted inference scheme. VHENN ensures privacy protection for the model, inference data, and inference results, while providing verifiability for model authenticity and inference correctness, all without requiring multi-party interaction—making it suitable for users with limited computational and communication resources. In the experimental evaluation, the scheme employs SIMD operation to reduce the number of constraints in the zk-SNARKs proof system, thereby improving the efficiency of trusted setup, proof generation, and verification. In the experimental examples, compared to computations without SIMD, our scheme reduces the number of constraints by 1 to 3 orders of magnitude. Furthermore, compared to other approaches, our scheme reduces the computation time across all stages by more than 4 orders of magnitude.

Keywords: Privacy-Preserving Machine Learning, Secure Multi-Party Computation, Homomorphic Encryption, Zero-Knowledge Proofs

目 录

摘要	I
Abstract	III
第 1 章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	3
1.2.1 神经网络隐私泄露威胁	3
1.2.2 神经网络训练阶段的隐私保护	4
1.2.3 神经网络推理阶段的隐私保护	6
1.2.4 总结与分析	12
1.3 研究内容	13
1.3.1 基于多密钥同态加密的隐私保护合作训练方案	14
1.3.2 可验证神经网络隐私保护多方推理方案	15
1.3.3 可验证神经网络隐私保护加密推理方案	15
1.4 章节结构	16
第 2 章 预备知识	18
2.1 密码学基础	18
2.1.1 群、环、域	18
2.1.2 双线性映射	19
2.1.3 敌手模型	19
2.2 同态加密	20
2.2.1 多密钥同态加密	20
2.2.2 BGV 同态加密	22
2.2.3 同态加密 SIMD 操作	23
2.3 安全多方计算	24
2.3.1 算术秘密共享	25
2.4 零知识证明	25
2.4.1 Groth16 协议	26
2.4.2 KZG 多项式承诺	28
2.4.3 二次环程序	29
2.4.4 环上 zk-SNARKs 协议	30

目 录

2.5 神经网络基础	31
2.5.1 神经网络基本结构.....	31
2.5.2 神经网络工作流程.....	32
2.5.3 拆分学习	33
2.6 本章小结	33
第 3 章 基于多密钥同态加密的隐私保护合作训练方案	34
3.1 问题描述	34
3.1.1 存在的挑战.....	34
3.1.2 设计目标与方法.....	35
3.1.3 本章贡献	36
3.2 方案概述	36
3.2.1 系统模型	36
3.2.2 威胁模型	37
3.2.3 SecureSL 方案流程.....	38
3.3 SecureSL 优化方法	41
3.3.1 点积优化方法.....	43
3.3.2 卷积层计算.....	45
3.3.3 全连接层计算.....	46
3.4 隐私保护及安全性分析	47
3.5 实验评估	48
3.5.1 实验设置	48
3.5.2 效率分析	49
3.5.3 隐私保护性能对比.....	50
3.5.4 SecureSL 准确率	53
3.6 本章小结	54
第 4 章 可验证神经网络隐私保护多方推理方案	55
4.1 问题描述	55
4.1.1 存在的挑战.....	56
4.1.2 设计目标与方法.....	57
4.1.3 本章贡献	57
4.2 方案概述	57
4.2.1 系统模型	57
4.2.2 威胁和安全模型.....	58

4.2.3 VSecNN 方案流程	59
4.3 VSecNN 方案设计	60
4.3.1 神经网络 QAP 构造	61
4.3.2 可信设置	62
4.3.3 准确率证明	63
4.3.4 多方推理	63
4.3.5 多方证明生成	64
4.3.6 验证	66
4.4 隐私保护和安全性分析	67
4.5 实验评估	70
4.5.1 实验设置	70
4.5.2 实验结果	71
4.6 本章小结	73
第 5 章 可验证神经网络隐私保护加密推理方案	75
5.1 问题描述	75
5.1.1 存在的挑战	76
5.1.2 设计目标与方法	77
5.1.3 本章贡献	78
5.2 方案概述	78
5.2.1 系统模型	78
5.2.2 威胁和安全模型	78
5.2.3 方案流程	79
5.3 可验证 BGV	80
5.3.1 环多项式乘法 QRP 构造	81
5.3.2 比特分解 QRP 构造	82
5.4 VHENN 方案设计	82
5.4.1 神经网络 QRP 构造	82
5.4.2 加密推理	85
5.4.3 证明生成	85
5.4.4 验证与解密	86
5.5 隐私保护和安全性分析	86
5.6 实验评估	88
5.6.1 实验设置	88
5.6.2 实验结果	89

目 录

5.7 本章小结	92
第 6 章 总结与展望	93
6.1 工作总结	93
6.2 未来工作展望	94
参考文献	96
攻读博士学位期间发表的论文及其它成果	110
攻读博士学位期间参加的科研工作	112
致 谢.....	113
作者简介	115

Contents

Abstract (In Chinese)	I
Abstract (In English)	III
Chapter 1 Introduction.....	1
1.1 Background and significance of the subject	1
1.2 Domestic and International Research Status.....	3
1.2.1 Privacy Leakage Threats in Neural Networks	3
1.2.2 Privacy Preservation in the Neural Network Training Phase	4
1.2.3 Privacy Preservation in the Neural Network Inference Phase	6
1.2.4 Summary and Analysis	12
1.3 Research Content	13
1.3.1 Privacy-Preserving Collaborative Training Scheme Based on Multi-Key Homomorphic Encryption	14
1.3.2 A Verifiable Privacy-Preserving Multi-Party Inference Scheme for Neural Networks	15
1.3.3 A Verifiable Privacy-Preserving Encrypted Inference Scheme for Neural Networks	15
1.4 Chapter Structure	16
Chapter 2 Preliminaries	18
2.1 Basics of Cryptography	18
2.1.1 Groups, Rings, and Fields.....	18
2.1.2 Bilinear Maps.....	19
2.1.3 Adversary Models	19
2.2 Homomorphic Encryption	20
2.2.1 Multi-Key Homomorphic Encryption	20
2.2.2 BGV Homomorphic Encryption	22
2.2.3 SIMD Operation for Homomorphic Encryption.....	23
2.3 Secure Multi-Party Computation	24
2.3.1 Arithmetic Secret Sharing.....	25
2.4 Zero-Knowledge Proof.....	25
2.4.1 Groth16	26
2.4.2 KZG Polynomial Commitment.....	28
2.4.3 Quadratic Ring Programs	29
2.4.4 zk-SNARKs Protocols on Rings.....	30

2.5 Basics of Neural Networks	31
2.5.1 Basic Structure of Neural Networks	31
2.5.2 Neural Network Workflow	32
2.5.3 Split Learning	33
2.6 Summary of This Chapter.....	33
Chapter 3 Privacy-Preserving Collaborative Training Scheme Based on Multi-Key Homomorphic Encryption.....	34
3.1 Problem Description.....	34
3.1.1 Challenges.....	34
3.1.2 Design Goals and Methods	35
3.1.3 Contributions	36
3.2 Overview.....	36
3.2.1 System Model	36
3.2.2 Threat Model.....	37
3.2.3 SecureSL Workflow	38
3.3 Optimization Methods of SecureSL.....	41
3.3.1 Optimization Algorithms	43
3.3.2 Convolution Layer Computation	45
3.3.3 Fully Connected Layer Computation.....	46
3.4 Privacy Preservation and Security Analysis	47
3.5 Experimental Evaluation	48
3.5.1 Experimental Setup.....	48
3.5.2 Efficiency Analysis	49
3.5.3 Comparison of Privacy Preservation Performance.....	50
3.5.4 Accuracy of SecureSL	53
3.6 Summary of This Chapter.....	54
Chapter 4 A Verifiable Privacy-Preserving Multi-Party Inference Scheme for Neural Networks	55
4.1 Problem Description.....	55
4.1.1 Challenges.....	56
4.1.2 Design Goals and Methods	57
4.1.3 Contributions	57
4.2 Overview.....	57
4.2.1 System Model	57
4.2.2 Threat and Security Models	58
4.2.3 VSecNN Workflow	59
4.3 VSECNN Scheme Design.....	60
4.4.1 QAP Construction for Neural Networks.....	61

4.3.2 Trusted Setup	62
4.3.3 Proof of Accuracy	63
4.3.4 Multi-Party Inference	63
4.3.5 Multi-Party Proof Generation	64
4.3.6 Verification	66
4.4 Privacy Preservation and Security Analysis	67
4.5 Experimental Evaluation	70
4.5.1 Experimental Setup.....	70
4.5.2 Experimental Results	71
4.6 Summary of This Chapter.....	73
Chapter 5 A Verifiable Privacy-Preserving Encrypted Inference Scheme for Neural Networks	75
5.1 Problem Description.....	75
5.1.1 Challenges.....	76
5.1.2 Design Goals and Methods	77
5.1.3 Contributions	78
5.2 Overview.....	78
5.2.1 System Model	78
5.2.2 Threat and Security Models	78
5.2.3 VSecNN Workflow	79
5.3 Verifiable BGV	80
5.3.1 QRP Construction for Polynomial Multiplication on Rings	81
5.3.2 QRP Construction for Bit Decomposition	82
5.4 VHENN Scheme Design.....	82
5.4.1 QRP Construction for Neural Networks	82
5.4.2 Encrypted Inference.....	85
5.4.3 Proof Generation and Verification	85
5.4.4 Verification and Decryption of Inference Results	86
5.5 Privacy Preservation and Security Analysis	86
5.6 Experimental Evaluation	88
5.6.1 Experimental Setup.....	88
5.6.2 Experimental Results	89
5.7 Summary of This Chapter.....	92
Chapter 6 Conclusion and Future Works	93
6.1 Summary of Research Work	93
6.2 Prospects for Future Work	94
References	96

Contents

Papers Published During Doctoral Studies	110
Research Work Participated in During Doctoral Studies	112
Acknowledgements	113
Author's Biography	115

第1章 绪论

1.1 研究背景及意义

神经网络作为推动人工智能领域发展的关键技术，其发展离不开海量高质量数据与强大算力的支撑。为了满足小微企业及个人等不具备海量数据和充足算力的用户对深度学习的需求，充分发挥数据潜在价值，各大公司相继推出机器学习即服务（Machine Learning as a Service, MLaaS），促进了诸如 ChatGPT、文心一言等以神经网络为主的机器学习推理服务的发展。然而，近年来隐私泄露事件的频繁发生，对个人和社会都造成了严重威胁。随着国内外数据安全与隐私保护立法逐步完善，人们的隐私保护意识逐渐增强，隐私泄露问题成为大数据时代备受关注的问题。出于对隐私泄露的担忧以及隐私保护立法日趋严格，各公司、组织及个人为防止机密数据泄露，加强了对隐私数据的保护，从而导致数据的流通性变差，在不同组织间逐渐形成“数据孤岛”。数据分散、规模不足，严重影响了目前以海量数据为支撑的神经网络的发展与应用。进一步的，数据无法通过有效手段发挥应有的效用，造成大量数据被浪费。为了解决机器学习中的隐私泄露问题，充分利用数据的潜在价值，推动人工智能领域发展，隐私保护机器学习成为人工智能领域的一大研究热点^[1, 2]。

首先，在神经网络模型训练阶段，为了整合来自各方的数据，并且防止集中式训练中由于收集用户数据造成的隐私泄露威胁，以联邦学习（Federated Learning, FL）、拆分学习（Split Learning, SL）为典型代表的多方合作训练模式得到了广泛的关注与研究^[3, 4]。然而，虽然多方合作的训练模式使得原始数据保留在参与方本地，但攻击者依旧可以通过参与方之间交互的参数信息中获取参与方本地数据的隐私信息。此外，训练模型的部分或全部也对所有参与者公开，模型的安全性难以保证。

其次，在神经网络推理阶段，服务方提供具有良好性能的机器学习模型，用户则可以根据需求将数据发送给服务方以获得推理服务。在这种新型服务模式下，推理数据和结果的隐私泄露以及在保证模型隐私的前提下对推理结果的验证，成为了两个关键问题。神经网络推理的隐私泄露主要存在于两个方面：一方面，直接将数据发送给模型持有方进行推理任务，用户的隐私信息可能会被泄露；另一方面，用户可能会通过模型窃取攻击挖掘模型的参数等敏感信息。攻击者也可能会通过模型逆向攻击推理训练数据的隐私信息。神经网络推理可验证问题主要在于，虽然目前大多神经网络推理服务由大型公司背书以保证服务的可信度，但用户依旧难以验证其得到的推理结果是否可信。而神经网络模

型为服务方的重要资产，无法通过向用户公开的方式来验证模型的真实性及推理结果的正确性，需要在保证模型隐私的前提下实现推理的可验证。因此，研究神经网络训练与推理中的隐私保护技术与推理结果的验证技术，对于充分发挥数据效用、奠定隐私保护机器学习领域的理论基础、推动神经网络在各种场景下的应用意义重大^[5]。

机器学习中通用的隐私保护技术的研究当前主要包括：以安全多方计算（Secure Multi-party Computation， MPC）和同态加密（Homomorphic Encryption， HE）为代表的加密技术，以差分隐私（Differential Privacy， DP）为代表的扰动技术，以及以可信执行环境（Trusted Execution Environment， TEE）为代表的基于硬件的隐私保护技术^[6]。其中，基于 MPC 和 HE 的隐私保护方法具有较高的安全性，能够满足机器学习中多种隐私保护需求，但由于通信和计算开销较大，目前还未能在实际场景中被广泛采用^[7]。基于 DP 的隐私保护方法由于其不会引入较多额外开销而被广泛应用，但通过添加噪音保护隐私会影响模型性能，需要在数据隐私与模型效用间进行权衡。因此，基于 DP 的隐私保护机器学习，在保证模型性能的前提下，往往隐私保护力度较低，无法应用于隐私保护需求较高的场景。此外，由于 DP 依赖于统计分析中的数据扰动机制，无法满足单个数据点的隐私保护需求，从而无法在神经网络推理中使用以保护推理数据的隐私^[8]。基于 TEE 的隐私保护方法的安全性高度依赖硬件环境，对所依赖的硬件设施有较高的要求^[9]。总的来说，若能有效降低通信与计算开销，基于 MPC 和 HE 的隐私保护技术能够适用于更多的机器学习应用场景，并提供更高的安全性。然而，目前基于 MPC 和 HE 的隐私保护机器学习的研究与应用，由于受限于较高的通信与计算开销，尚未能成熟的应用于机器学习领域。此外，在现有基于 MPC 和 HE 的神经网络安全推理中，未充分考虑对模型真实性和推理结果正确性的可验证性。为了在保护神经网络模型参数隐私的前提下，实现用户对推理结果的验证，零知识证明（Zero-Knowledge Proofs， ZKPs）作为一种潜在的解决方案，依然存在诸多问题需要进一步的探索与研究^[10]。

因此，本文旨在通过安全多方计算、同态加密、零知识证明等隐私计算技术，解决神经网络合作训练与推理中的隐私泄露和推理不可验证等问题。为了推动具有隐私保护的神经网络合作训练与推理在实际场景中的应用，研究基于上述密码技术的神经网络隐私保护方案的优化方法，以提高方案效率、安全性等性能。此外，本文前瞻性地探索了零知识证明与安全多方计算和同态加密等技术结合应用于神经网络推理过程的可行性，以实现在保证模型隐私的前提下，用户能够对模型真实性和推理结果正确性的验证。总的来说，在个人隐私

保护意识逐渐增强、国家相关法律法规逐渐严格与完善的背景下，本文所聚焦的神经网络隐私保护问题对于推动神经网络的进一步发展具有重要意义。

1.2 国内外研究现状

本节从神经网络中的隐私泄露威胁、神经网络合作训练与推理中的隐私保护、神经网络训练与推理可验证三个方面，对现有基于安全多方计算、同态加密及零知识证明的隐私保护解决方案的研究进展进行介绍与分析。

1.2.1 神经网络隐私泄露威胁

在神经网络的训练与推理过程中，除了模型和数据直接泄露导致的隐私问题，还存在多种攻击方法可能影响模型和数据的安全性。这些攻击方式主要分为两类：模型攻击和数据攻击，它们分别影响模型本身和训练或推理数据的隐私。在神经网络的合作训练中，攻击者会通过特定的攻击窃取模型和训练数据的隐私信息。在神经网络推理中，主要包括攻击者对模型隐私信息的窃取，以及用户向服务方发送推理数据时造成的隐私泄露。在合作训练阶段存在的攻击主要包括模型逆向攻击^[11-13]和模型提取攻击^[6, 14]。其中，模型逆向攻击又可分为成员推理攻击^[15, 16]、属性推理攻击^[17, 18]以及数据重构攻击^[18-20]等，发生在机器学习推理阶段或合作学习的训练阶段，通过逆向推理来获得训练集的信息；模型提取攻击^[21]发生在机器学习的推理阶段，通过推理结果挖掘模型的参数等敏感信息。在联邦学习的训练模式中，攻击者还可以通过梯度等参数窃取训练数据的隐私信息^[22, 23]。Lyu 等人^[24]总结了联邦学习中面临的数据隐私和安全威胁，以及对应的防御方法。他们将联邦学习中的隐私问题分为类表达推理、成员推理、属性推理、输入数据和标签推理四类，并介绍了基于 HE、DP、MPC 三种隐私保护技术。Wei 等人^[22]展示了敌手如何通过梯度泄露攻击，从共享的梯度或权重更新中重构本地训练数据的隐私信息，并分析了不同超参数设置和不同攻击算法设置对攻击效果的影响。Lam 等人^[23]证明在合作学习中不可信的聚合服务器可以通过梯度推理攻击恢复参与方的隐私训练数据。在以拆分学习为代表的纵向联邦学习中，攻击者还可以通过标签推理攻击挖掘参与方持有的标签中包含的隐私信息^[25, 26]。Fu 等人^[27]提出了针对纵向联邦学习的标签推理攻击方法，恶意攻击者可以从底部模型或更新的梯度中推理标签。Liu 等人^[28]探究了在利用同态加密对数据进行保护后的纵向联邦学习中，恢复标签信息的可能性。研究结果表明，通过训练一个梯度反转模型，标签可以以较高的准确率被重构。Li 等人^[29]针对两方拆分学习结构，提出了一种隐私损失度量方法以量化拆分学习中的标签泄露。此外，在拆分学习的训练过程中，

攻击者还可以通过特定的攻击，可以推理出用户数据中的隐私信息，甚至重构训练数据^[30, 31]。

为了解决上述神经网络合作训练与推理过程中的隐私泄露问题，并且提供较高的安全性，研究人员提出了基于密码学的隐私保护解决方案，以实现合作训练、推理中的数据、标签、模型等信息的隐私保护。

1.2.2 神经网络训练阶段的隐私保护

Shokri 等人^[32]提出在不共享数据的前提下，多方合作学习神经网络模型的方法，并采用差分隐私进一步保护各参与方的隐私。各参与方利用本地数据集独立训练模型，并选择性的共享部分关键参数，其中心思想与之后提出联邦学习^[3]类似，是较早提出的隐私保护神经网络合作训练方法。在此之前，大多隐私保护机器学习的研究，都是针对诸如线性回归、逻辑回归、决策树、支持向量机的传统机器学习算法，采用安全多方计算^[33, 34]，同态加密^[35]，差分隐私^[36, 37]等技术实现隐私保护。联邦学习提出后，研究人员开始更多的关注神经网络合作训练中的隐私保护问题。

（一）合作训练中的隐私保护研究

基于密码学方法的机器学习隐私保护方案，可以在不大幅影响模型准确率的前提下，提供较高的隐私保护力度，因此同样可以应用于神经网络中。在神经网络合作训练过程中，目前主要采用的方法包括通过添加掩码^[3, 38-40]、添加扰动^[4, 41, 42]、安全多方计算^[43-45]、同态加密^[46-49]等方式保护中间值、梯度以及权重等参数，防止攻击者通过中间值、更新的梯度或权重反推出本地训练数据的信息。采用添加掩码的隐私保护方式，对用户延迟及掉线的容忍度较低，虽然当前的研究通过门限加密、多重掩码等方式缓解了用户掉线的影响，但其效率，参数冗余等问题依旧亟待解决。诸如差分隐私等添加扰动的方式，要在隐私保护和数据精度之间进行权衡，添加的扰动越多，隐私保护越强，但数据的精度和实用性会降低。此外，差分隐私主要适用于统计分析和数据聚合，对单个数据点的保护效果有限，不适用于所有类型的数据操作和场景。下面我们主要对基于安全多方计算和同态加密的隐私保护合作训练方案进行分析。

1) **安全多方计算：**Mugunthan 等人^[43]结合安全多方计算和差分隐私技术实现联邦学习中的隐私保护，解决了仅采用差分隐私的方案存在的隐私保护力度较弱的问题。Zhang 等人^[44]提出，在基于安全多方计算的联邦学习聚合方法中，仍在存在一定的隐私威胁，并针对此提出了增强的安全多方计算方法，在与聚合服务器通信前，对更新参数进行两轮分解，服务器仅能访问随机的公共

模型参数。Liu 等人^[45]提出了一种高效的隐私保护拆分学习框架，该方案通过将与隐私数据相关联且不适合采用安全多方计算的部分保留在数据拥有者侧，将其余适合采用安全多方计算的部分（即低秩近似、模型构建和预测）委托给半诚实的服务器，从而提升隐私保护拆分学习的效率。此外，还有诸多采用安全多方计算以保护合作训练模式下训练数据和模型隐私的方案^[50]，这些方案旨在保护梯度、权重等神经网络参数以抵抗模型逆向攻击，达到不泄露原始训练数据的目的。

2) 同态加密：Yang 等人^[46]提出了一种拆分联邦学习框架，为了保护包括输入、模型参数、标签和输出等数据的隐私，该框架将网络分发给不同方进行托管，进一步的为了增强隐私保护力度，在客户端模型聚合过程中引入加法同态加密以防止各方之间的合谋。Khan 等人^[47]构造了基于同态加密的隐私保护拆分学习方案，在客户端加密激活映射并将其发送给服务器，服务器端执行加密操作，随后将加密后的结果返回给客户端。这个过程不会泄露任何客户端的隐私信息。Zhang 等人^[48]提出 BatchCrypt，一种面向跨库联邦学习的高效加法同态加密方案，BatchCrypt 将一个批次的梯度编码为一个长整数，并将其加密到一个密文中，从而降低采用同态加密带来的通信与计算开销。Sav 等人^[49]提出了 POSEIDON，一种基于多密钥同态加密的联邦学习加密训练方法，同时保证了数据和模型的机密性，并且可以抵抗合谋攻击。此外，POSEIDON 采用单指令多数据流（Single Instruction, Multiple Data, SIMD）的方式执行同态操作，可以有效降低同态加密的计算开销。

3) 小结与分析：上述方法主要集中于通过保护发送给服务器的中间值、梯度等信息以防止恶意方通过模型逆向攻击提取训练数据的隐私信息，较少关注模型自身的泄露问题。因此上述方案均忽略了一个较为重要的问题，即合作训练模型的安全问题。因为所有参与方都能得到全部或部分模型参数，这意味着该模型的机密性无法保证。采用全同态加密实现加密形式的模型与数据训练，是一种潜在的解决方案。但上述基于同态加密的方案，除 POSEIDON 外，均采用加法同态加密的隐私保护方式，计算开销相对较大，且无法防止客户端推理其他参与方的隐私信息。此外，上述方法大多集中于具有相同特征、不同样本空间的横向分布数据集的合作训练模型，如横向联邦学习，较少关注样本纵向分布时的合作训练。

（二）外包训练中的隐私保护研究

除了诸如联邦学习、拆分学习的合作训练模式，还有一些工作集中于外包训练过程的隐私保护问题。Phong 等人^[51]延续了 Shokri 等人^[32]的工作，指出了 Shokri 提出的多方合作学习中存在的隐私威胁，并采用加法同态加密增加

隐私保护力度。Lou 等人提出了 Glyph^[52], 一种快速准确的加密数据神经网络训练方法。Glyph 通过设计布尔型的 TFHE 同态加密和算术型的 BGV 同态加密的转换, 解决 BGV 同态加密计算非线性运算的低效问题。Li 等人^[53]提出了 NPMMML, 一种非交互式的隐私保护多方机器学习框架, 通过同态加密对数据进行保护。Mohassel 等人提出 SecureML^[54], 一种基于安全多方计算的隐私保护机器学习方案。数据所有者将数据通过秘密共享的方式外包给两个服务器, 服务器通过安全多方计算实现多个数据所有者的合作训练。Corrigan 等人^[55]提出多方数据聚合分析隐私保护技术, 多个用户将本地数据拆分为 n 个份额并分别共享给 n 个不同的服务器, 服务器持有统计分析函数, 通过秘密共享非交互式证明技术实现安全聚合分析。Mohassel 等人提出 ABY3^[56], 一种可以用于机器学习的安全三方混合协议, 数据所有者将数据的三个秘密份额共享给三个服务器, 服务器通过安全三方混合协议实现安全训练。其中线性运算采用秘密共享技术实现, 非线性函数被近似为分段线性函数, 并采用秘密共享和混淆电路实现。相似的研究还包括 ParSecureML^[57], BLAZE^[58], Trident^[59], Adam in Private^[60], Falcon^[61], Cerebro^[62], HOLMES^[63], BEACON^[64]等。此外, 由于二值神经网络轻量级的特点, 近年来, 有部分工作还致力于二值神经网络训练中的隐私保护问题^[65]。由于基于安全多方计算和同态加密的机器学习训练存在通信和计算开销过大的问题, 上述方案均在计算与通信开销上提出了针对性的解决方案。然而在实际应用中, 基于密码方案的隐私保护神经网络训练方案的效率仍然是亟待解决的问题。

1.2.3 神经网络推理阶段的隐私保护

机器学习即服务的快速发展, 推动了面向神经网络推理的隐私保护技术的研究进展。由于差分隐私的隐私保护作用于整个数据集, 单个数据的安全推理无法通过差分隐私技术实现, 因此目前针对神经网络推理的隐私保护方案大多采用安全多方计算和同态加密等密码技术实现。

(一) 基于安全多方计算的安全推理

1) 安全多方计算 (Secure Two-Party Computation, 2PC): Mohassel 等人提出 SecureML^[54], 基于 2PC 的机器学习安全训练与推理方案。在推理过程中, SecureML 采用秘密共享实现线性运算, 混淆电路实现非线性运算, 采用截断技术实现高效浮点运算, 并构造了 MPC 友好的激活函数以使非线性函数的计算更加高效。此外, 由于加法秘密共享中的乘法三元组与数据无关, 因此可以在线下采用同态加密或不经意传输生成, 并通过协议的向量化提高效率。ParSecureML^[57]通过 GPU 加速, 提高了 SecureML 的效率。Liu 等人提出了

MiniONN^[66]，不同于以往隐私保护推理方案要求改变模型的训练阶段，MiniONN 首次提出把已有的神经网络直接转换为支持隐私保护的 Oblivious 神经网络，并采用与 SecureML 相似的隐私保护技术实现安全推理。Rouhani 等人^[67]首次提出在不牺牲安全性的前提下，对分布式客户端生成的数据进行推理，通过混淆电路实现方案的安全性，并提出了一系列混淆电路优化方法以提高效率。Riazi 等人提出 Chameleon^[68]，一种用户机器学习的混合安全计算框架，采用的技术与 SecureML 相似，Chameleon 基于 ABY 框架^[69]实现安全计算框架，线性操作采用秘密共享，非线性操作采用 GC 或 GMW 协议。CrypTFlow2^[70]是一个面向 TensorFlow 平台的神经网络安全推理框架，提出了首个适用于大型神经网络的 2PC 推理协议。AriaNN^[71]通过一种轻量级的密码协议-函数秘密共享，构造了用于神经网络训练和推理的低交互隐私保护方法。DELPHI^[72]基于 GAZELLE 的工作并进行了改进，将 GAZELLE 中采用的同态加密替换为可以部分在线下计算的秘密共享技术，并提供了可以调整机器学习算法对性能-精度进行权衡规划器。Jha 等人^[73]指出，在神经网络安全推理过程中，主要的计算瓶颈来自如 ReLU 一类的非线性激活函数。因此他们提出 DeepReDuce，一种合理移除神经网络中部分对精度影响较小的 ReLU 函数的优化方法，以降低隐私推理延迟。DeepReDuce 采用 DELPHI 作为样例证明了方案的有效性。为了促进安全多方计算在机器学习实际场景中的应用，Knott 等人^[74]提出 CrypTen，一种通过机器学习框架中常见的抽象（如张量计算、自动微分和模块化神经网络）公开流行的安全 MPC 原语的软件框架，并给出了在文本分类、语音识别、图像分类模型上的基准测试。GForce^[75]是一种 GPU 友好的神经网络遗忘推理方案，解决了采用近似方法带来的延迟和准确率相矛盾的问题，通过制定随机舍入和截断层，将非线性和线性层之间的量化与去量化相结合，使方案摆脱浮点运算。Lehmkuhl 等人^[76]证明，恶意客户端可以通过模型提取攻击攻破半诚实安全的神经网络推理协议。因此提出 MUSE，一种能够抵御恶意客户端的高效两方安全推理协议，采用条件披露秘密协议，在经过认证的秘密共享份额和混淆电路标签之间进行切换，并改进的 Beaver 三元组的生成过程，以提升秘密共享的执行效率。SIMC^[77]对 MUSE 进行了改进，通过设计用于非线性激活函数的协议以提高通信效率，具体的，MUSE 采用 HE 和 Beaver 三元组实现非线性计算，SIMC 采用 OT 和 onetime pad 实现非线性计算。Rathee 等人^[78]建立了一个面向 32 位单精度浮点运算和数学函数的 2PC 库-SecFloat，解决了现有支持浮点数的 2PC 库计算不精确的问题，并展示了一个神经网络安全推理应用示例以证明所提 2PC 库的优越性。Huang 等人提出 Cheetah^[79]，线性层采用同态加密，非线性层采用混淆电路，并通过合理

设计基于 HE 的同态操作和低通信的非线性函数密码原语提高方案的效率。Dalskov 等人^[80]调研了基于 MPC 的神经网络安全推理任务中两个重要问题：如何从已有框架中获得 MPC 友好的模型，而不需要特定的转换协议或更改已有模型；如何将现有 MPC 框架以开箱即用的形式应用到神经网络安全推理中。并通过量化、MPC 技术的使用、优化、实验等方面对上述两个问题进行了探讨。此外，近期有部分工作^[81-84]针对 Transformer 模型和大语言模型的安全推理进行了研究，提出了基于 MPC 的多方推理方案。

2) 安全三方计算 (Secure Three-Party Computation, 3PC): Wagh 等人提出了 SecureNN^[85]，一种支持神经网络安全训练和推理的 3PC 协议，并首次提出了支持恶意敌手模型的神经网络训练和推理安全计算。Shen 等人^[86]延续了 SecureNN 的工作，提出了一种高效的安全三方框架以实现隐私保护神经网络推理。他们通过构造用于多种激活函数的高效 3PC 协议实现相较于 SecureNN 更高的效率。ABY3^[56]的推理阶段与训练阶段相似，相较于 SecureML 效率提高了上百倍，此外，作者还将 Chameleon 和 MiniONN 改造为基于 3PC 协议的方案，对比之下推理时间快了近千倍。CrypTFlow^[87]构建了一个包含三个组分的安全推理框架：Athos，TensorFlow 到 MPC 协议的端到端编译器；Porthos，基于 SecureNN 的 3PC 安全推理协议；Aramis，将针对半诚实敌手的 MPC 协议编译为针对恶意敌手的 MPC 协议的编译器。BLAZE^[58]面向安全外包计算的场景，构造了安全三方外包计算框架，并提出一种点积协议和截断方法提高效率。已有通过采用 MPC 友好的激活函数等方式实现安全推理的方案，由于需要对机器学习算法和 MPC 算法进行调整，可能对效率或准确率带来影响。为了解决这个问题，Attrapadung 等人^[60]针对 MPC 不友好的操作如整数除法、指数运算、开根运算等，提出安全有效的协议。Wagh 等人^[61]提出 Falcon，一种支持诚实大多数的恶意安全隐私保护深度学习框架。Falcon 结合 SecureNN 和 ABY3 中的技术，实现了效率的提升。Tan 等人提出 CryptGPU^[88]，一种建立在 PyTorch 和 CrypTen^[74]之上的 MPC 框架，在 CryptGPU 中，所有的线性和非线性操作都是在 GPU 上实现的，因此大大降低了神经网络安全训练和推理的时间。

3) 安全 n 方计算：Trident^[59]是一种面向隐私保护机器学习的 4PC 框架，该框架提供了算术、布尔、和混淆电路之间的高效转换，其效率超过了 ABY3、SecureNN 等安全三方计算。Byali 等人提出了 FLASH^[89]，一种快速、鲁棒的隐私保护机器学习框架。FLASH 采用 4PC 协议，实现了可保证输出交付的最强安全概念，并通过针对点积、截断的优化技术，提升了在 ABY3 和 SecureNN 中较弱的中止安全保障，提高了系统鲁棒性。此外，FLASH 同时实现了传统

深度神经网络和二值神经网络的安全推理。Tetrad^[90]改进了 Trident 的工作，通过无开销的截断、针对算术和布尔运算的多输入乘法协议、针对混淆电路的混合协议框架、不同运算之间的转换机制四个方面提高了方案的效率。Dalskov 等人^[91]提出了一种支持诚实大多数的恶意安全模型的 4PC 协议，以实现安全、鲁棒的神经网络安全外包计算。Koti 等人^[92]提出 SWIFT，一种面向神经网络安全外包计算的高效、恶意安全、输出交付保证的安全三方协议，并将其 3PC 协议扩展为 4PC。目前，多于三方设置的神经网络安全推理方案大多是出于增强安全性的考虑，使其方案能够抵抗诚实大多数的恶意安全模型，或抵抗恶意客户端模型^[93]。

4) 小结与分析：上述基于安全多方计算的神经网络安全推理研究，大多集中于提高效率、通过增加参与方数量提高安全性以及不同神经网络模型的适用性等方面的研究。对于推理的可验证性关注较少，且未能实现抵抗不诚实大多数的恶意安全模型。

(二) 基于同态加密的安全推理

Bost 等人^[94]作为较早期的基于同态加密的机器学习推理方案，基于 paillier 同态加密和层次同态加密构造了针对超平面决策、朴素贝叶斯、决策树等机器学习算法的加密数据分类方法。Dowlin 等人^[95]提出 CryptoNets，首个针对神经网络的加密推理方案，其采用全同态加密和多项式近似的方法实现加密数据的神经网络安全推理，并采用 SIMD 技术增加同态加密的计算效率。CryptoNets 构造 9 层的卷积神经网络（Convolutional Neural Networks, CNN），并应用于 MNIST 数据集，实现了 99% 的准确率，每小时可以预测 58982 条加密数据，每个处理器可以并行处理 4096 条数据，推理延迟为 250s。Hesamifard 等人提出 CryptoDL^[96]，针对基于同态加密的安全推理中非线性函数多项式近似问题，设计了激活函数近似方法，在效率与准确率间进行平衡，使得采用低阶多项式就能实现较高的推理准确率。CryptoDL 在 MNIST 数据集上实现了 99.52% 的准确率，每小时能对 16400 个数据进行预测。Juvekar 等人提出 GAZELLE^[97]，一种可扩展、低延迟的神经网络推理系统，线性计算采用打包的加法同态加密，非线性计算采用混淆电路。GAZELLE 支持 SIMD 的计算方式，并提出同态线性代数核，将神经网络线性层映射到优化的同态矩阵向量乘法和卷积例程。GAZELLE 的推理速度比 CryptoNets 快三个量级。Dathathri 等人^[98]针对基于全同态加密的神经网络推理方案，提出了一个同态加密编译器，优化同态加密在神经网络推理中的使用。通过该编译器，可以直接把 HEAAN 和 SEAL 库中的同态加密方案用于神经网络。Xu 等人^[99]提出了安全、可验证的神经网络推理外包方案，基于层次同态加密和多项式近似，将模型和

数据加密发送给云服务器，在云服务器进行加密外包推理。MP2ML^[100]扩展了nGraph-HE-一个与现有 DL 框架兼容的同态加密框架，并实现 CKKS 和 ABY 之间的转换以解决 nGraph-HE 中评估非多项式函数时可能造成的隐私泄露问题。Zhang 等人^[101]针对神经网络安全推理中基于 HE 的线性计算优化问题，提出了 GALA，通过减少基于 HE 的线性操作中低效的旋转操作（Perm）来降低整体计算开销。GALA 可以很好的应用到如 GAZELLE 等方案中以提升推理效率。Chen 等人^[102]设计了 BFV 和 CKKS 同态加密的多密钥变体，并将其应用在神经网络安全推理中。多密钥同态加密可以很好的解决神经网络安全训练和推理过程中的合谋攻击。类似的，Lu 等人^[103]提出了 PEGASUS，一种可以在不解密的情况下在打包的 CKKS 密文和 FHEW 密文之间的进行切换，从而有效利用 CKKS 评估算术函数，FHEW 评估非线性函数。PEGASUS 在决策树和 K-means 算法中进行了验证，并且有潜力应用于神经网络的加密推理中。Jovanović 等人^[104]首次提出支持隐私保护推理的可靠神经网络，采用全同态加密构建系统模块以保证推理系统的公平性和鲁棒性。SortingHat^[105]通过同态加密和译码实现决策树的隐私评估。部分工作^[106, 107]还致力于提高同态加密应用于神经网络推理中的效率。此外，部分基于全同态加密的二值神经网络安全推理方案^[108, 109]，采用 TFHE 和 FHEW 等全同态加密方案，在无需近似激活函数的前提下，实现了神经网络的加密推理。

小结与分析：上述基于同态加密的神经网络加密推理方案，面临的主要问题是计算开销与计算深度限制。采用有限级数同态加密限制了加密计算的深度，使得方案无法在深度神经网络中应用。而采用全同态加密会导致更高的计算开销。此外，基于同态加密的安全推理还存在非线性函数的计算问题，目前主要解决办法是采用多项式近似，少数研究提出了算术同态加密和布尔同态加密的转换以实现非线性函数的计算，但也带来了额外的计算开销。

（三）可验证神经网络推理

如上所述，目前已有较多研究解决神经网络推理中的隐私保护问题。然而，在神经网络推理服务中，存在两个不可忽视的问题影响推理结果：1) 由于服务提供方对于用户并不是完全可信的，他们可能会夸大其模型的准确率，或在推理服务中提供准确率较低的模型而非最初宣称的模型。2) 服务提供方可能会为了降低计算开销等目的，提供错误的推理结果。目前已有部分多方安全推理方案通过实现恶意安全模型防止错误的推理结果，但这些抵抗恶意安全的方案均假设参与方是诚实大多数的，仅能容忍 1 个^[59, 61]或 $t < n/2$ 个^[91]恶意方，其中 t 为腐败阈值， n 为参与方数量。因此，这些方案只能在恶意方的数量不超过腐败阈值时，才能保证计算的正确性。此外，上述安全推理方案也无法在

保证模型隐私不被泄露的前提下，实现对模型真实性的验证。

现有研究提出了基于零知识证明的可验证神经网络推理方案，可以在保证模型隐私的前提下，实现模型真实性或结果正确性的验证。当前基于零知识证明的神经网络推理方案主要分为两类^[110]，一类基于交互式零知识证明协议，另一类则基于常见的非交互式零知识证明协议-零知识简洁非交互式知识论证（Zero-Knowledge Succinct Non-Interactive Argument of Knowledge，zk-SNARKs）。

1) **交互式可验证神经网络推理：**SafetyNets^[111]是首个针对不可信服务器提供的可验证神经网络推理方案。该方案采用了和校验协议，需要用户与云服务器进行交互才能实现验证。然而，SaftyNets 只保证了计算结果的正确性，而没有关注模型真实性的验证。Liu 等人^[112]提出 zkCNN，一种面向卷积神经网络推理和准确性验证的零知识证明方案，采用和校验技术构造高效零知识证明协议，并结合多项式承诺协议，从而保证推理结果的正确性和完整性。Weng 等人^[113]提出 Mystique，一种支持算术和布尔值、公开承诺和私人验证、定点和浮点数高效转换的零知识证明协议，并将 Mystique 融入到基于 TensorFlow 的隐私保护框架 Rosetta 中，以实现在大型神经网络推理中的应用。Hao 等人^[114]提出了针对机器学习非线性函数的首个零知识证明框架，通过对查找表的构建实现高效的非线性激活函数的零知识证明构造。

2) **非交互式可验证神经网络推理：**vCNN^[115] 和 ZEN^[116] 均基于 zk-SNARKs 构造了可验证的神经网络推理方案。vCNN 提出通过高效二次算术程序（Quadratic Arithmetic Programs，QAPs）处理卷积层和全连接层，通过二次多项式程序（Quadratic Polynomial Program，QPP）高效处理 ReLU 函数和池化层，并将其连接以提高整体效率。ZEN 则通过构造一阶约束系统（Rank-1 Constraint System，R1CS）友好的神经网络量化过程以提升构造神经网络推理证明的效率。VeriML^[117] 提出了基于简洁非交互式知识论证（Succinct Non-Interactive Argument of Knowledge，SNARKs）的可验证 MLaaS 框架，保证机器学习任务在不受信任的服务器上正确执行，并支持线性回归、逻辑回归、神经网络、支持向量机、K-means 和决策树等多种机器学习模型。Weng 等人^[118]提出了 pvCNN，通过同态加密和 zk-SNARKs 来实现卷积神经网络推理中的隐私保护和可验证性。他们将模型分为 PriorNet 和 LaterNet，其中 PriorNet 保持私有，由模型所有者持有，而 LaterNet 为非隐私部分，委托给服务器进行计算。然后，这会导致委托模型缺乏足够的隐私保护。此外，pvCNN 是在不同的阶段使用 HE 和 zkSNARKs，其可验证性仅在服务器端有效。为了能够在较大的模型中应用 zk-SNARKs 协议，Chen 等人^[119]提出了 ZKML，可以应

用于先进的视觉模型、GPT-2 等大模型的验证。ZKML 设计了优化的编译器，将 TensorFlow 中的模型编译为 halo2 zk-SNARK 证明系统的电路，并通过各种优化策略提升效率。

3) 小结与分析: 上述方案前瞻性地解决了神经网络推理中的可验证问题，在保护模型不被泄露的前提下，实现用户对模型真实性和结果正确性的验证。然而，目前尚未有相关方案同时实现推理数据、推理结果、模型隐私保护及模型真实性和推理结果正确性的可验证。少数工作尝试在神经网络安全推理方案之上实现可验证，但这些方案或仅针对简单的机器学习模型，如支持向量机、线性回归等^[120, 121]；或采用诸如敏感样品生成、混合和检验等方法^[99, 122]，这些方案的可验证性本质上是概率性的，并且这些可验证方案仅验证计算结果的正确性，均未考虑对模型真实性的验证。

1.2.4 总结与分析

目前基于密码学技术的隐私保护机器学习，主要采用安全多方计算和同态加密技术保证训练和推理过程中的数据机密性，采用零知识证明保证推理结果的完整性和可验证性。

基于安全多方计算的隐私保护机器学习方案，大多集中于基于 2PC 和 3PC 构造支持半诚实对手安全模型的机器学习训练与推理方案。为了实现诚实大多数的恶意安全方案，部分工作基于 4PC 实现隐私保护机器学习。但这些抵抗恶意安全的方案仅能容忍部分参与方为恶意的，只有在恶意方的数量不超过阈值时，才能保证计算的正确性。此外，基于 MPC 的隐私保护机器学习，目前面临的主要问题是通信开销过大造成的效果问题，因此现有工作主要研究如何降低 MPC 的通信开销，如降低 MPC 协议的交互轮数、构造非交互式协议等。

基于同态加密的隐私保护机器学习方案，在训练过程中，当前多采用加法同态加密发送给聚合服务器的梯度或权重等交互信息，无法适应模型仅向部分参与方公开的场景；在推理过程中，采用全同态加密，保护推理数据的机密性，当前的研究主要分为三类：完全采用算术型同态加密如 CKKS，并通过多项式近似将非线性激活函数近似为线性函数；结合算术型同态加密和布尔型同态加密，线性操作通过算术型同态加密如 CKKS 实现，非线性操作通过布尔型同态加密如 FHEW 实现，并构造两种同态加密之间的转换协议；结合同态加密与安全多方计算，线性操作采用同态加密实现，非线性操作采用 MPC 实现，并设计 HE 与 MPC 的转换协议。已有基于 HE 的隐私保护方案，大多为半诚实安全，对构造恶意安全或可验证方案的研究较少。此外，目前基于 HE 的隐私保护机器学习，尤其是基于全同态加密的方案，主要面临的问题是

计算开销过大。由于层次同态加密对计算深度有限制，而将层次同态加密转换为全同态加密，消除计算深度的限制，又会造成更多的计算开销。因此，出于效率的限制，基于 HE 的训练与推理方案，相比于基于 MPC 的训练与推理方案，研究较少且难以在较深层的神经网络中实现高效应用。但由于 HE 不存在合谋攻击等问题，且不需要多方协作完成，因此在某些场景下比 MPC 更通用，无法被基于 MPC 的解决方案完全替代。

基于零知识证明的可验证神经网络推理方案，主要分为交互式和非交互式两类。交互式方案主要研究线性层和非线性层证明生成方案的转换，主要瓶颈为通信开销等带来的效率问题。非交互式方案主要研究如何更加高效的构造神经网络的计算电路，从而降低证明生成时间。主要问题在于公共参考字符串（Common Reference String, CRS）的尺寸较大、内存要求大、以及证明生成时间长。但非交互式方案无需用户参与交互，更适用于 MLaaS 的场景。目前基于零知识证明的可验证神经网络推理，仅关注了如何在保护模型隐私的前提下实现可验证，而忽略了用户推理数据和推理结果的隐私保护问题。

1.3 研究内容

本文从神经网络合作训练与推理中的隐私泄露问题出发，分析归纳现有相关研究存在或尚未解决的问题，基于安全多方计算、同态加密、零知识证明等密码学技术，以 MLaaS 应用中的实际需求为导向，开展面向神经网络训练与推理的隐私保护关键技术研究。如图 1-1 所示，本文聚焦于神经网络的两个关键阶段——训练阶段和推理阶段，针对现有隐私保护方案中存在的问题与挑战，提出了相应的解决方案。

首先，在神经网络训练过程中，虽然现有合作训练模式使得原始数据保留在参与方本地，从而保护训练数据的隐私。但攻击者依旧可以通过参与方之间交互的参数信息中获取参与方本地数据的隐私信息。另外，目前已有方案大多集中于横向合作训练，对纵向合作训练中的隐私保护研究较为缺乏。因此，本文针对上述问题，提出基于多密钥同态加密的隐私保护合作训练方案。

其次，在神经网络推理过程中，已有方案大多采用安全多方计算与同态加密实现推理过程中数据的隐私保护。这两种技术各自适用不同的场景。安全多方计算能够实现更加复杂的计算，且不存在计算深度限制，能够更加灵活地应用于不同深度和结构的模型中，相较于同态加密，当前的研究与应用更为广泛；而同态加密无需多方交互，在神经网络推理场景下对用户更友好，且避免了多方合谋攻击的问题。然而，虽有已有较多隐私保护推理方案的研

究,但现有方案均未能解决在保护模型隐私的前提下实现推理可验证的问题。因此,针对上述问题,本文提出两个能够满足不同推理场景下隐私保护与可验证需求的神经网络推理方案:可验证神经网络隐私保护多方推理方案、可验证神经网络隐私保护加密推理方案,两个方案面向用一类问题,但能够在不同的场景与需求下互相补充。本文主要研究内容如下:

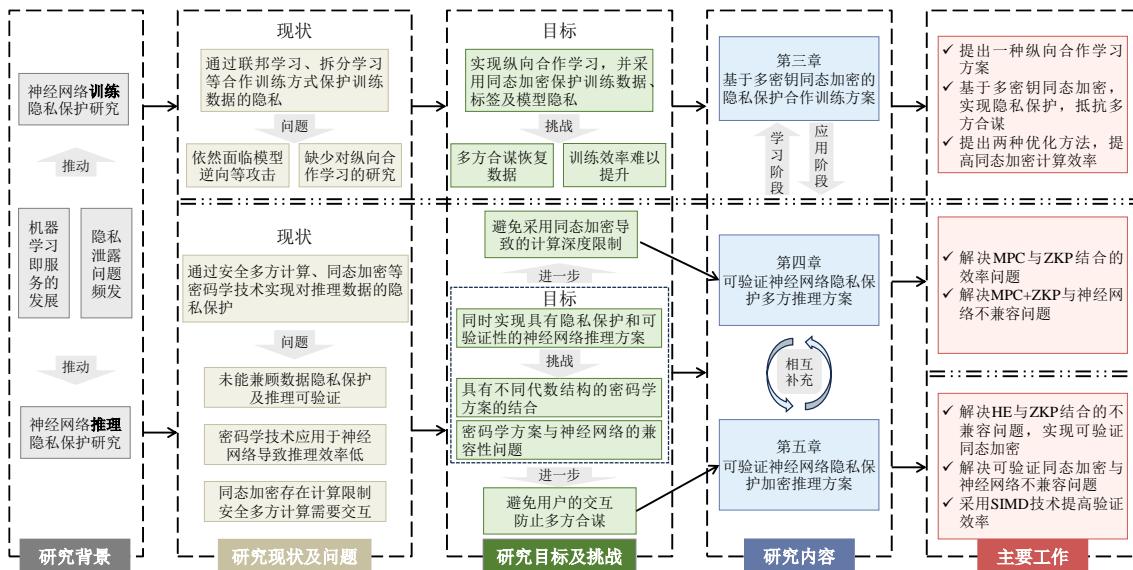


图 1-1 研究内容概览
Fig.1-1 Overview of research content

1.3.1 基于多密钥同态加密的隐私保护合作训练方案

目前已有神经网络隐私保护合作训练方案,大多集中于横向联邦学习,仅有少数方案通过拆分学习等方法实现了纵向合作学习框架。在基于拆分学习的合作训练中,虽然原始数据保留在参与方本地,恶意参与方仍然可以从拆分学习的交互信息中推理出参与方敏感信息,包括训练数据的属性,标签等。现有拆分学习方案中最常用的隐私保护方法是添加噪声扰动,但这种方法需要在隐私保护与模型效用之间进行权衡。考虑到上述问题,本文提出了提出一种基于多密钥同态加密的隐私保护合作训练方案-SecureSL。首先,本方案基于拆分学习实现纵向合作训练框架,以实现持有不同特征数据的多个参与方之间的神经网络合作训练;其次,本方案基于多密钥同态加密技术,以解决多方合作训练过程中训练数据、标签和模型的隐私泄露问题,以及多方合谋攻击的问题;此外,为了提高加密计算效率,采用单指令多数据流 (Single Instruction Multiple Data, SIMD) 技术并行处理加密计算,并提出两种 SIMD 友好的点积计算优化方法,同时对训练的计算过程进行适应性的

修改以兼容 SIMD 技术。实验结果表明，相比于原始计算方式，本方案所提优化方法一可以将明文矩阵与密文向量点积中的同态加密旋转操作由 $O(n^2)$ 降低至 $O(n_1 + n_2)$ ，其中 $n = n_1 \cdot n_2$ 。优化方法二可以将密文矩阵间乘法运算的加密、同态乘法、旋转开销分别由 $O(n)$, $O(n^2)$, $O(n^2)$ 降低至 $O(1)$, $O(n)$, $O(n)$ 。此外，相比于已有基于噪声扰动的隐私保护方案，本方案在对准确率不造成明显影响的前提下，可以实现更好隐私保护效果。

1.3.2 可验证神经网络隐私保护多方推理方案

随着对隐私泄露问题的关注度的提升，目前已有较多方案采用安全多方计算技术实现神经网络的安全推理。然而，如何在现有安全多方推理方案的基础上，实现对模型真实性和推理结果正确性的可验证，并且不泄露模型的隐私信息，成为神经网络推理服务下的一大挑战。为了解决上述问题，本文提出了一种可验证神经网络隐私保护多方推理方案-VSecNN，本方案可以同时实现对模型、推理数据和推理结果的隐私保护以及对模型真实性和推理结果正确性的可验证性。对推理结果的可验证性使得本方案可以抵抗主动敌手攻击，相比于已有安全多方推理方案具有更高的安全性。首先，本方案结合一种 zk-SNARK 协议-Groth16 与安全多方计算协议，构造一种多方证明生成方法，从而为现有安全多方推理方案实现可验证性。本方案通过切换基于不同代数结构的 MPC 协议实现了 Groth16 与 MPC 的高效结合；其次，将所构造的多方证明生成方法与神经网络推理过程相结合，并通过设计神经网络推理到二次算术程序（Quadratic Arithmetic Programs, QAPs）的转换，保证了神经网络推理过程与密码学方案的兼容，实现可验证的神经网络安全多方推理方案。本方案在多个公共数据集和不同结构的神经网络模型下进行了实验评估，实验结果表明，以单方可验证推理作为基准方案，本方案在多方执行证明生成过程中，对于全连接模型推理的证明生成的时间相较于基准方案降低了 2-9 倍，验证时间基本保持持平。

1.3.3 可验证神经网络隐私保护加密推理方案

在非交互式、无需多方参与的场景下，基于同态加密的神经网络加密推理方案已得到广泛研究。在现有方案中，通常设置执行加密推理的服务器为被动敌手模型，在这个假设中，服务器虽然会试图挖掘用户的隐私信息，但会按照既定的协议执行加密推理过程，并返回正确的计算结果。然而，在实际应用中，可能存在服务器为恶意或受到如密钥恢复攻击等恶意攻击的情况，从而破坏推理结果的正确性。因此，在基于同态加密的神经网络加密推理中，同样也面临

着推理不可验证的问题。针对上述问题，提出一种可验证神经网络隐私保护加密推理方案-VHENN。首先，本方案基于 Rinocchio^[123]，一种用于环上电路的 zk-SNARK 协议，设计了环多项式乘法、比特分解等同态加密中重要运算到二次环程序的转换方法，以实现基于环多项式构造的同态加密可验证方案；随后，将可验证同态加密方案与神经网络推理相结合，并对神经网络推理中的非线性运算进行适应性的调整，以构造可验证的安全推理方案，实现满足模型、推理数据、推理结果隐私保护以及模型真实性和推理正确性可验证的神经网络加密推理方案 VHENN。为了提高零知识证明的生成效率，本方案采用 SIMD 技术，通过并行计算降低约束数量。实验结果表明，相比于未采用 SIMD 技术的实现方式，本方案在零知识证明构造过程中约束数量降低幅度达到 1 至 3 个数量级。相比于对比方案，本方案在可信设置、证明生成和验证等环节的计算时间缩短了超过 4 个数量级。

1.4 章节结构

本文的章节结构主要包括研究背景及研究现状介绍、研究方案的预备知识、研究方案的详细描述、总结与展望等内容，各章节安排如下：

第一章为绪论，包括四个小节。第一节为研究背景及意义，首先介绍了神经网络在合作训练与推理阶段面临的隐私泄露威胁，并简要介绍了当前常用技术及存在的问题，进而阐述说明本文的研究意义；第二节为国内外研究现状，首先阐述了神经网络中导致隐私泄露的攻击方式，随后详细介绍了神经网络训练阶段与推理阶段的隐私保护技术及推理阶段的可验证方案，并且总结分析了目前已有方案的研究趋势与存在的问题；第三节为研究内容，本文主要解决的问题及研究内容；第四节为章节结构，介绍了本文各章节安排。

第二章为预备知识。介绍了本文三个研究内容共有的基础知识，包括群环域、双线性映射、敌手模型、同态加密、安全多方计算、零知识证明等密码学基础知识以及神经网络相关的基础知识。

第三章为基于多密钥同态加密的隐私保护合作训练方案。本章首先总结分析了神经网络合作训练中存在的问题，及本章的目标与贡献；然后对基于多密钥同态加密的拆分学习方案进行了概述；随后给出了 SIMD 友好的优化方法，基于多密钥同态加密的拆分学习方案在卷积层和全连接层的详细计算过程；最后通过安全性分析与实验评估证明了本方案的安全性及性能。

第四章为可验证神经网络隐私保护多方推理方案。本章首先总结分析了当前可验证神经网络推理与安全多方推理方案中存在的问题，及本章的目标与贡献；然后对基于 Groth16 与 MPC 协议的可验证神经网络隐私保护多方推理方

案进行了概述；随后给出了方案的详细设计；最后通过安全性分析与实验评估证明了本方案的安全性及性能。

第五章为可验证神经网络隐私保护加密推理方案。本章首先总结分析了基于安全多方计算构造的可验证神经网络安全推理方案存在的问题，及本章的目标与贡献；然后对方案进行了概述；随后给出了可验证同态加密方案以及VHENN 方案的详细设计；最后通过安全性分析与实验评估证明了本方案的安全性及性能。

第六章为总结与展望，总结分析了本文研究内容，并对未来工作进行了展望。

第 2 章 预备知识

2.1 密码学基础

2.1.1 群、环、域

在抽象代数中，群、环和域是三种基本的代数结构，它们在密码学中有着重要的应用。这些结构通过定义集合上的运算及其满足的特定性质，提供了研究和构建复杂密码算法和协议的基础。

1) 群是由一个非空集合 \mathbb{G} 和一个二元运算 $*$ 组成的代数结构，满足以下四个条件：

- 封闭性：对于 $\forall a, b \in \mathbb{G}$, $a * b \in \mathbb{G}$ ；
- 结合性：对于 $\forall a, b, c \in \mathbb{G}$, $(a * b) * c = a * (b * c)$ ；
- 单位元： $\exists e \in \mathbb{G}$, 对于 $\forall a \in \mathbb{G}$ 有 $e * a = a * e = a$ ；
- 逆元：对于 $\forall a \in \mathbb{G}$, $\exists b \in \mathbb{G}$, 使得 $a * b = b * a = e$ 。

其中，阿贝尔群在满足上述条件之外，还满足交换律，即：

- 交换律：对于 $\forall a, b \in \mathbb{G}$, 有 $a * b = b * a$ 。

群结构在密码学中的代表性应用包括：1) 基于椭圆曲线的密码学方案，椭圆曲线上的点集在点加法运算下形成一个群，利用这个群的数学性质实现安全且高效的公钥加密。2) 基于离散对数问题的密码学方案，基于群的离散对数问题是许多密码协议的基础，如 Diffie-Hellman 密钥交换和 ElGamal 加密等。

2) 环是由一个集合 R 和两个二元运算（加法 $+$ 和乘法 \cdot ）组成的代数结构，在群的性质之上，还需满足下述条件：

- 加法构成阿贝尔群，在群的性质之外，还需满足交换性：对于 $\forall a, b \in R$, 有 $a + b = b + a$ ；
- 乘法结合性：对于 $\forall a, b, c \in R$, 有 $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ ；
- 乘法封闭性：对于 $\forall a, b \in R$, 有 $a \cdot b \in R$ ；
- 分配律：对于 $\forall a, b, c \in R$, 有 $a \cdot (b + c) = a \cdot b + a \cdot c$, $(a + b) \cdot c = a \cdot c + b \cdot c$ 。

全同态加密是环结构最重要的应用之一，其利用环上的运算来支持任意次多项式计算，实现对加密数据的直接运算。多项式环的运算可以通过快速傅里叶变换等算法进行高效实现，这在实践中显著提高了全同态加密的性能。此外，环上的理想和模空间的概念在部分同态加密方案中起到了关键作用。

3) 域是由一个集合 F 和两个二元运算（加法 $+$ 和乘法 \cdot ）组成的代数结构，

在群的性质之上，还需满足下述条件：

- F 在加法下构成一个阿贝尔群；
- $F \setminus \{0\}$ 乘法构成阿贝尔群；
- 满足分配律。

有限域理论提供了丰富的数学工具，如多项式代数、矩阵运算等，这些工具可以用于构造和分析密码算法。主要应用包括：1) Paillier^[124]加法同态加密方案，利用有限域的结构进行加法和乘法运算。2) 安全多方计算中常用的密码原语如 Shamir 秘密共享^[125]方案基于有限域上的拉格朗日插值多项式实现秘密分发和重构。

2.1.2 双线性映射

设 $\mathbb{BG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p, g, h)$ 为双线性群，其中 \mathbb{G}_1 , \mathbb{G}_2 和 \mathbb{G}_T 是素数阶为 p 的椭圆曲线群， g 和 h 分别为 \mathbb{G}_1 和 \mathbb{G}_2 的生成元。定义双线性映射 e 为 $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, $\forall a, b \in \mathbb{Z}_p$ ，其满足双线性： $e(ag, bh) = ab \cdot e(g, h)$ ，对称性： $e(a, b) = e(b, a)$ ，非退化性。在后续章节，将使用 $[a]_1$ 表示 $a \cdot g$, $[b]_2$ 表示 $b \cdot h$, $[a]_1 \cdot [b]_2$ 表示 $e(ag, bh)$ 。

2.1.3 敌手模型

在密码学算法和协议中，由于安全性通常是相对于一种特定的敌手模型来定义的，因此需要假设参与实体的敌手模型。通过明确定义敌手的能力和行为，可以更清晰地描述协议在面对不同类型攻击时的安全性。敌手模型通常包括主动敌手模型（也称恶意模型）和被动敌手模型（也称半诚实模型），这些模型是对敌手行为的形式化描述。一个可以抵抗主动攻击模型的方案在保证机密性和正确性方面通常优于可以抵抗被动攻击模型的方案或协议。

主动敌手模型：在主动敌手模型中，攻击者被假定为具有最大攻击能力的恶意实体。这种模型下，敌手可以采取主动的攻击行为，包括但不限于提供虚假信息、修改或伪造消息、违反协议规定的步骤、试图中断或干扰协议的正常执行等。主动敌手的目标是尽可能地破坏协议的安全性、机密性或正确性。

被动敌手模型：被动敌手模型描述了攻击者的一种行为，该攻击者只能被动地监听和观察协议的执行，而不能主动地修改消息、违反规定或采取其他主动攻击行为。被动敌手只能被动地收集信息，并试图通过协议执行中的信息来获取关于系统和通信参与方的有关隐私信息。虽然被动敌手有能力分析信息，但其攻击能力受限于观察和分析，但不会背离预期的协议。

合谋攻击：合谋攻击是一种独立于主动和被动敌手模型的特殊攻击，主要

存在于分布式或多方合作的场景。多个参与方可以联合恢复秘密，而在假设不合谋的前提下，这些参与方无法获取该秘密。在安全多方计算中，合谋攻击模型用于确保即使部分参与者联合，仍无法获得其他参与者的私密信息。MPC 的安全协议通常设计为在一定数量的合谋参与者情况下仍然保持安全。

2.2 同态加密

同态加密允许在加密数据上直接执行特定的操作，得到的结果解密后与采用相同操作处理未加密的数据得到的结果一致。因此可以在数据保持加密状态下完成计算，从而保护数据隐私^[126]。

同态加密可以根据支持的运算类型和数量进行分类：

1) 部分同态加密 (Partial Homomorphic Encryption, PHE) 仅支持一种运算，如加法或乘法。常见的乘法同态加密算法如 RSA 和 ElGamal，常见的加法同态加密算法如 Paillier。

2) 全同态加密 (Fully Homomorphic Encryption, FHE) 允许在密文上进行任意次数的加法和乘法操作，同时保证解密后的结果与在明文上直接操作的结果一致。TFHE (Fast Fully Homomorphic Encryption) 是一种常见的全同态加密方案，在处理布尔电路和门级操作上表现优异。其支持快速 Bootstrapping 的特点使其在构造全同态加密方案时占据了一定的优势。

3) 有限级数同态加密 (Leveled Homomorphic Encryption, LHE) 在加密过程中通过设定初始参数来指定最大计算深度，即允许的最大同态加法和乘法次数的组合。在这个预设深度内，LHE 方案能够确保噪声的累积在可控范围内，从而保证解密结果的正确性。LHE 方案可以通过 bootstrapping 技术转换为 FHE 方案，但为了保证效率，若可以满足计算需求，LHE 仍然是应用更加广泛的同态加密方案。LHE 主要的代表方案包括 BGV (Brakerski-Gentry-Vaikuntanathan)、CKKS (Cheon-Kim-Kim-Song) 等。

2.2.1 多密钥同态加密

第三章方案采用 CKKS (Cheon-Kim-Kim-Song) 同态加密方案的变体，多密钥 CKKS 方案 (MK-CKKS)^[102]。假设有 n 方参与到 MK-CKKS 方案中，每个参与方持有一对公私钥 (pk_i, sk_i) 。首先，各参与方利用持有的公钥 pk_i 按照底层 CKKS 方案对明文进行加密。在执行同态操作前，通过一个公共的预处理程序将与单一参与方关联的密文转换为与 n 方相关联的统一密文。解密操作由 n 个参与方的私钥 sk_i 合作完成。MK-CKKS 包括以下算法：

1) $pp \leftarrow Setup(1^\lambda)$ ：该算法以安全参数 λ 为输入，输出是一个公共参数 pp 。

- 2) $(pk_i, sk_i) \leftarrow KeyGen(pp)$: 该算法为密钥生成算法, 以公共参数 pp 为输入, 输出是 n 个密钥对 (pk_i, sk_i) , $i = 1, 2, \dots, n$ 。
- 3) $ek_i \leftarrow EvkGen(sk_i)$: 该算法为评估密钥生成算法, 以每个参与方的私钥 sk_i 为输入, 输出为用于重线性化的评估密钥 ek_i 。
- 4) $ct_i \leftarrow Enc(pt_i, pk_i, pp)$: 该算法为加密算法, 对于每个参与方 i , 输入为明文 pt_i , 公钥 pk_i 以及公共参数 pp , 输出为密文 ct_i , 该密文仅与参与方 i 相关联。
- 5) $\bar{ct} \leftarrow PreP(ct_i, pk_0, \dots, pk_n)$: 该算法将仅与参与方 i 相关联的密文 ct_i 进行统一处理, 输入为 ct_i 和各参与方的公钥 pk_1, \dots, pk_n , 输出为与 n 方相关联的密文 \bar{ct} 。
- 6) $pt \leftarrow Dec(\bar{ct}, sk_1, \dots, sk_n)$: 该算法为解密算法, 输入为密文 \bar{ct} 和各参与方的私钥 sk_1, \dots, sk_n , 输出为明文 pt 。
- 7) $pt \leftarrow Dec'(\bar{ct}, P_{ct_i}, sk_j)$: 该算法为分布式解密算法, P_{ct_i} 是参与方 i 对密文 \bar{ct} 中对应 i 的组分进行解密得到的明文组分。解密授权方 j 首先将 \bar{ct} 中对应的组分发送给参与方 i 得到明文组分 P_{ct_i} 。该算法输入为 \bar{ct} , P_{ct_i} 和 sk_j , 输出为明文 pt 。
- 8) $\bar{ct} \leftarrow Add(\bar{ct}_1, \bar{ct}_2)$: 该算法为同态加法, 输入为两个密文 \bar{ct}_1, \bar{ct}_2 , 输出为密文 \bar{ct} , \bar{ct} 对应的明文等于 \bar{ct}_1, \bar{ct}_2 对应的明文之和。
- 9) $\bar{ct} \leftarrow Mult_{pt}(\bar{ct}_1, pt_2)$: 该算法为同态明文与密文乘法, 输入为一个密文 \bar{ct}_1 和一个明文 pt_2 , 输出为密文 \bar{ct} , \bar{ct} 对应的明文等于 \bar{ct}_1 对应的明文与 pt_2 之积。
- 10) $\bar{ct}'' \leftarrow Mult_{ct}(\bar{ct}_1, \bar{ct}_2)$: 该算法为同态乘法, 输入为两个密文 \bar{ct}_1, \bar{ct}_2 , 输出为密文 \bar{ct}'' 。
- 11) $\bar{ct}' \leftarrow Relin(\bar{ct}'', \{ek_i, pk_i\}_{1 \leq i \leq n})$: 该算法为重线性化算法, 对密文 \bar{ct}'' 进行重线性化处理, 为了消除了同态乘法操作产生的非线性条目, 以确保解密的正确性。该算法的输入为密文 \bar{ct}'' 以及各方的评估密钥和公钥 $\{ek_i, pk_i\}_{1 \leq i \leq n}$, 输出为重线性化后的密文 \bar{ct}' 。由于所有同态乘法操作均需要重线性化, 因此在后续该算法默认包含在同态乘法算法中。
- 12) $\bar{ct} \leftarrow Rescale(\bar{ct}')$: 该算法为重缩放算法, 它解决了同态乘法操作引起的密文大小增加和误差增大两个问题。重缩放算法输入为密文 \bar{ct}' , 输出为缩放后的密文 \bar{ct} 。该算法在同态操作后按需使用, 默认包含同态操作算法中。

2.2.2 BGV 同态加密

目前神经网络加密推理方案中，通常采用 BGV (Brakerski-Gentry-Vaikuntanathan)^[127]或 CKKS (Cheon-Kim-Kim-Song)^[128]作为底层同态加密方案。虽然 CKKS 支持浮点数运算，在神经网络中的应用更为广泛。但由于 zk-SNARKs 协议通常支持定点数运算，二者结合可能会存在较多困难。因此，第五章方案选择 BGV 作为底层同态加密方案。

BGV 是一种基于环多项式构造的有限级数同态加密方案，可以通过 bootstrapping 技术转换为全同态加密方案，常被应用于神经网络加密推理中。在本方案中，采用基于环上误差学习 (Ring Learning with Errors) 的 BGV 方案， λ 为选定的安全参数， $R = R(\lambda)$ 为环，一般采用 $R = \mathbb{Z}[X]/(X^d + 1)$ ，其中 $d = d(\lambda)$ 通常为 2 的幂。 $R_q = R/qR$ ，其中 $q = q(\lambda)$ 是一个奇模数， qR 由 q 的倍数构成的理想， R_q 表示 R 通过模除理想得到的商环。 $\chi = \chi(\lambda)$ 定义为环 R 上的一个噪音分布。算法的具体描述如下：

- 1) $Setup(1^\lambda, 1^L) \rightarrow (params)$ ：以安全参数 λ 和级数 L 为输入，输出为每一级的参数 $params$ 。选择一个 μ 比特的模数 q_0 ，对于 $j=L \rightarrow 0$ ，选择大小为 $(j+1) \cdot \mu$ 比特的模数 q_j ，以得到阶梯式的模数，即从 q_L 到 q_0 。最后得到 $params = (q_L, \dots, q_0, d, \chi)$ ， $params_j = (q_j, d, \chi)$ 。
- 2) $KeyGen(params_j) \rightarrow (pk_j, sk_j, sk_j^+, sk_j^-, \tau_{sk_j^+ \rightarrow sk_j^-})$ ：对于 $j=L \rightarrow 0$ ，采用参数 $params_j$ 生成对应层级 j 的密钥。然后生成如下密钥：
 - ① 选择 $s_j \leftarrow \chi$ ，令 $sk_j = (1, s_j) \in R_{q_j}^2$ 。选择 $a_j \leftarrow R_{q_j}$ ， $e \leftarrow \chi$ ，整数 t ，计算 $b_j \leftarrow a_j \cdot s_j + te$ ， $pk_j = (b_j, a_j)$ 。 sk_j 为密文在 j 层级的原始密钥。
 - ② 计算 $sk_j^+ = sk_j \otimes sk_j \in R_{q_j}^{[2]}$ ，即 sk_j^+ 为 sk_j 的张量积。 sk_j^+ 为对应 sk_j 的两个密文执行乘法运算后，得到的密文对应的密钥。
 - ③ $\tau_{sk_j^+ \rightarrow sk_{j-1}} \leftarrow SwitchKeyGen(sk_j^+, sk_{j-1})$ ，在执行密文乘法后，密文和密钥扩张为对应的张量积，为了保持密文和密钥的规模，需要采用密钥交换方法，将密文和对应的密钥转换为降低规模后的密文和密钥。 $\tau_{sk_j^+ \rightarrow sk_{j-1}}$ 为用于密钥交换的参数。首先计算 $pk_{j-1} = (a_{j-1}, b_{j-1})$ ， $\tau_{sk_j^+ \rightarrow sk_{j-1}} = pk_{j-1} + Powersof2(sk_j^+)$ ，其中 $Powersof2(sk_j^+) = (sk_j^+, 2 \cdot sk_j^+, \dots, 2^{\lfloor \log q_j \rfloor} \cdot sk_j^+)$ 。
- 3) $Enc(params, pk, PT) \rightarrow CT$ ：对于明文 $PT \in R_t$ ，记 $\mathbf{PT} = (PT, 0)$ 选择 $v \in R_2$ ， $e' = (e_0, e_1) \leftarrow \chi^2$ ， $CT = pk \cdot v + \mathbf{PT} + te'$ 。这里 pk 表示第 L 层级的 pk_L 。
- 4) $Dec(params, sk, CT_j) \rightarrow PT$ ：计算 $PT = [[<CT_j, sk_j>]_{q_j}]$ 。
- 5) $Add(pk, CT_1, CT_2) \rightarrow CT_3$ ：对两个相同层级相同密钥 sk_j 下的密文 CT_1, CT_2 执行加法运算， $CT_3 = CT_1 + CT_2$ 。由于加法不会导致密文规模增大，也不会造成

过大的噪音增加，因此直接在密文各自的分量上执行加法即可。

6) $Mul(pk, CT_1, CT_2) \rightarrow CT_4$: 对两个相同层级相同密钥 sk_j 下的密文 CT_1, CT_2 执行乘法运算，有 $\langle CT_1, sk_j \rangle \cdot \langle CT_2, sk_j \rangle = \langle CT_1 \otimes CT_2, sk_j \otimes sk_j \rangle$ ，该等式是由克罗内克积的性质得到。因此 CT_1 与 CT_2 的乘法可表示为 CT_1 与 CT_2 的张量积，即 $CT_3 = CT_1 \otimes CT_2$ ，且 CT_3 对应密钥 $sk'_j = sk_j \otimes sk_j$ 。为了保持密文和密钥的规模，并降低乘法带来的噪声累加，执行 $CT_4 \leftarrow Refresh(CT_3, \tau_{sk'_j \rightarrow sk_{j-1}}, q_j, q_{j-1})$ 。

7) $Refresh(CT_j, \tau_{sk'_j \rightarrow sk_{j-1}}, q_j, q_{j-1}) \rightarrow CT_{j-1}$: 该算法对一个对应密钥 sk'_j 的密文 CT_j 执行密钥交换与模交换，以恢复密文密钥带来的扩张并降低乘法操作带来的噪音增加。

① $CT'_{j-1} \leftarrow SwitchKey(CT_j, \tau_{sk'_j \rightarrow sk_{j-1}}, q_j)$: $CT'_{j-1} = BitDecomp(CT_j)^T \cdot \tau_{sk'_j \rightarrow sk_{j-1}}$ ，其中 $BitDecomp(CT_j, q_j)$ 表示将 $CT_j \in R_{q_j}^n$ 分解为 $(\mathbf{u}_0, \dots, \mathbf{u}_{\lceil \log q_j \rceil}) \in R_2^{n \lceil \log q_j \rceil}$ ，使得 $CT_j = \sum_{i=0}^{\lfloor \log q_j \rfloor} 2^i \cdot \mathbf{u}_i$ 。 CT'_{j-1} 对应密钥 sk_{j-1} ，模数为 q_j 。

② $CT_{j-1} \leftarrow Scale(CT'_{j-1}, q_j, q_{j-1})$ ，对密文 CT'_{j-1} 的模数进行缩放， CT_{j-1} 对应密钥 sk_{j-1} ，模数为 q_{j-1} 。

执行上述步骤完成模交换与密钥交换，密文恢复原有规模，级数降低，当级数降低为 0 时，便无法继续进行乘法操作。

2.2.3 同态加密 SIMD 操作

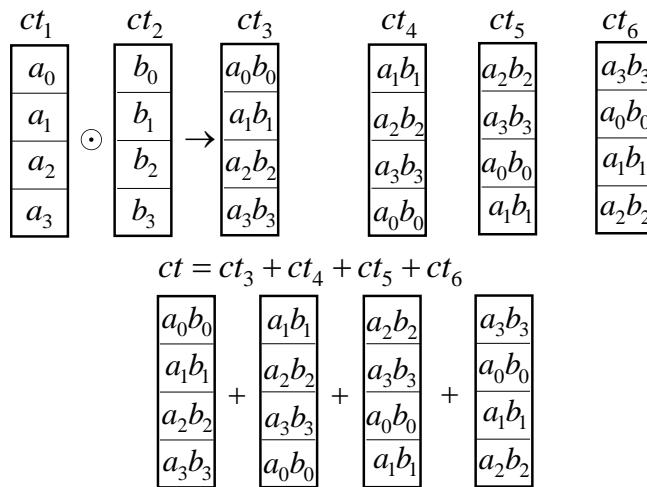


图 2-1 基于 SIMD 的密文向量点积计算过程
Fig.2-1 SIMD-based encrypted vector dot product computation process

CKKS 和 BGV 同态加密方案支持基于 SIMD 的高效批量同态操作，通过将多个明文编码并加密到一个密文中，每个明文对应密文中的一个槽，从而实现在两个不同密文的多个槽之间并行地执行同态操作。由于神经网络的计算中包含大量点积运算，SIMD 技术实现的并行操作可以大大提高神经网络的加密计

算效率。采用 SIMD 技术时，无法在同一密文的不同槽之间执行操作，因此无法采用底层的计算方法执行诸如向量点积一类的操作。基于高斯自同构的旋转（Rotation）技术是最常用的解决方案。基于 SIMD 操作的加密向量点积的计算过程如图 2-1 所示，其中 \odot 表示 ct_1 和 ct_2 间的分量乘积， $ct_4 = Rot_{L,1}(ct_3)$ ， $ct_5 = Rot_{L,2}(ct_3)$ ， $ct_6 = Rot_{L,3}(ct_3)$ ， $Rot_{L/R,i}(ct)$ 表示对密文 ct 的旋转操作， L 表示向左旋转， R 表示向右旋转， i 为旋转的步长。由于旋转的成本相对于同态乘法和加法较高，有必要设计优化方法，以减少实际应用中的旋转操作次数。

2.3 安全多方计算

安全多方计算包括一系列加密协议，可以使各方在不公开各自输入 $x_i (i=1, \dots, M)$ 的情况下共同计算一个函数 $f^{[129]}$ 。若将 MPC 协议表示为 Π ，其中 N 个互不信任的计算方在 MPC 协议下共同计算函数 $y = \Pi(f(x_1, x_2, \dots, x_M))$ 。如果参与方无法得到除函数的输出 y 以外的任何信息，则认为协议 Π 是安全的。对安全多方计算的研究可追溯到 1982 年姚期智提出的百万富翁问题^[130]，这个问题激发了对 MPC 的研究，最终发展成了一个广泛应用于数据隐私保护的领域。MPC 并非一个单一的技术，而是由多种密码学方案构成的协议，主要采用到的技术如下：

1) 秘密共享^[125] (Secret Sharing, SS): 秘密共享是 MPC 中的核心技术之一，用于将一个秘密分割成若干份，并分配给不同的参与者。只有当达到预定数量的参与者联合其份额时，才能恢复原始秘密。在 MPC 协议中，常用的秘密共享方案包括 Shamir 秘密共享、算术秘密共享、布尔秘密共享。其中 Shamir 秘密共享和算术秘密共享主要用于线性运算，如加法和乘法。布尔秘密共享用于布尔电路评估、逻辑运算和条件判断等计算任务中，可以作为算术秘密共享的补充，用于 MPC 中的非线性运算。

2) 混淆电路^[131] (Garbled Circuits, GC): 混淆电路允许两个或多个参与方安全地计算一个布尔电路的输出，而不暴露各自的输入。其核心思想是将布尔电路的每个逻辑门，如 AND、OR、XOR、NOT 等，进行混淆，使得参与方无法直接知道电路内部的中间值，但仍然可以正确地计算出最终输出。混淆电路通常与秘密共享方案混合使用，以实现对线性和非线性操作的高效运算。

3) 不经意传输^[132] (Oblivious Transfer, OT): 不经意协议确保在传输过程中，发送方不知道接收方选择了哪些数据，而接收方只获得所选择的数据且无法获取其他数据。OT 协议通常被用于秘密共享等方案的构造，如在部分算术秘密共享中，为了提高计算效率，部分资源密集型且与输入数据无关的操作可

以转移到离线阶段执行。在离线阶段，需要通过 OT 协议或同态加密等方法生成 Beaver 乘法三元组，以为后续在线阶段提供必要的辅助。

MPC 协议的构造，通常由多个密码学技术组合，除了上述常用技术，还包括零知识证明、消息认证码、向量不经意线性评估等技术。如目前广泛应用于隐私保护机器学习领域的安全多方计算协议 ABY^[69], ABY3^[56]等，通过算术秘密共享、布尔秘密共享、混淆电路之间的高效转换，来实现复杂操作的高效计算。SPDZ 协议则结合零知识证明、消息认证码等技术，实现具有恶意安全模型的 MPC 协议^[133]。

2.3.1 算术秘密共享

基于算术秘密共享的安全多方计算协议通常分为在线阶段和离线阶段，部分资源密集型且与输入数据无关的操作可以转移到离线阶段执行。在离线阶段，通过同态加密等方法生成 Beaver 乘法三元组。在在线阶段， N 个参与方合作执行包括加法和乘法的算术运算。假设有 M 个分布在有限域 F_p 的机密数据 $x_i (i=1, \dots, M)$ 。 x_i 的 N 个秘密份额 $(\langle x_i \rangle_1, \dots, \langle x_i \rangle_N)$ 由 N 个参与方持有，其中 $x_i = \langle x_i \rangle_1 + \dots + \langle x_i \rangle_N$ 。每个参与方 i 以交互的方式计算 $\langle y \rangle_i = f(\langle x_1 \rangle_i, \dots, \langle x_M \rangle_i)$ ，其中可能包括两个共享值间的加法、一个常数与一个共享值间的乘法，以及两个共享值间的乘法。

对于加法运算 $y = \sum_{j=1}^M x_j$ ，持有 $\langle x_j \rangle_i$ 的每个参与方 i 需要计算 $\langle y \rangle_i = \sum_{j=1}^M \langle x_j \rangle_i$ 。对于操作 $y = ax$ （其中 a 是常数， x 是机密数据），持有 a 和 $\langle x \rangle_i$ 的每个参与方 i 需要计算 $\langle y \rangle_i = a \langle x \rangle_i$ 。上述两个操作只需要参与方在本地对持有的秘密份额进行计算，不需要参与方之间的交互，也不需要额外的参数辅助计算。最后，可以通过计算 $y = \sum_{i=1}^N \langle y \rangle_i$ 来恢复 y 。

对于乘法运算 $y = \prod_{j=1}^M x_j$ ，参与方显然无法简单地通过计算 $\langle y \rangle_i = \prod_{j=1}^M \langle x_j \rangle_i$ 得到 $\langle y \rangle_i$ 。两个共享值间的乘法运算需要多方交互实现，并且每次乘法运算都会消耗一个在离线阶段生成的 Beaver 乘法三元组。

2.4 零知识证明

零知识证明^[134] (Zero-Knowledge Proof, ZKP) 允许证明者向验证者证明某个声明的真实性，而不泄露任何关于声明的其他信息。一个零知识证明协议满足以下三个性质：

1) 完备性 (Completeness): 如果声明是真实的，诚实的证明者能够说服验证者，验证者会接受证明。

2) 可靠性 (Soundness): 如果声明是虚假的，任何企图欺骗的证明者都无

法说服诚实的验证者，验证者会拒绝证明。

3) 零知识性 (Zero-Knowledge): 如果声明是真实的，验证者在接受证明的过程中，除了声明的真实性外，无法获得任何关于证明过程的其他信息。

零知识证明主要分为交互式零知识证明和非交互式零知识证明，在交互式零知识证明中，证明者和验证者通过一系列的交互步骤进行证明过程；在非交互式零知识证明中，证明者只需向验证者发送一个证明，而不需要进一步的交互。**zk-SNARKs** 是非交互式零知识证明中的典型协议，具有简洁性、非交互性和高效性。**zk-SNARKs** 的核心思想是通过一系列数学变换，将原本复杂的计算任务转化为简洁的多项式计算和验证。**zk-SNARKs** 协议主要包括如下步骤：

- 1) 设置阶段：生成公共参数，用于证明和验证过程。
- 2) 电路构造：将要证明的语句转化为一个算术电路。算术电路由加法门和乘法门组成，形成一个约束系统。每个约束可以表示为多项式等式。
- 3) 证明生成：证明者使用公共参数、公开输入和证明者的私有输入生成一个证明，证明其私有输入满足步骤 2) 中构造的约束系统，而不泄露其私有输入的隐私信息。
- 4) 验证：验证者使用验证密钥验证步骤 3) 中生成证明的有效性。

在零知识证明中，通常结合承诺协议使用，称为 Commit-and-Prove 零知识证明协议。承诺协议的绑定性可以保证一旦证明者作出承诺，便无法更改其承诺的内容，从而保证证明的真实性。验证者可以通过承诺方案验证证明者的承诺是否与揭示的内容一致，确保证明过程的可靠性。

2.4.1 Groth16 协议

Groth16 是一种 **zk-SNARK** 协议，因其在证明大小和验证时间方面的优势被广泛应用。在 **Groth16** 协议中，证明者会对一个关系 R 构造一个证明 π ，以证明存在满足条件的声明 (statement) x 和见证 (witness) w 使得 $R(x, w) = 1$ 。验证者可以以 $O(|R|)$ 的计算效率检查该证明是否通过验证，无需获取任何关于见证 w 的信息。关系 R 以二次算术程序(Quadratic Arithmetic Programs, QAP)的形式表示，QAP 则是由一阶约束系统(Rank 1 Constraint System, R1CS)转换而来。**R1CS** 通常用于表示算术电路中变量之间的相互关系，而算术电路则是由需要证明的实例转换而来的。在此，我们举例说明将一个算术电路转换为 **R1CS**，再转换为 **QAP** 的过程。

如图 2-2 所示，一个算术电路 C 由声明 $x = (x_1, x_2, x_3)$ ，见证 $w = (w_1, w_2, w_3, w_4)$ 以及电路门（包括乘法门和加法门）组成。证明者的目的是证明他拥有一个满足给定算术电路的见证 w 使其满足该算术电路 C 。为了方便表示，设

$\mathbf{a} = (a_0, a_1, \dots, a_n) = (1, \mathbf{x}, \mathbf{w})$, U, V, W 是三个 $m \times (n+1)$ 维矩阵, R1CS 可以用下列等式表示:

$$(U \cdot \mathbf{a}) \circ (V \cdot \mathbf{a}) = W \cdot \mathbf{a} \quad (2-1)$$

其中 m 是约束数量, $n = |\mathbf{a}| - 1$ 。

选择 (r_1, r_2, \dots, r_m) 作为拉格朗日基数, 对于矩阵 U, V, W 中的每个元素, 设置 $U_{i,j} = u_j(r_i)$, $V_{i,j} = v_j(r_i)$, $W_{i,j} = w_j(r_i)$, 对于 $i=1, 2, \dots, m$, 满足以下 m 个等式:

$$\sum_{j=0}^n a_j u_j(r_i) \cdot \sum_{j=0}^n a_j v_j(r_i) = \sum_{j=0}^n a_j w_j(r_i) \quad (2-2)$$

因此, 矩阵的每一列 j 都可以通过拉格朗日插值转换为三个阶为 $m-1$ 的 QAP 多项式 $u_j(X), v_j(X), w_j(X)$ 。我们定义一个阶为 m 的目标多项式 $t(X) = \prod_{i=1}^m (X - r_i)$, 使得以下等式成立:

$$\sum_{j=0}^n a_j u_j(X) \cdot \sum_{j=0}^n a_j v_j(X) - \sum_{j=0}^n a_j w_j(X) = h(X) t(X) \quad (2-3)$$

其中, $h(X)$ 为商多项式。

最后, 对于所有的 $(x, w) \in R$, 也就是, 所有满足电路 C 的声明 x 和见证 w , QAP 关系 R 可以表示为:

$$R = (\text{aux}, \{u_j(X), v_j(X), w_j(X)\}_{j=0}^n, t(X)) \quad (2-4)$$

其中 aux 为一些额外的参数, 如协议基于的椭圆曲线群等信息。

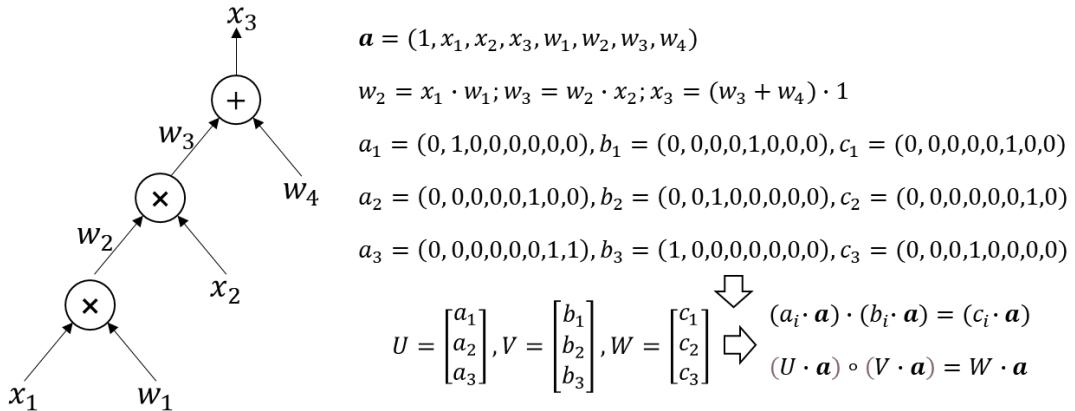


图 2-2 算术电路到 R1CS 的转换
Fig.2-2 The conversion of an arithmetic circuit into R1CS

Groth16 协议具体算法描述如下:

- $ZK_Setup(1^\lambda, R) \rightarrow CRS$: 该算法以安全参数 λ 和关系 R 为输入, 输出为公共参考字符串 (Common Reference String, CRS)。CRS 是一组对证明者和验证者均公开的值, 这些值与 QAP 关系 R 相关联。每个 CRS

可以用于同一个关系 R 的多个证明生成过程。

- $\text{Prove}(R, \text{CRS}, x, w) \rightarrow \pi$: 该算法以关系 R 、CRS、声明 x ，和见证 w 为输入，生成证明对 $(x, w) \in R$ 的证明。
- $\text{Verify}(R, \text{CRS}, x, \pi) \rightarrow \{0,1\}$: 该算法以关系 R 、CRS、声明 x ，和证明 π 为输入，如果验证成功，则输出 1；否则，输出 0。

Groth16 证明系统具有以下特性：

- 完备性 (Completeness): 若一个诚实的证明者持有 $(x, w) \in R$ ，并基于该关系 R 生成了证明 π ，诚实的验证者一定会接受该证明。
- 合理性 (Soundness): 如果一个由计算受限的证明者生成的证明被验证者接受了，则这个证明一定对应了一个真实的陈述，且证明者一定拥有对应的见证。
- 零知识性 (Zero-knowledge): 整个证明系统不会向验证者泄露任何关于见证的信息。

2.4.2 KZG 多项式承诺

KZG (Kate, Zaverucha, and Goldwasser) 多项式承诺^[135]是 zk-SNARKs 协议中常用的加密原语。结合承诺方案的零知识证明协议通常被称为 Commit-and-Prove 零知识证明协议。当与承诺方案相结合时，zk-SNARKs 可以在生成关系时增加对该承诺的约束，以此来保证证明者生成的证明中对指定值的承诺。证明者可以利用 KZG 多项式承诺协议对向量中的元素进行批量承诺，因此其对具有大量参数的神经网络具有较高的适用性。KZG 多项式承诺协议算法描述如下：

- $\text{KZG_Setup}(1^\lambda) \rightarrow \text{SRS}$: 该算法以安全参数 λ 作为输入，输出为结构化参考字符串 (Structured Reference String, SRS)。KZG 承诺中的 SRS 是一种特殊的 CRS。不同于 Groth16 协议中的 CRS，SRS 是通用的，与所承诺的多项式无关。
- $\text{Commit}(\text{SRS}, W) \rightarrow cm_W$: 该算法以 SRS 和需要承诺的数据 W 作为输入，输出为对 W 的承诺 cm_W 。
- $\text{Com_Open}(\text{SRS}, cm_W, W) \rightarrow \{0,1\}$: 该算法以 SRS、承诺 cm_W 和承诺的数据 W 作为输入，以揭示承诺。若输出为 1，则证明该承诺为 W 的承诺，若输出为 0，则验证失败。

在 zk-SNARKs 协议中结合承诺时，由于承诺的计算过程是以约束的形式添加到 zk-SNARKs 的关系中，因此不需要执行承诺揭示过程，即 $\text{Com_Open}()$ 。

因此，KZG 承诺并不违反 zk-SNARKs 的零知识属性。具体地，在零知识证明中增加承诺协议能够保证证明者在证明其拥有特定数据的同时，不泄露其机密数据的隐私信息。公开的承诺值和其他公开值被验证者用于验证，且验证者无法直接访问该承诺值对应的机密数据。由于 KZG 承诺协议的计算绑定和完美隐藏的特性，证明者无法修改已承诺的数据，而验证者也无法获得与承诺数据有关的任何敏感信息。

2.4.3 二次环程序

一个基于有限交换环 R 上的 QRP 关系 Q 包括三组多项式：
 $U = \{u_k(X) : k \in [0, m]\}$, $V = \{v_k(X) : k \in [0, m]\}$, $W = \{w_k(X) : k \in [0, m]\}$, 以及一个目标多项式 $t(X)$, 这些多项式均属于 $R[X]$, 也就是, 这些多项式的系数均属于环 R , $R[X]$ 代表环多项式。假设 \mathcal{C} 是一个具有 n 个输入和 n' 个输出的基于环的算术电路, 如果满足下列条件, 则 Q 是电路 \mathcal{C} 的 QRP:

存在一组有效的输入/输出变量赋值 $a_1, \dots, a_n, a_{m-n'+1}, \dots, a_m \in R^{n+n'}$, 使得存在 $a_{n+1}, \dots, a_{m-n'} \in R^{m-n-n'}$, 满足 $t(X)$ 整除 $p(X)$, 其中:

$$p(X) = U(X) \cdot V(X) - W(X) \quad (2-5)$$

$$U(X) = (u_0(X) + \sum_{k=1}^m a_k \cdot u_k(X)) \quad (2-6)$$

$$V(X) = (v_0(X) + \sum_{k=1}^m a_k \cdot v_k(X)) \quad (2-7)$$

$$W(X) = (w_0(X) + \sum_{k=1}^m a_k \cdot w_k(X)) \quad (2-8)$$

我们定义 Q 的大小为 m , 目标多项式 $t(X)$ 的度为 Q 的度。上述多项式 $U(X), V(X), W(X) \in R[X]$ 以及对应的变量赋值组成电路 \mathcal{C} 的 QRP 关系。

为了构造电路 \mathcal{C} 的 QRP, 首先选择一个特殊集 A , 其中的元素两两之间的差值都是可逆的。为每个乘法门 $g \in \mathcal{C}$ 选择元素 $r_g \in A$, 并定义目标多项式 $t(X) = \prod_{g \in \mathcal{C}} (X - r_g)$ 。根据中国剩余定理 (Chinese Remainder Theorem, CRT), 多项式 $u_k(X), v_k(X), w_k(X)$ 可以通过对 $r_g \in A$ 进行插值得到, 这与 2.4.1 节中描述的 QAP 构造过程相似。对于定义为 $I_g = x - r_g$ 的 $I_1, \dots, I_{\deg(t(X))}$, 由于 A 是一个特殊集, $I_1, \dots, I_{\deg(t(X))}$ 必然是互素的。对于 $p(X) \equiv p(r_g) \pmod{(X - r_g)}$, 满足:

$$\phi: R[X]/t(X) \simeq R[X]/I_1 \times \dots \times R[X]/I_{\deg(t(X))} \quad (2-9)$$

$$p(X) \rightarrow (p(r_1), \dots, p(r_{\deg(t(X))})) \quad (2-10)$$

上述同构表明, 当且仅当 $p(r_g) = 0$ 时, 才可以满足 $t(X)$ 整除 $p(X)$, 记 $t(X)$ 除 $p(X)$ 得到的商多项式为 $h(X)$, 则满足 $p(X) = t(X)h(X)$ 。

2.4.4 环上 zk-SNARKs 协议

Groth16 是一种基于椭圆曲线群构造的 zk-SNARKs 协议，其在证明大小和验证时间方面具有较大的优势，能够很好的适用于神经网络推理中资源受限的用户。本文第五章方案采用了 Rinocchio^[123]中构造的具有 Groth16 类似结构的环上 zk-SNARKs 协议（后续称为 R-Groth16）。首先将 QRP 多项式由在一个秘密点处计算的多项式编码表示，该编码在基于环的计算中具有加同态特性。我们将编码方案表示为算法 (*Gen*, *Encode*)，具体过程如下：

- 1) $\text{Gen}(1^\lambda) \rightarrow (pk, sk)$ ：该算法为密钥生成算法，以安全参数 λ 为输入，输出为公钥 pk 和私钥 sk 。
- 2) $\text{Encode}(a, sk) \rightarrow E(a)$ ：该算法为一种概率编码算法，将一个环元素 $a \in R$ 映射到编码空间 S ，使得集合 $\{\{E(a)\} : a \in R\}$ 成为编码空间 S 的一个划分。

假设一个环 R 上的电路 C 具有 m 个线和 n 个乘法门，电路 C 的 QRP 关系表示为 $Q = (\{u_k(X), v_k(X), w_k(X)\}_{k=0}^m, t(X))$ 。使 $I_s = 1, 2, \dots, l$ 与电路的声明值相对应， $I_w = l+1, \dots, m$ 与电路的见证值相对应。 $(\text{Gen}, \text{Encode})$ 表示一个安全编码方案， $A^* \in R^*$ 表示一个特殊集，其中 R^* 为环 R 的单位群，该单位群是由 R 中所有存在乘法逆元的元素组成的集合。R-Groth16 的方案构造如下：

- 1) $\text{Setup}(1^\lambda, Q) \rightarrow (CRS, vk)$ ：该算法由可信第三方执行，以安全参数 λ 和关系 Q 为输入，输出为公共参考字符串 CRS 和验证密钥 vk 。首先由可信第三方执行 $\text{Gen}(1^\lambda)$ 算法，生成公私钥对 (pk, sk) 。随后，随机选择 $\alpha, \beta, \gamma, \delta \leftarrow R^*$ ， $\varepsilon \leftarrow A^*$ ，并计算 CRS ：

$$CRS = \begin{pmatrix} pk, \{E(\varepsilon^j)\}_{j=0}^{n-1}, E(\alpha), E(\beta), \\ \{E(\gamma^{-1}(\beta u_k(\varepsilon) + \alpha v_k(\varepsilon) + w_k(\varepsilon)))\}_{k \in I_s}, \\ \{E(\delta^{-1}(\beta u_k(\varepsilon) + \alpha v_k(\varepsilon) + w_k(\varepsilon)))\}_{k \in I_w}, \\ \{E(\delta^{-1}(\varepsilon^j t(\varepsilon)))\}_{j=0}^{n-1} \end{pmatrix} \quad (2-11)$$

$$vk = (sk, CRS, \varepsilon, \gamma, \delta) \quad (2-12)$$

在上述参数中， CRS 用于生成证明，被发送给证明方。其中， $\{E(\gamma^{-1}(\beta u_k(\varepsilon) + \alpha v_k(\varepsilon) + w_k(\varepsilon)))\}_{k \in I_s}$ 在证明生成中作用于与声明相对应的参数， $\{E(\delta^{-1}(\beta u_k(\varepsilon) + \alpha v_k(\varepsilon) + w_k(\varepsilon)))\}_{k \in I_w}$ 在证明生成中作用于与见证相对应的参数。 vk 用于验证，被发送给验证方。

- 2) $\text{Prove}(Q, CRS, x, w) \rightarrow \pi$ ：该算法由证明方执行，以 QRP 关系 $Q = (\{u_k(X), v_k(X), w_k(X)\}_{k=0}^m, t(X))$ ，公共参考字符串 CRS ，声明 $x = (a_1, \dots, a_l)$ ，

见证 $w = (a_{l+1}, \dots, a_m)$ 为输入，输出为与关系 Q 对应的电路的证明 π 。设 $a_0 = 1$ ，
 $u_w(\varepsilon) = \sum_{k=l+1}^m a_k u_k(\varepsilon)$ ，
 $v_w(\varepsilon) = \sum_{k=l+1}^m a_k v_k(\varepsilon)$ ，
 $w_w(\varepsilon) = \sum_{k=l+1}^m a_k w_k(\varepsilon)$ 。证明方进行如下计算，并证明 $\pi = (A, B, C)$ 发送给验证方：

$$A = E(A_u) = E\left(\alpha + \sum_{k=0}^m a_k u_k(\varepsilon)\right) \quad (2-13)$$

$$B = E(B_v) = E\left(\beta + \sum_{k=0}^m a_k v_k(\varepsilon)\right) \quad (2-14)$$

$$C = E(C_w) = E\left(\frac{\beta u_w(\varepsilon) + \alpha v_w(\varepsilon) + w_w(\varepsilon) + h(\varepsilon)t(\varepsilon)}{\delta}\right) \quad (2-15)$$

3) $Verify(Q, vk, x, \pi) \rightarrow \{0, 1\}$ ：该算法由验证方执行，以 QRP 关系 Q ，验证密钥 vk ，声明 x ，证明 π 为输出，如果验证成功，则返回 1。若验证不成功，则返回 0。首先验证方计算 $f_s = (\beta u_s(\varepsilon) + \alpha v_s(\varepsilon) + w_s(\varepsilon)) / \gamma$ ， $F = E(f_s)$ ，其中 $u_s(\varepsilon) = \sum_{k=0}^l a_k u_k(\varepsilon)$ ， $v_s(\varepsilon) = \sum_{k=0}^l a_k v_k(\varepsilon)$ ， $w_s(\varepsilon) = \sum_{k=0}^l a_k w_k(\varepsilon)$ 。随后，验证下列等式是否成立：

$$AB = E(\alpha)E(\beta) + \gamma F + \delta C \quad (2-16)$$

若成立则返回 1，若不成立则返回 0。

2.5 神经网络基础

2.5.1 神经网络基本结构

神经网络的基本结构是受生物神经网络启发而设计的计算模型，由大量互联的节点（也称神经元）组成。这些节点按照一定的层次结构排列，并通过权重连接。神经网络具有多种类型，每种类型在处理不同类型数据和任务时具有独特的优势。前馈神经网络适合基本的分类和回归任务，卷积神经网络在图像处理上表现出色，循环神经网络及其变种擅长处理序列数据，自编码器用于特征提取和数据降维，生成对抗网络用于数据生成，Transformer 在自然语言处理领域引领潮流，图神经网络则适用于图结构数据的分析^[136]。一个典型的神经网络包括输入层、隐藏层和输出层。以下是对神经网络基本结构的详细描述。

1) 输入层：输入层是神经网络的第一层，负责接收外部输入数据。输入层的节点数等于输入数据的特征数量，每个节点对应一个输入特征。主要功能为将输入的数据传递到下一层，在输入层不进行任何计算。

2) 隐藏层：隐藏层位于输入层和输出层之间，负责特征提取和非线性变换。一个神经网络可以有多个隐藏层。可以任意设定节点数，通常通过实验调整，以平衡计算复杂度和模型性能。每个节点接收上一层的输入，进行加权求

和、加偏置等运算，然后通过激活函数输出结果。其中，加权求和计算主要存在于卷积层、全连接层等结构中，包括乘法和加法的线性运算。激活函数诸如 ReLU (Rectified Linear Unit), Sigmoid 主要为指数、比较等非线性运算。

3) 输出层：输出层是神经网络的最后一层，生成最终的输出结果。每个节点对应一个输出特征或类别。节点数取决于具体的任务，如，二分类任务中有一个节点，多分类任务中节点数等于类别数。主要负责接收来自最后一个隐藏层的输出，进行加权求和和激活，生成最终的预测结果。输出层通常将 Sigmoid 函数用于二分类问题，输出值在 0 到 1 之间；将 Softmax 函数用于多分类问题，将输出值归一化为概率分布。

2.5.2 神经网络工作流程

神经网络的工作流程涵盖了从数据准备到模型训练和推理的各个步骤，具体如下：

1) 数据准备：收集训练和测试数据，并对收集到的数据进行预处理，包括数据清洗、标准化/归一化、数据分割等。

2) 网络初始化：根据任务需求，定义神经网络的结构，包括神经网络类型、层的类型、数量和节点数。例如，选择使用全连接层、卷积层、循环层等，并初始化神经网络的权重和偏置。

3) 神经网络训练，主要包括下列步骤：

① 前向传播：前向传播是从输入层到输出层的信号传递过程，计算每一层的输出，主要包括诸如卷积层、全连接层的线性计算以及激活函数的非线性计算。

② 损失计算：计算网络的输出与实际标签之间的差异，使用损失函数衡量。

③ 反向传播：反向传播是从输出层到输入层的梯度计算过程，用于更新权重和偏置。主要包括：计算损失梯度：计算损失函数相对于输出层每个节点的梯度；反向传播梯度：利用链式法则，将梯度从输出层依次传递到隐藏层和输入层，计算各层权重和偏置的梯度。梯度累积：对于每个训练样本的梯度进行累积。

④ 权重更新：使用优化算法根据计算出的梯度更新网络的权重和偏置，并迭代执行损失计算与反向传播，以更新权重。

4) 模型评估：在验证集和测试集上评估模型的性能，使用指标包括准确率、精确率、召回率、F1 得分等。

5) 模型推理：使用训练好的模型对新数据进行推理，生成最终的结果，并应用于实际任务中，如分类、回归、检测等。模型推理过程与训练过程的前向

传播计算相同。

2.5.3 拆分学习

拆分学习可以有效地实现持有不同特征数据集的参与方之间的合作训练，假设数据集样本已对齐，多个数据方持有具有不同特征的数据集，标签方持有对应的数据标签。拆分学习的基本过程如下：

- 1) 初始化：参与方确定模型结构并初始化全局模型 GM 的参数。
- 2) $(BM_s, TM) \leftarrow SplitGM(GM)$ ：选择一个参与方 P ，将全局模型 GM 拆分为子模型，包括一个顶部模型 TM 和若干个底部模型 BM_s 。将子模型分配给相应的各参与方。
- 3) $X_{s,i} \leftarrow Forward_C(BM_i, X_i)$ ：每个持有底部模型的数据方将本地数据 X_i 与底部模型 BM_i 作为输入，并执行前向传播算法。底部模型的输出 $X_{s,i}$ 被发送给标签方。
- 4) $\bar{y} \leftarrow Forward_S(TM, X_{s,i})$ ：标签方汇总所有数据方的输出，将汇总结果作为输入，并执行前向传播算法，得到顶部模型的输出 \bar{y} 。
- 5) $(g, Updated_TM) \leftarrow Back_S(TM, y, \bar{y})$ ：标签方以 \bar{y} 和标签 y 为输入，在顶部模型 TM 上执行反向传播算法，更新顶部模型参数，并将切割层的梯度 g 发送给数据方。
- 6) $Updated_BM_i \leftarrow Back_C(BM_i, g)$ ：每个数据方将梯度 g 作为输入，并利用反向传播算法更新 BM_i 的模型参数。迭代执行步骤 3)-6)，直至模型收敛。
- 7) $(GM) \leftarrow RecoverGM(BM_s, TM)$ ：训练结束后，合并子模型（顶部模型和底部模型）以恢复全局模型。

2.6 本章小结

本章介绍了本文研究内容所涉及的部分密码学基础知识和神经网络基础知识，包括：密码学中的群环域、双线性映射、敌手模型、同态加密、安全多方计算、零知识证明等基础，神经网络的基本结构及工作流程、拆分学习的基本流程等知识。

第3章 基于多密钥同态加密的隐私保护合作训练方案

3.1 问题描述

为了解决神经网络在传统集中式训练中由于收集各方数据造成的隐私泄露及高昂的通信成本等问题，以联邦学习、拆分学习为代表的多方合作训练模式得到广泛应用^[137]。在多方合作学习中，多个数据所有者可以在本地合作训练神经网络模型，而无需将本地数据发送至中心服务器，以保证数据的隐私信息不被泄露，且避免了原始数据传输造成的通信开销。当前，对合作学习的研究主要针对多方持有相同特征数据的场景，如横向联邦学习。对于多方持有不同特征数据的合作学习（我们在下文称之为纵向合作学习）的研究相对较少，且存在较多的挑战。

以如下场景为例：在智慧医疗领域，深度学习技术取得的显著进展，以及大量医疗领域数据的积累，推动了人工智能医生（AI 医生）的研究与应用。通过收集大量医疗数据，对 AI 医生模型进行训练，以处理复杂的医疗诊断等任务，使得 AI 医生在提高医疗效率、准确性和个性化治疗等各方面都展示了出巨大的潜力。由于医疗数据中可能包含大量患者的隐私信息，通过收集数据进行集中式的训练方式面临着隐私泄露威胁，因此需要采用多方合作的方式进行模型训练以实现对患者隐私信息的保护。此外，在采用患者数据训练神经网络模型时，仅使用一类数据（如单一的医疗记录）训练的模型可能会导致模型泛化能力不足，模型存在局限性。使用由不同数据所有者持有的具有不同特征的数据（例如多种医疗数据、消费数据、运动记录、环境信息）共同训练神经网络模型，可以缓解上述问题，提高 AI 医生的整体性能。因此，纵向合作学习，作为一种具有隐私保护的神经网络训练方法，在诸如上述场景中的多种应用中，都具有较大的发展前景。

3.1.1 存在的挑战

已有研究^[4, 18, 29]提出采用拆分学习实现神经网络纵向合作学习。目前，一些研究将拆分学习视为联邦学习的一种特殊实现形式，而另一些研究则认为拆分学习和联邦学习是两种不同的多方合作学习方式。在本文中，采用后者观点将拆分学习视为一种独立于联邦学习的多方合作学习方法。相比于联邦学习，拆分学习的性质使其更适用于构造纵向合作学习方案。首先，联邦学习要求每个参与方对完整的模型进行本地训练，拆分学习则将完整的模型拆分为多个子模型并分发给不同的参与方，参与方仅需对子模型进行本地训练，使得诸如智

慧医疗中的可穿戴设备等资源受限的用户，也能参与到合作学习中。此外，持有不同特征数据的参与方可以在子模型上进行本地训练，降低了构建纵向合作学习的难度。其次，联邦学习通常需要向聚合服务器发送整个模型更新的参数，而在拆分学习中，仅需共享切割层的中间值与更新梯度即可实现模型更新。因此在相似的设置下，相比于联邦学习，拆分学习可以实现更低的通信开销。

然而，在多方合作训练模式中，虽然原始数据保留在参与方本地，攻击者依旧可以通过推理攻击等方式获取本地数据的隐私信息。部分研究表明^[11-13, 18-20, 25, 26, 29, 30]，恶意参与方可以从拆分学习的交互信息中推理出参与方敏感信息，包括训练数据的属性，标签等。在联邦学习中，已有研究通过同态加密、安全多方计算、差分隐私等技术保护梯度、权重等参数实现训练过程的隐私保护。由于拆分学习中不存在联邦学习中的聚合服务器，参与方均需在本地训练相应的子模型，因此难以直接采用安全多方计算实现隐私保护。目前拆分学习中最常用的隐私保护方法是通过差分隐私等方法添加噪声扰动^[27, 29]，但这种方法需要在隐私保护与模型效用之间进行权衡。基于同态加密的隐私保护方法，由于其较高的安全性，在拆分学习的隐私保护中展现出了较高的应用潜力。

3.1.2 设计目标与方法

考虑到上述问题，本章方案主要目标为：1) 基于拆分学习实现持有不同特征数据的多个参与方之间的合作学习；2) 在不影响模型效用的前提下，保护训练数据和数据标签的隐私信息，并在整个训练过程中保证模型的机密性。为满足上述目标，本章提出 SecureSL，一种基于多密钥同态加密的隐私保护纵向合作训练方案。本方案基于拆分学习构造，首先将神经网络模型分割成多个子模型，包括一个顶部模型和数个底部模型，并根据参与方（如终端设备）持有的数据特征进行分配。持有纵向分布数据的参与方（数据方）对各自的底部模型进行本地训练。持有数据标签的参与方（标签方）以其他参与方发送的子模型输出作为输入对顶部模型进行训练。在上述过程中结合同态加密，数据方将底部模型的输出加密后发送给标签方进行后续的训练，从而保证拆分学习在训练过程中的数据、标签以及训练模型的机密性。在拆分学习中采用同态加密实现隐私保护时，持有密钥的数据方可能会与标签方合谋获取并解密其他数据方的输出结果。因此，本方案采用多密钥同态加密（Multi-Key Homomorphic Encryption, MKHE），数据方可以分别使用各自的密钥对底部模型输出结果进行加密，标签方可以对使用不同密钥加密的数据进行同态计算，从而防止数据方与标签方的合谋。此外，为了缓解引入同态加密后普遍存在的弊端-计算效率问题，本方案提出两种基于 SIMD 技术的优化算法，对基于 MKHE 的明文矩阵

和密文向量之间的点积以及密文矩阵间的点积运算进行优化。

3.1.3 本章贡献

本章主要贡献总结如下：

- 1) 提出一种基于拆分学习的神经网络合作训练框架 SecureSL, 从而实现持有不同特征数据的多个参与方之间的纵向合作神经网络训练。
- 2) 采用多密钥同态加密技术, 实现纵向训练过程中的隐私保护, 保证训练数据、标签和模型的机密性, 防止多方合谋。
- 3) 基于 SIMD 操作的特点提出两种点积计算优化方法, 并对训练过程适应性的修改以兼容基于 SIMD 的计算过程, 达到提高 SecureSL 计算效率的目的。

3.2 方案概述

3.2.1 系统模型

如图 3-1 所示, SecureSL 框架中包含端侧和边缘侧, 其中端侧包含 $n(n \geq 1)$ 个数据方群组, 一个持有数据标签的标签方。边缘侧则由若干个边缘节点组成的边缘区块链构成。不同数据方群组持有的数据具有不同的特征 (称为纵向数据集), 群组数量取决于模型训练中包含的纵向数据集数量。每个群体包含若干数据方, 相同群体的数据方持有的数据具有相同的特征。在初始化阶段, 全局模型被拆分为 n 个底部模型 (靠近输入层的部分) 和一个顶部模型 (靠近输出层的部分), 其中拆分层被称为切割层, 指代底部模型的输出层与顶部模型的输入层。 n 个底部模型被分发给 n 个数据方群组, 顶部模型被分发给边缘节点。端侧和边缘侧的训练过程如下:

1) 端侧训练

持有相同特征数据的终端设备被分组为一个数据方群组。每个数据方群组的终端设备在本地训练所持有的底部模型, 并将底部模型的输出加密后发送到边缘节点。在该过程中, 终端设备的数据保存在本地, 只有底部模型加密后的输出被发送到边缘节点。为了抵抗推理攻击和合谋攻击, 本方案采用多密钥同态加密对底部模型的输出数据进行加密, 从而防止数据方与边缘侧节点合谋获取其他数据方的数据并从中获取隐私信息。

2) 边缘侧训练

边缘节点收到数据方的加密输出后, 会将各数据方的数据进行聚合, 并将聚合结果作为顶部模型的输入, 生成加密输出。顶部模型的加密输出随后发送给标签方。边缘节点形成区块链, 通过共识机制确保计算结果的正确性和一致性。在该过程中, 边缘节点无法从加密数据中获取任何敏感信息。

3) 模型更新

标签方通过比较其持有的数据标签和顶部模型的加密输出计算出加密误差。随后，将加密误差返回给边缘节点。边缘节点执行反向传播算法更新顶部模型，并将切割层的更新梯度发送给数据方。同样的，数据方执行反向传播算法更新底部模型。在该过程中，顶部模型和底部模型的权重等参数都是加密的。加密数据只有在所有端侧参与方的合作下才能解密。即使模型参数处于保密状态，边缘节点更新过程的准确性和一致性仍然可以通过边缘区块链保证。由于模型参数是以密文形式进行更新的，因此边缘节点和数据方的子模型均为保密的，从而保证了模型的机密性。

4) 模型推理

训练结束后，模型根据具体要求以分布式的方式持有，或由各方合作解密后由授权方持有。前者要求推理过程同样以分布式加密形式执行，后者训练的模型参数只有被授权方持有，从而确保了训练模型的机密性。

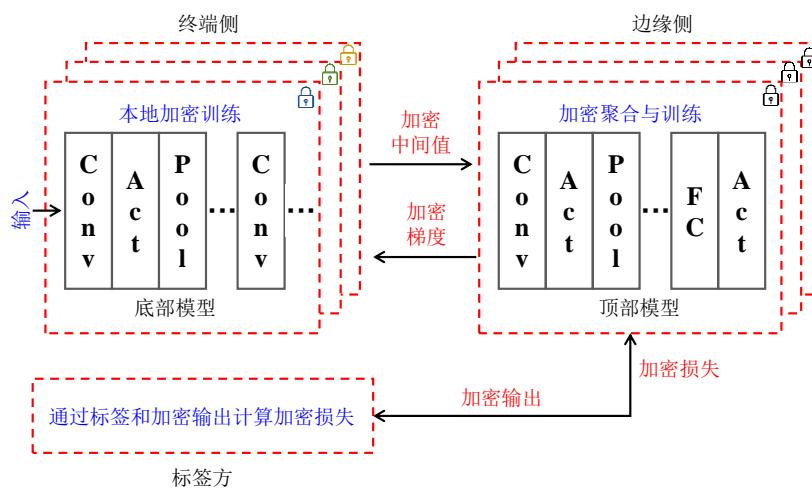


图 3-1 系统模型
Fig.3-1 System model

3.2.2 威胁模型

1) 终端侧威胁模型

本方案为数据方与标签方设定被动敌手模型，也称为半诚实敌手模型。在该模型中，数据方与标签方会遵守既定的协议执行模型训练任务。但他们试图使用属性推理攻击和成员推理攻击等获取其他数据方的隐私信息。数据方会试图采用标签推理攻击来获取标签方的隐私信息。此外，各方可能会与其他参与方或恶意攻击者合谋挖掘训练过程中的可能会泄露隐私的数据。

2) 边缘侧威胁模型

在边缘侧，本方案为边缘节点设定为主动敌手模型。边缘节点不仅会试图窃取训练过程中的隐私信息，还可能会通过主动攻击，破坏数据或模型的完整性和机密性。

3.2.3 SecureSL 方案流程

SecureSL 的方案流程包括初始化、密钥生成、参数加密、前向传播、反向传播、恢复全局模型等六个步骤，如算法 3-1 至 3-6 所示。本节使用的符号说明见表 3-1。

表 3-1 符号说明

Table 3-1 Notations

符号	含义
W_{BM_i}	底部模型 BM_i 的权重或卷积核
W_{TM}	顶部模型 TM 的权重或卷积核
X_i	数据方 D_i 的输入数据样本
$X_{s,i}$	底部模型 BM_i 的输出
$\llbracket y \rrbracket_i$	由数据方 D_i 的公钥加密 y 的密文
$\llbracket y \rrbracket$	对 $\llbracket y \rrbracket_i$ 统一处理后的全局密文
$\llbracket y \rrbracket^i$	密文 $\llbracket y \rrbracket$ 中与数据方 D_i 相对应的分量
$P_{\llbracket y \rrbracket_i}$	明文 y 中与数据方 D_i 相对应的分量
W^U	更新的权重或卷积核
K	授权的最终模型持有者

在初始化阶段，首先选择可信第三方 P，由 P 负责选取安全参数，全局模型初始化，及初始化算法的执行。最终输出公共参数 pp ，底部模型 BM_i 及顶部模型 TM 。由于同一个数据方群组中的终端设备拥有相同的底部模型和相同特征的数据集，为了方便说明，在后续的表述中，将数据方群组内终端设备的本地训练和输出数据聚合统一表示为数据方 D_i 的本地训练。

算法 3-1 初始化

输入：安全参数 λ ，初始化的全局模型 GM ，选择可信第三方 P 执行算法

输出：公共参数 pp ，底部模型 BM_i ，顶部模型 TM

- 1: $pp \leftarrow Setup(1^\lambda)$
 - 2: $(BM_s, TM) \leftarrow SplitGM(GM)$
 - 3: for $i = 0 \rightarrow n$
 - 4: 将 BM_i 发送给 D_i
 - 5: 将 TM 发送给边缘节点
 - 6: end for
-

在密钥生成阶段，分别由边缘节点 D_0 和数据方 D_i 执行算法，以公共参数 pp 为输入，为边缘节点和每个数据方生成公私钥对 (pk_i, sk_i) 与评估密钥 ek_i ，其中 sk_i 对其他用户保密， pk_i 和 ek_i 是公开的。

算法 3-2 密钥生成

输入：公共参数 pp
 输出：公私钥对 (pk_i, sk_i) ，评估密钥 ek_i

```

/* Running on edge nodes D0 */
1: (pk0, sk0) ← KeyGen(pp)
2: ek0 ← EvkGen(sk0)
/* Running on Di */
3: for i = 0 → n
4:     (pki, ski) ← KeyGen(pp)
5:     eki ← EvkGen(ski)
6: end for

```

在参数加密阶段，首先各参与方与边缘节点分别采用其持有的公钥将本地持有的底部模型进行加密，随后对密文进行预处理，得到各底部模型和顶部模型多密钥形式下的加密参数。

算法 3-3 参数加密

输入：底部模型 BM_i 的权重或卷积核 W_{BM_i} ，公钥 pk_i ，公共参数 pp
 输出：加密的模型参数 $\llbracket W_{BM_i} \rrbracket$ 和 $\llbracket W_{TM} \rrbracket$

```

/* Running on Di */
1: for i = 0 → n
2:      $\llbracket W_{BM_i} \rrbracket_i \leftarrow Enc(W_{BM_i}, pk_i, pp)$ 
3:      $\llbracket W_{BM_i} \rrbracket \leftarrow PreP(\llbracket W_{BM_i} \rrbracket_i, pk_0, \dots, pk_n)$ 
4: end for
/* Running on edge nodes D0 */
5:  $\llbracket W_{TM} \rrbracket_0 \leftarrow Enc(W_{TM}, pk_0, pp)$ 
6:  $\llbracket W_{TM} \rrbracket \leftarrow PreP(\llbracket W_{TM} \rrbracket_0, pk_0, \dots, pk_n)$ 

```

在前向传播阶段，首次迭代时，各参与方在模型参数为明文的情况下执行前向传播算法，随后加密底部模型的输出并将其发送给边缘节点。由于反向传播只能得到密文下的梯度更新，从而只能对加密的模型参数进行更新，因此在随后的迭代过程中，各参与方在模型参数为明文的情况下执行前向传播算法，并将底部模型的加密输出发送给边缘节点。边缘节点将各方密文聚合，作为顶部模型的输入执行前向传播算法，并将加密输出发送给标签方。

算法 3-4 前向传播

输入: W_{BM_i} , W_{TM} , $\llbracket W_{BM_i} \rrbracket$ 和 $\llbracket W_{TM} \rrbracket$, 数据方 D_i 的输入数据样本 X_i
 输出: 加密的底部模型和顶部模型输出 $\llbracket X_{s,i} \rrbracket$, $\llbracket y' \rrbracket$

/ Running on D_i */*

- 1: if $iteration=0$ then
- 2: $X_{s,i} \leftarrow Forward_C(W_{BM_i}, X_i)$
- 3: $\llbracket X_{s,i} \rrbracket_i \leftarrow Enc(X_{s,i}, pk_i, pp)$
- 4: $\llbracket X_{s,i} \rrbracket \leftarrow PreP(\llbracket X_{s,i} \rrbracket_i, pk_0, \dots, pk_n)$
- 5: else
- 6: $\llbracket X_{s,i} \rrbracket \leftarrow Forward_C(\llbracket W_{BM_i} \rrbracket, X_i)$
- 7: D_i 将 $\llbracket X_{s,i} \rrbracket$ 发送给边缘节点

/ Running on edge nodes D_0 */*

- 8: $\llbracket X_s \rrbracket \leftarrow Agg_{i=1 \rightarrow n}(\llbracket X_{s,i} \rrbracket)$
- 9: if $k=0$, then
- 10: $\llbracket y' \rrbracket \leftarrow Forward_S(W_{TM}, \llbracket X_s \rrbracket)$
- 11: else
- 12: $\llbracket y' \rrbracket \leftarrow Forward_S(\llbracket W_{TM} \rrbracket, \llbracket X_s \rrbracket)$
- 13: 边缘节点将 $\llbracket y' \rrbracket$ 发送给标签方

在反向传播阶段, 标签方将顶部模型的加密输出发送给各参与方, 各参与方采用私钥计算相应的组份返还给标签方。各参与方的解密组份不会泄露加密输出的隐私信息。标签方使用各解密组份计算输出, 通过误差函数 $l()$ 计算误差, 随后将误差加密发送给边缘节点。边缘节点和数据方执行加密的反向传播过程。

算法 3-5 反向传播

输入: 加密的顶部模型输出 $\llbracket y' \rrbracket$ 和标签 y
 输出: 误差 $\llbracket loss \rrbracket$, 梯度和更新后的模型参数

/ Running on the label party */*

- 1: for $i=1 \rightarrow n$
- 2: 标签方将 $\llbracket y' \rrbracket^i$ 发送给 D_i , D_i 将对应的部分解密得到 $P_{\llbracket y' \rrbracket^i}$
- 3: end for
- 4: $y' \leftarrow Dec'(y', P_{\llbracket y' \rrbracket^i}, sk_0)$
- 5: $\llbracket loss \rrbracket \leftarrow \llbracket l(y, y') \rrbracket$
- 6: 标签方将 $\llbracket loss \rrbracket$ 发送给边缘节点

/ Running on edge nodes D_0 */*

- 7: $(\llbracket g \rrbracket, \llbracket W_{TM}^U \rrbracket) \leftarrow Back_S(\llbracket W_{TM} \rrbracket, \llbracket loss \rrbracket)$
- 8: 边缘节点将 $\llbracket g \rrbracket$ 发送给每个数据方

/ Running on D_i */*

- 9: $\llbracket W_{BM_i}^U \rrbracket \leftarrow Back_C(\llbracket W_{BM_i} \rrbracket, \llbracket g \rrbracket)$

训练结束后，顶部模型和底部模型由多方持有，并且可以在多方执行推理过程。若需要将模型归属为确定的某方，则恢复全局模型。首先，由加密的底部模型和顶部模型恢复加密的全局模型。随后，由持有密钥的各方使用各自的私钥得到对应的解密组份并发送给模型归属方。最后得到解密的全局模型。

算法 3-6 恢复全局模型（可选）

输入：加密的模型参数 $\llbracket BM_i \rrbracket$ 和 $\llbracket TM \rrbracket$

输出：全局模型 GM

/ Running on K */*

1: $\llbracket GM \rrbracket \leftarrow RecoveryGM(\llbracket BMs \rrbracket, \llbracket TM \rrbracket)$

2: 将 $\llbracket GM \rrbracket^0$ 发送给边缘节点， $\llbracket GM \rrbracket^i$ 发送给 D_i ，得到 $P_{\llbracket GM \rrbracket^0}$ 和 $P_{\llbracket GM \rrbracket^i}$

3: $GM \leftarrow Dec'(\llbracket GM \rrbracket, P_{\llbracket GM \rrbracket^i}, sk_k)$

3.3 SecureSL 优化方法

引入同态加密带来的巨大的计算开销，成为阻碍神经网络加密训练进一步研究与发展的障碍，因此研究如何提高模型加密训练中的同态操作计算效率是至关重要的。本章方案通过 SIMD 技术实现的并行计算来提高计算效率，并进一步的对神经网络加密训练中的点积计算过程进行优化，以减少昂贵的同态操作如旋转、同态乘法。表 3-2 展示了同态加法 add 、同态明文与密文乘法 $Mult_{pt}$ 、同态乘法 $Mult_{ct}$ 以及旋转 (*Rotation*) 操作在插槽数 slot=4096 时的开销对比，其中 add , $Mult_{pt}$ 和 $Mult_{ct}$ 总开销表示在包含 4096 个明文的两个密文之间执行同态操作的成本，平均开销表示对两个密文的插槽中的单个明文进行同态操作的平均开销。旋转的单次开销表示将包含 4096 个插槽的密文旋转一个插槽的开销。从表 3-2 得出，旋转操作的开销远远高于其他操作。而同态乘法 $Mult_{ct}$ 的开销虽然比旋转操作开销小，但相比于 add 和 $Mult_{pt}$ 仍然较大，相比之下， add 和 $Mult_{pt}$ 的开销可以忽略不计。

表 3-2 同态操作开销对比

Table 3-2 The Cost of Homomorphic Operations

操作	开销（毫秒，slot=4096）
$Mult_{ct}$ (总开销/平均开销)	5133/1.253
$Mult_{pt}$ (总开销/平均开销)	264/0.064
Add (总开销/平均开销)	81/0.020
$Rotation$ (单次开销)	3723

本章方案主要针对 CNN 模型，其卷积层和全连接层涉及大量线性操作，如矩阵点积、向量点积，因此点积运算的效率是影响模型训练效率的关键因素之

一。在基于 SIMD 的 SecureSL 的加密计算中，点积运算主要由同态加法 add 、同态明文与密文乘法 $Mult_{pt}$ 、同态乘法 $Mult_{ct}$ 以及旋转（*Rotation*）操作组成。由表 3-2 可得出，降低 SecureSL 计算开销的最有效方法是减少 $Mult_{ct}$ 和旋转操作的数量。如图 3-2 所示，本节提出两种针对点积运算的优化方法： $\langle PT_M, CT_V \rangle$ 用于明文矩阵和密文向量的点积， $\langle CT_M, CT_M \rangle$ 用于两个密文矩阵的点积。此外，为了将上述两个优化算法融入到神经网络训练中，本节介绍了如何调整前向传播中卷积层和全连接层的计算过程，以兼容基于 SIMD 的加密训练方式。

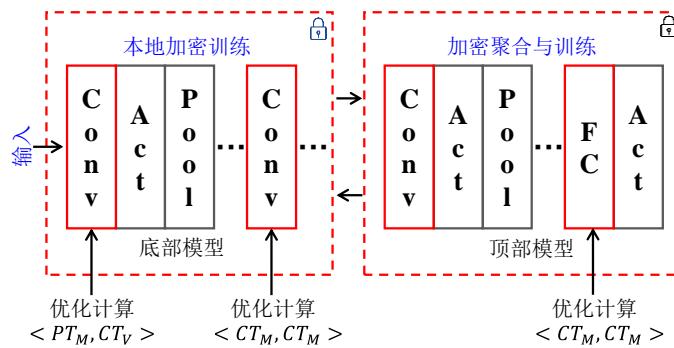


图 3-2 SecureSL 优化
Fig.3-2 The optimization of SecureSL

对于非线性操作，本章方案采用多项式近似将激活函数近似为多项式。因此，在训练过程中仅包含线性运算。反向传播中的矩阵转置可以通过旋转和线性变换完成，其它线性运算方法与前向传播类似，本节使用的符号说明见表 3-3。

表 3-3 符号说明

Table 3-3 Notations

符号	含义
\mathbf{W}	黑体大写字母表示矩阵
\mathbf{x}	黑体小写字母表示向量
$\mathbf{W}^{m \times n}$	维度为 $m \times n$ 的矩阵
\mathbf{x}^n	维度为 n 的向量
\mathbf{w}_j	由矩阵 \mathbf{W} 的第 j 行元素组成的向量。
$W_{i,j}$	矩阵 \mathbf{W} 的第 i 行第 j 列元素
x_i	向量 \mathbf{x} 的第 i 个元素
$\langle \mathbf{W}, \mathbf{x} \rangle$	矩阵 \mathbf{W} 和向量 \mathbf{x} 的点积
$[\mathbf{k}], [\mathbf{K}]$	向量和矩阵的密文
$Rot_{L/R,i}([\mathbf{x}])$	对密文 $[\mathbf{x}]$ 执行向左/右旋转 i 个槽
$Diag(\mathbf{W}, i)$	矩阵 \mathbf{W} 的第 i 个对角线包含的元素
$ShiftR(Diag(), kn_1)$	将 $Diag()$ 中的元素向右移动 kn_1
$Flatten(\mathbf{W})$	将矩阵 \mathbf{W} 按行拉平为一个向量

3.3.1 点积优化方法

(1) 优化一 $\langle PT_M, CT_V \rangle$

本节介绍计算明文矩阵和密文向量点积的优化方法 $\langle PT_M, CT_V \rangle$ 。为了提高效率，同态运算是通过 SIMD 的并行方式进行的，而其中的点积运算需要通过较为昂贵旋转技术来实现。在原始方法中，在将明文矩阵编码为同态加密的明文形式时，需要将矩阵的每一行编码为一个明文，密文向量被编码到一个密文中。随后在 SIMD 的计算形式下，执行明文与密文乘法，随后旋转求和得到点积运算结果。原始方法需要 n 个 $Mult_{pt}$ ， $(n-1)^2$ 个旋转操作。已有研究^[138]提出采用对角线方法减少矩阵向量乘法的旋转数量，当前大多基于同态加密的隐私保护机器学习方案^[97, 139, 140]均沿用了此方法。不同于原始方法，对角线方法将矩阵对角线元素进行提取，并将每组对角线元素编码到同一个明文中，随后执行明文与密文乘法和旋转操作。对角线方法将 $(n-1)^2$ 个旋转操作降低至 $n-1$ 个。为了进一步的降低乘法和旋转操作的数量，本方案采用折半搜索方法对明文矩阵和密文向量点积的计算过程进行了进一步的优化。该方法由大步小步算法（baby-step giant-step）求解离散对数问题的思路所启发，在对对角线元素编码前对对角线元素进行适当的变换，从而减少同态旋转的操作，折半搜索方法将 $n-1$ 个旋转操作降低至 n_1+n_2-2 个，其中 $n=n_1 \cdot n_2$ ， $n_1, n_2 \approx \sqrt{n}$ 。三种方法的示例见图 3-3，图 3-4 以及图 3-5。具体过程见算法 3-7。

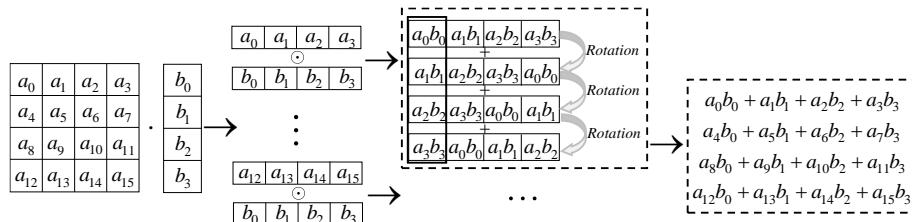


图 3-3 原始方法
Fig.3-3 Naive method

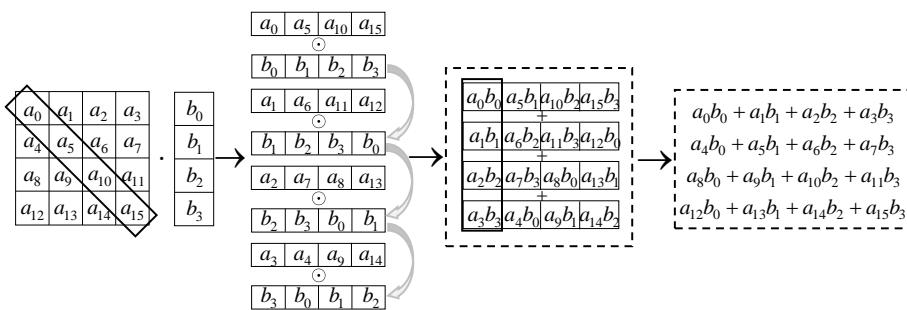


图 3-4 对角线方法
Fig.3-4 Diagonal method

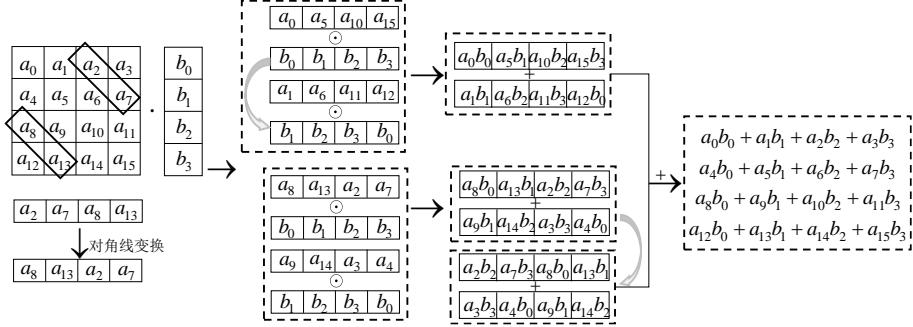


图 3-5 折半搜索方法
Fig.3-5 Meet-in-the-middle method

算法 3-7 $\langle PT_M, CT_V \rangle$

输入: $W^{n \times n}$, $\llbracket x^n \rrbracket$

输出: $\llbracket y \rrbracket = \llbracket \langle W, x \rangle \rrbracket$

原始方法 (n 个 $Mult_{pt}$, $(n-1)^2$ 个 $Rotation$)

1: for $j = 0 \rightarrow n-1$

2: $\llbracket a_j \rrbracket = W_j \cdot \llbracket x \rrbracket$

3: $\llbracket b_j \rrbracket \leftarrow \sum_{i=0}^{n-1} Rot_{L,i}(\llbracket a_j \rrbracket)$

4: end for

5: $\llbracket y \rrbracket \leftarrow (b_{0,1}, b_{1,1}, \dots, b_{n-1,1})$

优化 1 对角线方法 (n 个 $Mult_{pt}$, $n-1$ 个 $Rotation$)

6: for $i = 0 \rightarrow n-1$

7: $\llbracket y_i \rrbracket \leftarrow Diag(W, i) \cdot Rot_{L,i}(\llbracket x \rrbracket)$

8: $\llbracket y \rrbracket += \llbracket y_i \rrbracket$

9: end for

优化 2 折半搜索方法 (n 个 $Mult_{pt}$, $n_1 + n_2 - 2$ 个 $Rotation$)

10: for $k = 0 \rightarrow n_2 - 1$

11: $Diag'() = ShiftR(Diag(), kn_1)$

12: for $j = 0 \rightarrow n_1 - 1$

13: $\llbracket y_i \rrbracket = Diag'(W, kn_1 + j) \cdot Rot_{L,j}(\llbracket x \rrbracket)$

14: $\llbracket y' \rrbracket += \llbracket y_i \rrbracket$

15: end for

16: $\llbracket y \rrbracket += Rot_{L,kn_1}(\llbracket y' \rrbracket)$

17: end for

(2) 优化二 $\langle CT_M, CT_M \rangle$

本节介绍计算两个密文矩阵点积的优化方法 $\langle CT_M, CT_M \rangle$, 其核心思想由论文^[102]中提出的方法衍生。假设有两个 $n \times n$ 维矩阵 A 和 B , 首先采用 CKKS 算法对这两个矩阵进行加密得到 $\llbracket A \rrbracket$ 和 $\llbracket B \rrbracket$, 并计算 $\llbracket A \rrbracket$ 和 $\llbracket B \rrbracket$ 的点积。实现上述目标的一种简单方法是对矩阵中的每个条目分别加密, 然后进行同态乘法和同

态加法运算，这种计算方法需要的加密开销为 $O(n^2)$ ，同态乘法开销为 $O(n^3)$ 。在采用 SIMD 的方式对矩阵进行加密时，首先需要将矩阵 \mathbf{A} 和 \mathbf{B} 拉平为两个维度为 n^2 的向量，然后将每个向量的 n^2 个条目加密到同一个密文中，得到两个密文 $[\![\mathbf{A}]\!]$ 和 $[\![\mathbf{B}]\!]$ 。矩阵 \mathbf{A} 和 \mathbf{B} 的点积可以表示为 $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i=0}^{d-1} \mathbf{A}_i \odot \mathbf{B}_i$ ，其中 \mathbf{A}_i 和 \mathbf{B}_i 是通过对 \mathbf{A} 和 \mathbf{B} 进行特定的转换得到的， \odot 表示对 \mathbf{A}_i 和 \mathbf{B}_i 按分量逐位求积。在密文形式下， $[\![\mathbf{A}_i]\!]$ 和 $[\![\mathbf{B}_i]\!]$ 是对 $[\![\mathbf{A}]\!]$ 和 $[\![\mathbf{B}]\!]$ 进行线性变换得到的。这种计算方法仅需要 2 次加密，旋转和乘法的开销均为 $O(n)$ 。图 3-6 展示了密文矩阵点积计算的优化示例，其中加粗框表示密文，未加粗框表示明文。对两个 $n \times n$ 维矩阵 \mathbf{A} 和 \mathbf{B} 的计算方法可以扩展到 $\langle \mathbf{A}^{m \times n}, \mathbf{B}^{n \times n} \rangle$ 的情况，其中 m 是 n 的倍数。

$$\begin{array}{ccccccccc}
\mathbf{A} & \mathbf{B} & \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{A}_3 & \mathbf{B}_3 \\
\begin{array}{|c|c|c|} \hline a_0 & a_1 & a_2 \\ \hline a_3 & a_4 & a_5 \\ \hline a_6 & a_7 & a_8 \\ \hline \end{array} & \begin{array}{|c|c|c|} \hline b_0 & b_1 & b_2 \\ \hline b_3 & b_4 & b_5 \\ \hline b_6 & b_7 & b_8 \\ \hline \end{array} & = & \begin{array}{|c|c|c|} \hline a_0 & a_1 & a_2 \\ \hline a_4 & a_5 & a_3 \\ \hline a_8 & a_6 & a_7 \\ \hline \end{array} & \odot & \begin{array}{|c|c|c|} \hline b_0 & b_4 & b_8 \\ \hline b_3 & b_7 & b_2 \\ \hline b_6 & b_1 & b_5 \\ \hline \end{array} & + & \begin{array}{|c|c|c|} \hline a_1 & a_2 & a_0 \\ \hline a_5 & a_3 & a_4 \\ \hline a_6 & a_7 & a_8 \\ \hline \end{array} & \odot & \begin{array}{|c|c|c|} \hline b_3 & b_7 & b_2 \\ \hline b_6 & b_1 & b_5 \\ \hline b_0 & b_4 & b_8 \\ \hline \end{array} & + & \begin{array}{|c|c|c|} \hline a_2 & a_0 & a_1 \\ \hline a_3 & a_4 & a_5 \\ \hline a_7 & a_8 & a_6 \\ \hline \end{array} & \odot & \begin{array}{|c|c|c|} \hline b_6 & b_1 & b_5 \\ \hline b_0 & b_4 & b_8 \\ \hline b_3 & b_7 & b_2 \\ \hline \end{array}
\end{array} \\
\\
\boxed{[A_1] \leftarrow [A]} & & \boxed{[A_2] \leftarrow [A_1]} & &
\end{array}$$

图 3-6 密文矩阵点积优化

Fig.3-6 The optimization method to compute the dot product of two ciphertext matrices

3.3.2 卷积层计算

本节通过一个较为简单的举例说明在 SecureSL 中卷积层是如何进行计算的，该举例可以较为简单地扩展到复杂的神经网络训练中。假设在一个二维卷积神经网络中，SecureSL 的底部模型前两层是卷积层，模型输入是一个 $n \times n$ 维矩阵 \mathbf{I} ，该矩阵的维度为填充 0 后的维度。在第一层卷积中，卷积核 \mathbf{K}_1 的维度为 $m_1 \times m_1$ ，步长为 s_1 ，通道数为 1。在第一层卷积中，卷积核 \mathbf{K}_2 的维度为 $m_2 \times m_2$ ，步长为 s_2 ，通道数为 1。我们设每个密文包含的插槽个数可以在初始化时通过调节特定的参数确定，假设插槽个数为 sl ，即每个密文中包含 sl 个明文，这些明文可以并行执行同态评估。当要加密的明文数量（如一个明文向量的元素数量）少于 sl 时，未使用的插槽可以用 0 进行填充或在每个密文中加密多个数据组（如在同一个密文中加密两个向量）。相反，如果需要加密的明文数量多于

sl , 则需要将这些明文分组加密成多个密文。因此, 在初始化阶段需要根据实际需求合理设置插槽数量。

在预处理阶段, 卷积核 \mathbf{K}_1 被按行拉平为向量, 该向量被加密为密文。由于除输入层外, 每一层的输入 (即上一层的输出) 都是密文形式, 因此需要在加密前对卷积核 \mathbf{K}_2 进行变换以适应基于 SIMD 的密文计算的规则。假设第二层的输入是一个大小为 $n_1 \times n_1$ 的矩阵, 为了正确执行密文点积运算, 需要在 \mathbf{K}_2 的每一行都填充 $(n_1 - m_2)$ 个 0, 并将其拉平为 n_1^2 维度的向量 \mathbf{k}_2 。然后, 对 \mathbf{k}_2 加密得到 $\llbracket \mathbf{k}_2 \rrbracket$ 。

在第一层卷积的计算中, 首先根据卷积核大小和步长将每个输入的数据矩阵 \mathbf{I}_i 分成 n_1^2 ($n_1 = (n - m_1) / s + 1$) 个 $m_1 \times m_1$ 维的特征图 $\mathbf{I}_{i,j}$ 特征图。随后, 将每个特征图的矩阵 $\mathbf{I}_{i,j}$ 拉平为一个向量, 并按照 $\mathbf{I}_{1,1}, \mathbf{I}_{2,1}, \dots, \mathbf{I}_{b,1}, \dots, \mathbf{I}_{k,n_1^2}, \dots, \mathbf{I}_{k+b,n_1^2}$ 的顺序对所有向量进行排序整合为一个矩阵。因此, 所有输入数据的矩阵可以通过上述方法转换为一个维度为 $n_1^2 \cdot k \times m_1^2$ 的矩阵 \mathbf{C} 。卷积核 \mathbf{K}_1 在预处理阶段加密为 $\llbracket \mathbf{k}_1 \rrbracket$ 。因此, 卷积计算过程被转换为明文矩阵和密文向量的点积计算过程 $\langle \mathbf{C}, \llbracket \mathbf{k}_1 \rrbracket \rangle$ 。该过程可以通过优化一 $\langle PT_M, CT_V \rangle$ 实现。为了兼容优化一, 需要将矩阵 \mathbf{C} 通过填充零转换成方形矩阵。最后得到输出的密文可以视为一个加密矩阵, 它是通过对大小为 $n_1 \times n_1$ 的 k 个特征图 $\mathbf{X}_1, \dots, \mathbf{X}_k$ 进行拉平后按列排序得到的。

在激活函数的计算中, 通过多项式近似方法将激活函数转换为多项式, 再以密文方式进行计算。已有较多工作^[95, 102]提出高效准确的近似方法, 因此在本节不再详述。在后续的描述中, 线性层的输出默认为相应层的激活函数输出。

在第二层卷积的计算中, 输入是大小为 $n_1^2 \times k$ 的矩阵 \mathbf{X} 。卷积核 \mathbf{K}_2 在预处理阶段被加密为密文 $\llbracket \mathbf{k}_2 \rrbracket$ 。首先, 通过计算 $\llbracket \mathbf{K} \rrbracket = \sum_{i=0}^{n_2-1} \sum_{j=0}^{n_2-1} Rot_{R,r}(\llbracket \mathbf{k}_2 \rrbracket)$ 将 $\llbracket \mathbf{k}_2 \rrbracket$ 转换为 $\llbracket \mathbf{K} \rrbracket$, 其中 $n_2 = (n_1 - m_2) / s_2 + 1$, $r = i \cdot s \cdot n_1 + j \cdot s + (i \cdot n_2 + j)n_1^2$ 。因此, 卷积计算过程被转换为两个加密矩阵的点积计算过程 $\langle \llbracket \mathbf{K} \rrbracket, \llbracket \mathbf{X} \rrbracket \rangle$, 该过程可以通过优化二 $\langle CT_M, CT_M \rangle$ 实现。最后得到输出的密文可以视为一个加密矩阵, 它是通过对大小为 $n_2 \times n_2$ 的 k 个特征图 $\mathbf{Y}_1, \dots, \mathbf{Y}_k$ 进行拉平后按列排序得到的。

3.3.3 全连接层计算

如果底部模型的第一层是全连接层, 假设输入向量为 n 维向量 \mathbf{a} , 权重是 $m_1 \times n$ 维的矩阵 \mathbf{W}_1 。在预处理阶段, 矩阵 \mathbf{W}_1 被拉平为一个向量 \mathbf{w}_1 , 加密向量 \mathbf{w}_1 得到密文 $\llbracket \mathbf{w}_1 \rrbracket$ 。输入向量 \mathbf{a} 被复制 m 次, 这 m 个向量组成一个新的向量 \mathbf{r} 。然后进行如下计算:

$$\llbracket \mathbf{y}_1 \rrbracket \leftarrow \text{Mult}_{pt}(\llbracket \mathbf{w}_1 \rrbracket, \mathbf{r}) \quad (3-1)$$

$$\llbracket \mathbf{y} \rrbracket \leftarrow \sum_{i=0}^{n-1} \text{Rot}_{L,i}(\llbracket \mathbf{y}_1 \rrbracket) \quad (3-2)$$

对于 $i = 1 \rightarrow m \cdot n$

$$v_i = \begin{cases} 1 & i \bmod n = 1 \\ 0 & \text{others} \end{cases} \quad (3-3)$$

$$\mathbf{v} = \{v_1, \dots, v_{m \cdot n}\} \quad (3-4)$$

$$\llbracket \mathbf{y} \rrbracket \leftarrow \text{Mult}_{pt}(\llbracket \mathbf{y} \rrbracket, \mathbf{v}) \quad (3-5)$$

利用上述过程同时处理 n 个输入，得到 n 个密文 $\llbracket \mathbf{y}^{(0)} \rrbracket, \dots, \llbracket \mathbf{y}^{(n-1)} \rrbracket$ 。然后计算 $\llbracket \mathbf{Y} \rrbracket \leftarrow \sum_{i=0}^{n-1} \text{Rot}_{R,i}(\llbracket \mathbf{y}^{(i)} \rrbracket)$ 。该层的输出密文可视为一个由 n 个 m_1 维向量 $\langle \mathbf{W}_1, \mathbf{a}_i \rangle$ ($i = 1 \rightarrow n$) 按列排列得到的加密矩阵。

对于除第一层外的底部模型和顶部模型中的全连接层，输入是密文 $\llbracket \mathbf{Y} \rrbracket$ 。权重矩阵 \mathbf{W} 在预处理阶段被加密为 $\llbracket \mathbf{W} \rrbracket$ 。因此，全连接层的计算过程被转换为两个加密矩阵的点积计算过程 $\langle \llbracket \mathbf{W} \rrbracket, \llbracket \mathbf{Y} \rrbracket \rangle$ ，该过程可以通过优化二 $\langle CT_M, CT_M \rangle$ 实现。

3.4 隐私保护及安全性分析

本章方案的主要保护参与拆分学习的多方本地数据和标签的隐私信息以及模型的隐私信息。潜在威胁包括：① 终端侧数据方与标签方会试图挖掘其他参与方和模型的隐私信息；② 终端侧数据方与标签方可能会与其他参与方或恶意攻击者合谋挖掘训练过程中的隐私信息；③ 边缘节点会试图窃取训练过程中的数据和模型的隐私信息；④ 边缘节点会通过主动攻击破坏数据或模型的完整性和机密性。上述隐私保护问题主要依赖于所基于的多密钥同态加密的安全性，其安全性分析在已有研究^[102, 141]中已给出详细证明。因此，本方案的隐私保护以“MK-CKKS 是选择明文攻击下的不可区分性（Indistinguishability under Chosen-Plaintext Attack, IND-CPA）安全的”为前提，从机密性、完整性和抗合谋等方面对方案的隐私保护效果进行分析。

1) 机密性：机密性主要包括训练数据、标签以及模型的机密性。首先训练数据由终端侧数据方在本地持有并训练，仅将底部模型的输出值发送给边缘节点。由于边缘节点或恶意攻击者可能会采用数据重构攻击从底部模型的输出值中恢复训练数据的隐私信息，本方案将底部模型的输出值采用 MK-CKKS 加密后发送至边缘节点。得益于 MK-CKKS 的安全性，边缘节点无法从加密的输出值中获取任何关于训练数据的有效信息，从而防止数据重构攻击，保护了训练数据的机密性。标签由标签方持有，边缘节点将加密输出发送给标签方，标签

方计算加密损失返还给边缘节点，边缘节点与数据方无法从加密损失中推断出任何标签的有效信息，从而保证了标签的机密性。模型包括由终端侧持有的底部模型和边缘侧持有的顶部模型，模型以加密的形式进行训练，从而保证了模型的机密性。因此，方案满足机密性能够有效防止情况①和情况③造成的潜在隐私泄露威胁。

2) 完整性：由于边缘节点被假设为主动攻击敌手，因此边缘节点可能会在训练过程中给出错误的训练结果，破坏模型的完整性。在本方案中，负责执行顶部模型训练过程的边缘节点组成区块链，通过共识机制确保计算结果的正确性、一致性和可靠性，从而保证了模型的完整性。因此，方案满足完整性和机密性能够有效防止情况④造成的潜在隐私泄露威胁。

3) 抗合谋攻击：本方案中的合谋主要考虑终端侧数据方与其他数据方或边缘侧节点合谋。若采用常规同态加密，数据方采用相同的公钥对底部模型输出值进行加密后发送给边缘节点，密钥持有方（通常为数据方）可以与边缘节点合谋，解密输出值，并重构训练数据。本方案采用多密钥同态加密，各数据方分别采用不同的公私钥对本地底部模型的输出进行加密，从而防止了合谋攻击。此外，最终模型需要由多密钥同态加密中的所有 N 个参与方联合才可解密，可以防止 $N-1$ 合谋攻击。因此，方案可以抗合谋攻击能够有效防止情况②造成的潜在隐私泄露威胁。

3.5 实验评估

本章主要从优化方法效率、隐私保护效果、准确率等方面评估 SecureSL 的性能。首先，在效率方面，对所提针对明文矩阵和密文向量点积运算、密文矩阵间的点积运算的两种优化方法的效率进行了测试，并与常规运算方法进行了对比分析。其次，在隐私保护方面，以没有额外隐私保护措施的拆分学习方案以及基于噪声扰动的数据和标签隐私保护方案为基准方案。通过测试 SecureSL 针对拆分学习的不同攻击方法^[29, 30]的防御能力，以及已有解决方案^[29]的隐私保护效果，评估了 SecureSL 针对数据和标签等隐私信息推理攻击的隐私保护能力。随后，将 SecureSL 与文献^[27]中提出的拆分学习方案进行了比较，以展示 SecureSL 的隐私保护方法对模型准确率的影响。

3.5.1 实验设置

本章实验是在配备了英特尔酷睿 (TM) i7-12700F @ 2.10 GHz CPU、16GB 内存和英伟达 NVIDIA RTX 3060 GPU 的 Windows 操作系统电脑上进行的。

本章方案在 Criteo 和 MNIST 两个数据集上进行了训练和测试。Criteo 是一

个点击率预测基准数据集，包含 26 个分类特征和 13 个连续特征。本章将数据集分为 80,000 个训练样本和 20,000 个测试样本。Criteo 用于分析 SecureSL 的标签保护能力和准确性。MNIST 是一个包含 60,000 个训练样本和 10,000 个测试样本的图像数据集。MNIST 用于证明 SecureSL 抵御数据重构攻击的能力。

3.5.2 效率分析

本节对 SecureSL 的优化算法效率进行了分析。首先对优化一中的明文矩阵与密文向量乘法优化进行了对比。我们选择 256 维度的矩阵与向量进行测试，对比了算法 3-7 中展示的原始方法、对角线方法及本方案采用的折半搜索方法下执行明文矩阵与密文向量乘法所需要的时间开销。如表 3-4 所示，相比于对角线方法与折半搜索方法，原始计算方式远远大于优化后的计算方式；相比于对角线方法，折半搜索方式又进一步的将计算效率进行了提升。主要原因在于，在原始计算方式中，明文矩阵的每一行单独编码，并于密文向量相乘。随后执行旋转操作。计算过程需要 n 次明文与密文乘法，及 $(n-1)^2$ 次旋转操作，在本实验中 $n=256$ 。在对角线方法中，将矩阵的对角线元素提取后编码为密文，随后与密文向量执行乘法和旋转操作，计算过程需要 n 次明文与密文乘法，及 $n-1$ 次旋转操作。在折半搜索方法中，将矩阵的对角线元素进一步的细化，降低旋转操作的执行次数，计算过程需要 n 次明文与密文乘法，及 n_1+n_2-2 次旋转操作。

表 3-4 优化一效率对比

Table 3-4 Efficiency Comparison for Optimization 1

方法	时间 (s)
原始方法	524.22
对角线方法	2.12
折半搜索方法	1.24

在实际的卷积计算中，若采用基本的加密计算方法，则一个 $m_1 \times m_1$ 维卷积核 \mathbf{K}_1 需要被加密为 m_1^2 个密文，矩阵 \mathbf{K}_1 中的每个条目都需要单独的加密运算。当采用基于 SIMD 的优化方法时，卷积核 \mathbf{K}_1 被拉平为向量，所有条目被编码并加密到一个密文中，只需要一次加密操作。在第一层卷积的计算中，通过采用优化一 $<PT_M, CT_V>$ ，基于 SIMD 的卷积操作可以从 m_1^2 个 $Mult_{pt}$ 操作和 m_1^4 个旋转操作减少到最多 m_1^2 个 $Mult_{pt}$ 操作和 $m_{11}^2 + m_{12}^2$ （其中 $m_1^2 = m_{11}^2 \times m_{12}^2$ ）个旋转操作。

其次，对优化二中的密文矩阵与密文矩阵乘法优化进行了对比。我们选择

两个 64 维度的加密矩阵进行测试，首先选择目前采用较为广泛的加密矩阵乘法计算方法作为本方案的对比基准^[139, 142]。基准一设置为将矩阵中的每一行（或每一列）元素加密到同一个密文中，对两个矩阵的行、列执行密文乘法和逐次旋转操作，最后实现密文矩阵的乘法。基准二在对比基准一的基础上优化了旋转操作，将旋转操作由 $O(n^2)$ 降低至 $O(n \log n)$ 。随后，对比了在基准方法和优化二中的方法执行密文矩阵乘法所需要的时间开销。如表 3-5 所示，基准方法需要的加密开销为 $O(n)$ ，同态乘法开销为 $O(n^2)$ ，旋转开销为 $O(n^2)$ 或 $O(n \log n)$ 。优化二中的计算方法仅需要 2 次加密，旋转和乘法的开销均为 $O(n)$ 。在实际的卷积和全连接计算中，可以将计算过程统一为两个密文矩阵之间的乘法操作，通过采用优化二 $\langle CT_M, CT_M \rangle$ 提升计算效率。

表 3-5 优化二效率对比

Table 3-5 Efficiency Comparison for Optimization 2

方法	时间 (s)
基准一	4098.43
基准二	215.39
优化二	131.9

3.5.3 隐私保护性能对比

在拆分学习中，数据方和标签方都存在隐私泄露的风险。对于数据方，在训练过程中，恶意参与方可能会试图进行数据重构攻击，以重构其他数据方的本地隐私数据。对于标签方，恶意参与方可能会在训练过程中尝试进行标签推理攻击，以提取标签中的隐私信息。本节给出了对 SecureSL 抵抗数据重构攻击和标签推理攻击方面的有效性的测试。

(1) 数据泄露评估

本小节对 SecureSL 抵御数据重构攻击的有效性进行了评估。首先，基于 FSHA^[30] 方案实现重构攻击，以挖掘数据方本地数据的敏感信息，并采用 MNIST 数据集评估 SecureSL 对数据重构攻击的抵抗效果。具体实验设置与 FSHA 中的设置相同。FSHA 攻击模型允许攻击者利用数据方发送到边缘节点的底部模型输出重构数据方的本地数据。数据方的数量不会影响攻击的效果，因为每个数据方都会在本地独立计算底部模型，因此在模拟实验中建立了一个单一数据方，分别以明文数据和加密数据作为底部模型的输入进行实验。随后，攻击者对明文数据和加密数据两个实验中的底部模型输出进行重构攻击。如图 3-7 所示，在不采取任何保护措施的情况下，攻击者对以明文数据作为输入的底部模型输出进行重构攻击，该攻击成功地重构了训练数据。相对的，在图 3-8 中，显示

了攻击者采用 FSHA 方案中的重构攻击对 SecureSL 进行攻击时挖掘到的信息。由于底部模型的输出是加密形式的，因此攻击者在执行重构攻击时无法获得任何隐私信息。

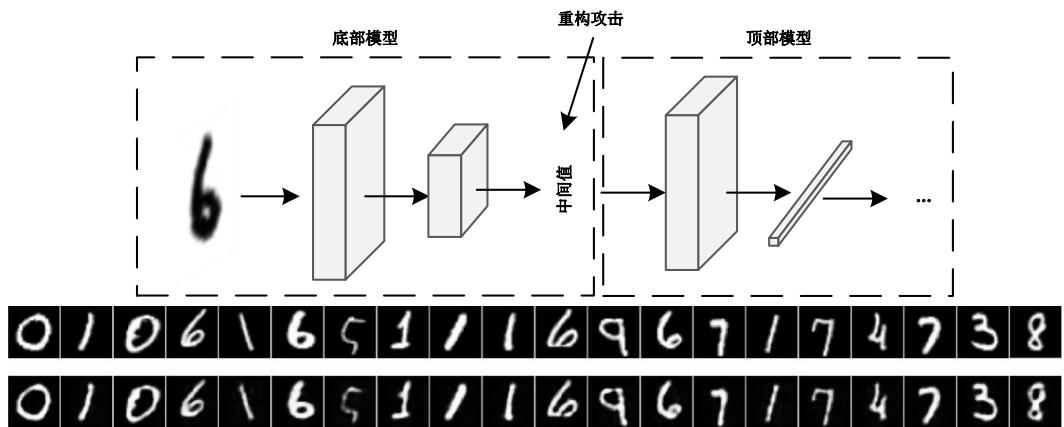


图 3-7 对无隐私保护的拆分学习的重构攻击结果
Fig.3-7 Reconstruction attack result on split learning without protection

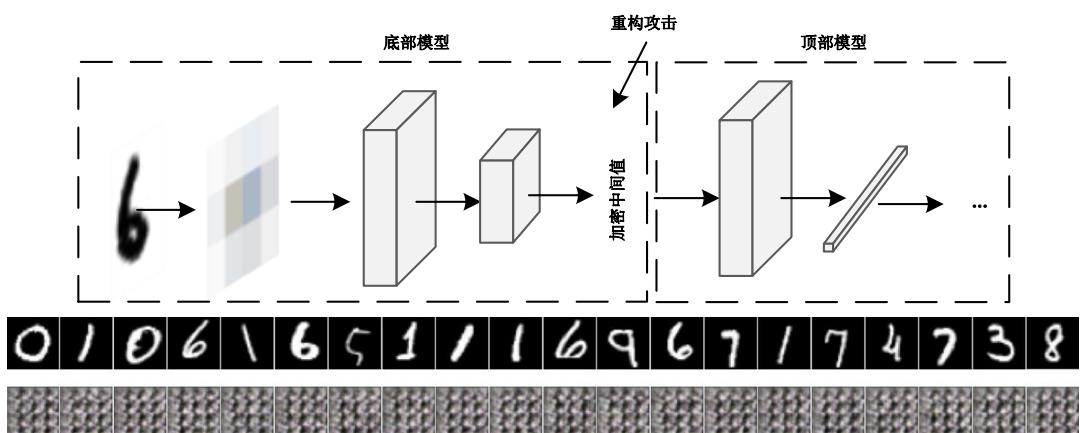


图 3-8 对 SecureSL 的重构攻击结果
Fig.3-8 Reconstruction attack result on SecureSL

同态加密方案提供的安全性确保攻击者无法从底部模型的加密输出中提取任何敏感信息。这一安全特性在使得 SecureSL 在抵御重构攻击时效果显著。此外，采用同态加密的多密钥变体，可以保证方案对合谋攻击的抵御能力，从而确保即使恶意攻击者是合作学习的参与者，也无法通过重构攻击获取敏感信息。

(2) 标签泄露评估

除了保护数据方免受数据重构攻击外，SecureSL 还能确保数据标签的机密性。本小节对 SecureSL 抵御标签推理攻击的有效性进行评估，攻击者的目的是恢复标签方持有的标签中的敏感信息。在本小节中，采用基于规范的攻击（norm-based attack）和基于方向的攻击（direction-based attack）两种攻击方式

[29]，从 norm leak 和 cosine leak 两方面评估了 SecureSL 的标签保护效果，其中 norm leak 是通过对置信度差异以及梯度范数分布表现与标签的相关性的分析设定的，cosine leak 是通过对不同样本梯度方向的余弦相似度设定的。本节采用 Criteo 数据集开展实验，实验结果如图 3-9。

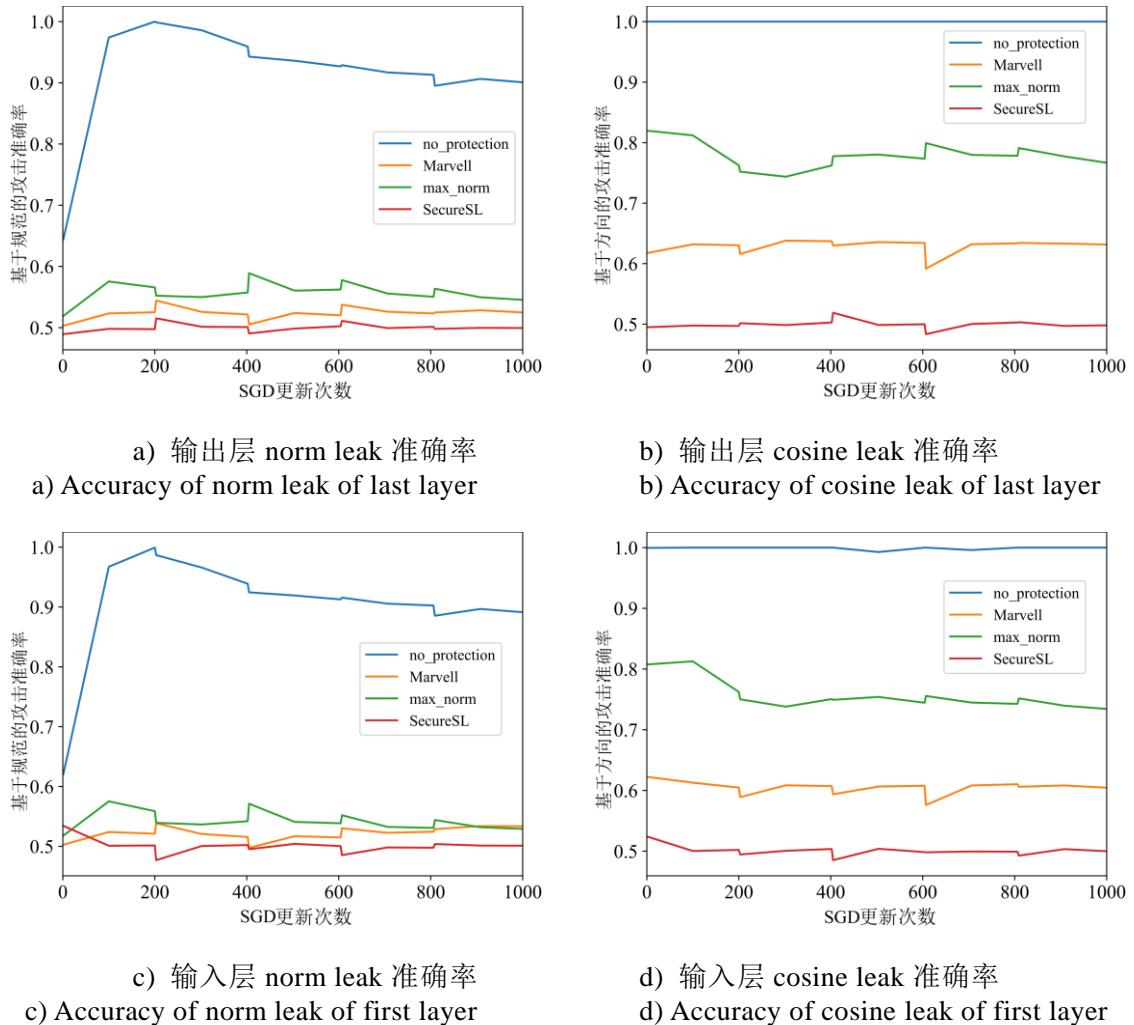


图 3-9 基于 norm-based attack 和 direction-based attack 的标签泄露
Fig.3-9 Label leakage under norm-based attack and direction-based attack

标签的重构是通过攻击模型更新过程中发送给数据方的梯度来实现的。SecureSL 采用同态加密技术保护标签免受此类攻击，具体的，标签方在计算了顶部模型输出与标签间的误差后，将该误差加密发送给边缘节点。顶部模型和底部模型计算参数更新所采用的梯度，均是由该加密误差计算并反向传播得到的，因此采用的梯度均为加密形式。此外，本方案采用多密钥同态加密使得各数据方即使拥有部分密钥，但仅当所有参与方联合持有的密钥时，才能成功解密，因此数据方只能使用加密的梯度对模型进行更新。相比之下，基于噪声扰动的方法会影响模型效用，需要在隐私和效用之间进行均衡。此外，基于噪声

扰动的方法隐私保护效果大大低于同态加密方案提供的隐私保护。

本节将 SecureSL 与现有的基于噪声扰动的两种隐私保护方法 max_norm 和 Marvell 进行比较^[29]。在实验中，选择底部模型第一层和最后一层的激活梯度进行攻击。如图 3-9 所示，相比于其他两个方案，SecureSL 对基于规范的攻击和基于方向的攻击都能达到最佳防护效果。在基于规范的攻击中，SecureSL 和另外两种基于扰动的方法在隐私保护效果方面类似。在基于方向的攻击中，SecureSL 的隐私保护效果远远超过另外两种基于扰动的隐私保护方法。

3.5.4 SecureSL 准确率

本小节对 SecureSL 的准确性进行评估。由于在 SecureSL 采用了 CKKS 作为底层同态加密方案保护训练过程中的数据，标签和模型等敏感信息，该方案会在加密过程中引入噪声。虽然引入的噪声会随着对加密数据执行乘法等操作而累积，最终可能导致训练的模型准确性下降，但 CKKS 中固有的重缩放技术和计算级数限制可以将噪声限制在较小的范围内，从而将其对准确性的影响降至较低。另一方面，现有方案^[27, 29]提出的隐私保护方案，利用噪声扰动来保护切割层输出和梯度。这些方案具有一个共同的特点，即添加的噪声越多，隐私保护的程度就越高，但为保护隐私而添加的过多噪声反过来会大大降低数据的可用性，从而需要在隐私保护和模型效用之间进行权衡。

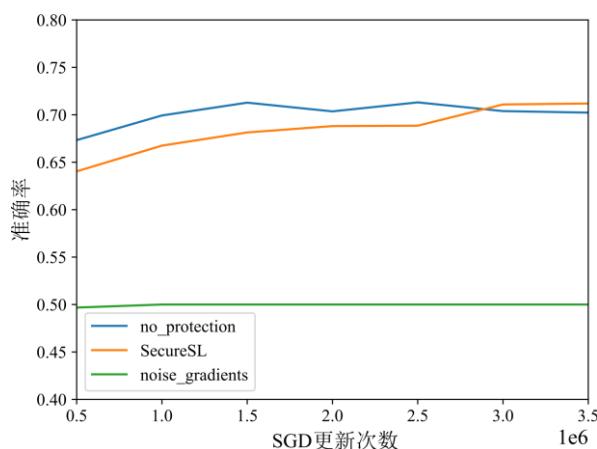


图 3-10 准确率对比
Fig.3-10 The comparison of the accuracy

本节实验采用了与已有方案^[27]相同的方法模拟纵向分布的数据（即具有不同特征的数据集），均采用 Criteo 数据集和相同的模型结构。本节将 SecureSL 的性能与无隐私保护的拆分学习、带噪声扰动的拆分学习方法进行了比较。对带噪声扰动的拆分学习方案设定噪声规模为 1e-2，以提供与 SecureSL 相接近的隐私保护水平。如图 3-10 所示，在拆分学习中添加噪声扰动会显著降低模型的

准确性，而 SecureSL 能够达到与无隐私保护的拆分学习相接近的准确率。

3.6 本章小结

本章提出了一种支持隐私保护的合作学习方案：SecureSL。SecureSL 构造了一个基于拆分学习的边-端合作学习框架。本方案通过将拆分学习和多密钥同态加密相结合，使得持有不同特征数据的各参与方能够合作训练模型，并确保参与者和模型所有者的隐私，抵御合谋攻击。此外，本章提出了两种优化算法，并以兼容 SIMD 的方式优化了模型训练的计算过程。实验表明 SecureSL 可以有效抵御数据重构攻击和标签推理攻击，并且相比于已有的基于噪声扰动的方法，能提供更好的隐私保护性能。在准确率方面，与基于噪声扰动的拆分学习相比，SecureSL 可以在提供相似的隐私保护效果的前提下，达到更高的模型准确率。

不足的是，虽然 CKKS 同态加密方案由于其可以支持浮点数的加密计算，而在神经网络中得到了广泛关注，但由于其固有的计算开销大、有限的级数限制等问题，基于 CKKS 的神经网络训练方案仍然处于研究与探索的初步阶段，距离实际落地应用还有较大的差距。此外，在实际开发与实现过程，由于神经网络与同态加密之间的壁垒，目前仅能在一些简单的网络中进行模拟实现，本章的研究仅能展示部分简单结果，为神经网络合作训练的隐私保护方法提供一个可行的思路。后续研究将继续优化同态加密在神经网络中的应用效率，以期达到实际应用中的需求。

第 4 章 可验证神经网络隐私保护多方推理方案

4.1 问题描述

随着对数据安全与隐私问题关注度的提升，神经网络推理的云服务中存在的下述两个主要问题得到了研究人员的关注：

1) 隐私保护：在神经网络推理云服务中，用户需要将待推理的数据传输给部署了神经网络模型的云服务器。然而，由于云服务器对于用户不是可信的，用户数据中的敏感信息可能会被泄露或滥用。为了解决这个问题，已有部分相关工作^[54, 79, 143]提出基于安全多方计算的神经网络安全推理。这种方法将模型和数据通过秘密共享的方式拆分为若干份额，然后由多个计算方以合作的方式进行计算。由于每个计算方只持有模型和数据的秘密份额，因此保护了模型和数据中的隐私信息。

2) 可验证性：在神经网络推理服务中，存在两个不可忽视的问题影响服务效果。首先，模型所有者可能会夸大其模型的准确率，或在推理服务中提供准确率较低的模型而非最初宣称的模型。其次，云服务器可能会为了降低计算开销等目的，提供错误的推理结果。因此，为了保证模型的真实性和推理结果的正确性，实现神经网络推理的可验证性至关重要。已有部分相关研究^[113, 115, 116]表明，零知识证明在实现可验证神经网络推理方面具有较大的潜力。然而，由于基于 ZKPs 的解决方案存在一些局限性，对于该领域的研究仍处于早期阶段^[110]。

以如下场景为例：在 AI 医生的场景中，患者可能希望能够在不泄露个人隐私信息和诊断结果的情况下寻求服务。同时，患者希望对所宣称的 AI 医生准确性的真实性和诊断结果的正确性有所保证，使用 AI 医生进行医疗诊断过程的可验证性对患者来说是至关重要的。因此，同时确保 AI 医生诊断过程的隐私保护和可验证性，可以为患者提供更加安全的服务。

现有基于 MPC 的隐私保护推理方案，可以保证模型和推理数据的机密性。在这些方案中，由于推理任务是由多方合作执行的，难以确保所有计算方的计算结果均是正确的，因此对计算结果的可验证性变得尤为重要。目前已有部分多方安全推理方案通过实现恶意安全模型防止错误的推理结果，但这些抵抗恶意安全的方案均假设参与方是诚实大多数的，仅能容忍 1 个^[59, 61]或 $t < N / 2$ 个^[91]恶意方，其中 t 为腐败阈值， N 为参与方数量。因此，这些方案只能在恶意方的数量不超过腐败阈值时，才能保证计算的正确性。部分工作^[99, 122]采用敏感样本生成、混合检查等方法来验证推理结果的完整性或正确性，这些方案中的

验证方法类似于抽样检查，其可靠性和有效性是概率性的，并且只能支持批数据推理的验证。此外，上述方案，以及现有其他可验证多方推理方案^[120, 121, 144]，都未能保证推理过程中模型的真实性。

现有基于 ZKP 的可验证神经网络推理方案中，ZKP 的零知识性保证了神经网络模型的机密性，这些方案主要包括：1) 基于交互式 ZKP 的可验证推理方案^[111-114]要求在生成证明时验证者与证明者进行交互；2) 基于非交互式 ZKP 的可验证推理方案，主要采用零知识简洁非交互知识论证（Zero-Knowledge Succinct Non-Interactive Argument of Knowledge, zk-SNARKs）作为底层技术^[115-117, 119, 145]。虽然 zk-SNARKs 在证明生成过程对内存的要求较高，但它不要求验证者与证明者的交互，因此在神经网络推理中仍然具有较大的应用潜力。然而，上述工作并未考虑推理数据的隐私保护问题。

总的来说，上述现有工作均未能同时解决神经网络推理过程中神经网络模型、推理数据和推理结果的隐私保护问题以及神经网络模型真实性和推理结果正确性的可验证问题。

4.1.1 存在的挑战

直观来看，在神经网络推理过程中直接结合 MPC 和 ZKP 就能满足上述可验证性和隐私保护的需求。已有研究结合 MPC 和 ZKP 实现了可审计/可验证 MPC^[146, 147]、用于共享数据的零知识证明^[148-150]等方案。然而，在实际结合过程中，仍然存在一些挑战亟待解决：

首先是如何高效结合 MPC 和 ZKP。由于基于 MPC 的神经网络推理过程是由多个计算方合作执行的，数据和模型均以秘密共享的方式由多个计算方持有，因此 ZKP 中的证明生成过程也需要由多方合作执行。如上所述，交互式 ZKP 需要验证者（即用户）和证明者之间进行多轮交互才能生成对推理过程的证明，这对请求推理服务的用户来说并不友好。因此在本章方案中选择 zk-SNARKs 作为底层 ZKP 方案，从而消除用户参与交互的需求。但直接对 MPC 和 zk-SNARKs 协议进行结合存在一些障碍。例如，MPC 通常支持在有限域上的操作，而 zk-SNARKs 通常基于椭圆曲线群构造，直接结合这两类协议，在生成证明的过程中会产生大量的计算和通信开销。

其次是神经网络推理过程与 MPC 和 zk-SNARKs 的兼容性问题。虽然现有方案提出了一些针对 MPC 和神经网络推理过程兼容的解决方案，例如提出缩放和截断方法以解决多方计算不支持浮点数的问题，但引入 zk-SNARKs 仍然会带来一些额外的问题。例如在基于 MPC 的神经网络推理中，可以使用混合 MPC 协议，即采用算术秘密共享实现线性操作的多方计算，使用混淆电路实现

ReLU 函数的多方计算。然而, zk-SNARKs 需要将所有计算首先转换为 QAP 的形式, 而这只能通过将推理中的各种运算均设置为只包含一个乘积的等式的形式进行转换。因此, 将 ReLU 函数中的不等式直接转换为 QAP 形式是不可行的。

4.1.2 设计目标与方法

鉴于目前已有较多基于 MPC 的安全多方推理方案, 本章研究目标主要是为现有安全多方推理提供可验证的解决方案。本方案首先基于 Groth16^[151]构造了一种多方证明生成方法。该证明方法由于其在证明大小 (仅包含椭圆曲线上的三个点) 和验证时间方面的优势, 对验证者十分友好。随后, 将多方证明生成方法融入到基于 MPC 的神经网络安全推理过程中, 从而实现 VSecNN-一种可验证的神经网络安全多方推理方案。具体来说, 本方案通过在推理和证明生成过程中分别采用基于传统有限域的 MPC 协议和基于椭圆曲线的 MPC, 以更加兼容的方式结合 MPC 与 Groth16, 提高多方证明生成的效率。此外, 通过多项式近似和针对 ReLU 函数 QAP 构造方法以解决神经网络推理计算中的非线性运算与证明生成过程的不兼容问题, 从而实现神经网络推理的隐私保护与可验证。

4.1.3 本章贡献

本章主要贡献总结如下:

- 1) 基于 Groth16 构造了一种多方证明生成方法, 并通过在不同运算阶段采用不同代数结构的 MPC 协议, 提高多方证明生成的效率。
- 2) 基于所构造的多方证明生成方法, 实现可验证的安全多方神经网络推理方案-VSecNN。为了解决神经网络推理与密码方案的兼容性问题, 对神经网络推理过程进行了适应性的修改以构造相应的二次算术程序。
- 3) 在多个公共数据集和神经网络模型上对 VSecNN 进行了实验评估, 并对实验结果进行了详细的分析。

4.2 方案概述

4.2.1 系统模型

VSecNN 中主要由五类实体组成, 包括可信第三方、模型所有者、模型测试者、数据所有者和 N 个服务器。对每个实体的描述如下:

- **可信第三方:** 负责在初始化阶段生成 Groth16 的 CRS 和 KZG 承诺的 SRS。用于生成 CRS 和 SRS 的密钥必须保密, 并在初始化过程结束后销毁。

- **模型所有者:** 首先对其持有的模型参数进行承诺，并生成神经网络模型对应的 QAP 关系。随后向用户证明其所宣称的模型准确率。此外，为了实现安全推理，模型所有者通过秘密共享协议将模型参数分成 N 个份额，并将这 N 个份额分配给 N 个服务器。
- **模型测试者:** 选择测试数据集发送给模型所有者进行推理并获得推理结果与对推理过程的证明。随后，模型测试者验证推理结果，从而实现对模型准确率的验证。
- **数据所有者:** 通过秘密共享协议将待推断的数据分为 N 个份额。这 N 个份额同样也被发送至 N 个服务器上。在收到推理结果和相应的证明后，数据所有者会对推理结果进行验证。
- **服务器:** 由模型所有者、数据所有者、或其他服务器组成，持有模型参数和推理数据的相应秘密份额，并以合作的方式执行推理过程并生成对推理过程的证明。是否要求模型所有者与数据所有者参与计算决定了方案是否能够抵抗 N 个服务器的合谋攻击。

4.2.2 威胁和安全模型

本方案将服务器设定为主动敌手模型，服务器可以主动操纵、干扰或修改数据。与大多数基于安全多方计算的神经网络安全推理方案中假设的被动敌手模型不同，主动敌手模型赋予了攻击者更强大的能力。另一方面，本方案为模型所有者、模型测试者和数据所有者设定被动敌手模型。他们可能试图从神经网络模型或推理的数据中提取隐私信息，但不会主动参与操纵、干扰或修改数据。此外，本方案假设 N 个服务器中，至多有 $N-1$ 个服务器合谋。若模型所有者与数据所有者也作为计算的参与方，由于涉及到这两方的隐私信息，模型所有者与数据所有者不会参与到合谋中，则不对服务器合谋的数量进行限制。后续主要从模型所有者与数据所有者不参与计算的角度进行描述。

本方案主要关注两个阶段的隐私保护：多方推理阶段和多方证明生成阶段。在多方推理阶段，主要包括推理数据、模型和推理结果的隐私保护；而在多方证明生成阶段，则侧重于模型的隐私保护和验证的正确性。

1) 多方推理阶段依赖于秘密共享机制的安全性，如果秘密共享机制满足以下定义的机密性（Confidentiality），抗 $N-1$ 合谋性（ $N-1$ Collusion Resistance），则本方案的多方推理方案被认为是安全的，也就是，在对模型进行多方推理时，只要合谋方小于 N ，就不会泄露推理数据和推理结果的隐私信息。此外，如果本方案的多方证明协议是安全的，则多方推理方案满足可验证性。

定义 4-1（机密性） 在一个算术秘密共享方案中，只有当参与方的数量达

到阈值 N 时，才能重构出原始的秘密。对于任何少于 N 个参与方联合攻击，他们拥有的共享份额不会泄露关于原始秘密的任何信息。这种机密性是信息论安全的，即使攻击者具有无限的计算能力，只要未达到阈值，就无法推断出任何关于秘密的信息。

定义 4-2（抗 N-1 合谋性） 在算术秘密共享协议中，即使多个参与方合谋，他们也无法通过各自的共享份额推断出秘密，除非合谋方的数量达到阈值 N 。

2) 在多方证明生成阶段，设 Γ 为多方证明协议，如果协议 Γ 满足以下定义的完备性 (Completeness)、知识合理性 (Knowledge Soundness) 和 N -零知识性 (N -zero-knowledge)，则协议 Γ 被认为是安全的，也就是，在对模型推理过程生成证明时，不会泄露模型的隐私信息，且可以抵抗 $N-1$ 个恶意服务器合谋。

定义 4-3（完备性） N 个诚实的服务器，每个服务器都持有一个声明和一个见证的秘密份额，他们可以协作生成一个证明。在验证这个证明时，验证者输出 0 的概率 $negl(\lambda)$ 是可以忽略不计的，其中 λ 是安全参数。

定义 4-4（知识合理性） 对于任意 $t \leq N$ 个计算能力有限且不持有见证的敌手，存在一个计算能力有限的提取器 ε ，它可以完全访问 t 个敌手的状态。每当 t 个敌手和 $N-t$ 个诚实的服务器协作生成一个有效的证明时，提取器 ε 就可以计算出一个相应的见证，使得 $(x, w) \notin R$ 和证明 π 说服验证者的概率可以忽略不计。

定义 4-5（ N -零知识性） 对于一个计算能力有限的敌手，它从 N 个服务器生成的 N 个证明份额中获取任何关于见证的秘密份额的信息的概率 $negl(\lambda)$ 是可以忽略不计的。

4.2.3 VSecNN 方案流程

为了同时实现神经网络推理的可验证性和隐私保护，本方案将 Groth16 与多方神经网络安全推理进行了结合，从而实现了 VsecNN。神经网络推理的可验证性主要涉及两个方面：1) 对模型参数的验证，确保推理任务中采用的模型与所承诺的模型一致；2) 保证 N 个服务器执行的推理过程（即前向传播算法）的正确性，确保每个服务器都按照既定的协议执行任务。如图 4-1 所示，VsecNN 的方案流程主要包括：初始化、准确率证明、多方推理、多方证明生成以及验证，图中采用红色突出显示的值表示秘密值，用蓝色突出显示的值表示公开值。每个步骤的说明如下：

1) **初始化**：该步骤包括构建神经网络模型的 QAP 关系 R 以及 Groth16 和 KZG 承诺的可信设置。

2) **准确率证明**：在向用户提供推理服务前，模型所有者需要提供神经网络

模型的公开承诺 cm_W 。任何用户（模型测试者）都可以向模型所有者提交测试数据集，并获得推理结果和所承诺的模型推理过程的证明。使得模型测试者能够验证该证明，以确认所承诺的模型达到了所宣称的准确率。

3) 多方推理：模型所有者将模型参数分为 N 个秘密份额，并将其分发给互不谋合的 N 个服务器。当数据所有者请求推理任务时，也要将其数据分为 N 个秘密份额并发送给服务器。 N 个服务器以合作的方式执行推理过程，并将推理结果发送给用户。服务器无法通过该过程获得任何有关模型或数据的敏感信息。

4) 多方证明生成：在神经网络推理过程中， N 个服务器还需要合作生成对推理过程的证明，该步骤可以确保推理结果的可验证性。

5) 验证：收到推理结果和相应的证明后，用户对其进行验证。如果验证失败，则表明至少有一个服务器没有遵守约定的计算协议，或者在推理过程中使用了不正确的模型参数。

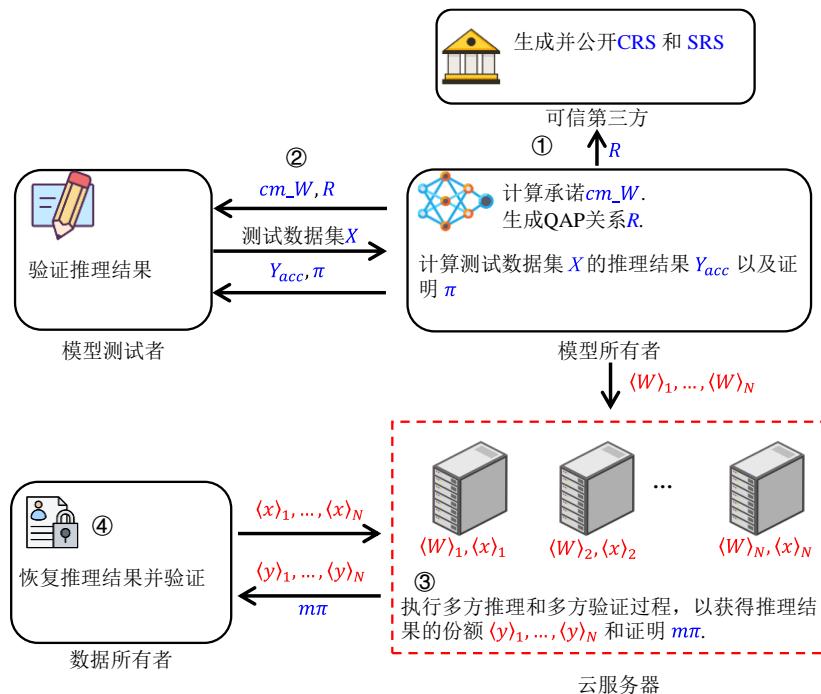


图 4-1 VSecNN 方案流程
Fig.4-1 Overview of VSecNN Workflow

4.3 VSecNN 方案设计

本节对 VSecNN 进行了详细的描述。假设有一个卷积神经网络（Convolutional Neural Network, CNN）模型，其中的参数，包括权重和偏置，表示为 W 。推理过程，即前向传播过程，可以被表示为求解函数 $f(d, W)$ 的过

程，其中“ d ”代表被推理的数据。在本节中，首先介绍将神经网络模型的推理过程转换为 QAP 关系的方法，以及对该关系的可信设置。其次，描述模型所有者对所持模型的准确率的证明方法。随后，详细介绍多方推理和多方证明生成的过程。最后，描述数据所有者对推理结果的验证过程。

4.3.1 神经网络 QAP 构造

在 Groth16 证明系统中，为了使证明过程更加紧凑、高效，需要将神经网络的推理过程转换为 QAP 关系。首先，将函数 $f(d, W)$ 表示为有限域上的算术电路，该电路是由加法门和乘法门构造的。随后，电路中的加法门和乘法门被转换为 R1CS 约束。最后，将整个 R1CS 约束系统转换为 QAP 关系。在上述过程中，我们将电路的部分输入（输入的推理数据）和输出（推理结果）指定为声明。剩余部分包括权重、偏置和中间值指定为见证。我们可以得到一个满足相应声明和见证的 QAP 关系 R' 。模型所有者可以基于关系 R' 生成一个证明，以说服数据所有者，他拥有一个神经网络模型 W ，推理结果 y 是通过对某数据 d 在模型 W 上进行推理得到的。

在构造神经网络的 QAP 关系时，存在两个主要问题。首先，在基于上述构造的关系 R' 生成证明时，能确保服务器执行推理过程的正确性。换句话说，任何满足关系 R' 对应的电路的神经网络都可以被用于该推理过程，且生成的证明均可以验证成功。但基于关系 R' 的证明无法保证用于推理的模型是所宣称的模型。因此需要对模型进行承诺，并将承诺的计算过程也添加到约束系统中，从而转换为 QAP 关系，以确保推理过程始终使用的同一个承诺的模型。在这个过程中，除了输入数据 d 和推理结果 y ，模型参数的承诺值 cm_W 也被添加到声明中。

最终，我们可以得到一个 QAP 关系，表示为：

$$R_{f_cm} = (\mathbb{BG}, l, \{u_i(X), v_i(X), w_i(X)\}_{i=0}^n, t(X)) \quad (4-1)$$

该关系满足下列条件：

- $\mathbb{BG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p, g, h)$ 为证明系统所基于的双线性群。
- $\{u_i(X), v_i(X), w_i(X)\}_{i=0}^n$ 是度为 $m-1$ 的 QAP 多项式，其与神经网络模型的推理过程和模型参数的承诺计算过程相关联。
- $t(X)$ 是度为 m 的目标多项式， $h(X)$ 为对应的商多项式。
- $x = (a_1, \dots, a_l)$ 为声明，包括输入的数据 d ，模型 W 的承诺值 cm_W 以及推理结果 y 。
- $w = (a_{l+1}, \dots, a_n)$ 为见证，包括模型 W 的参数，如权重和偏置，以及在推

理过程中生成的中间结果，即每层网络的输出。

- 对于 $(x, w) \in R_{f_cm}$, $\sum_{i=0}^n a_i u_i(X) \cdot \sum_{i=0}^n a_i v_i(X) = \sum_{i=0}^n a_i w_i(X) + h(X)t(X)$, 其中 $a_0 = 1$ 。

在上述关系中， m 是由前向传播算法和多项式承诺引入的约束数量。约束的数量是影响证明生成效率的主要因素，可以通过优化约束构造方法来提高证明生成的效率。

存在的第二个问题在于，CNN 模型的前向传播算法中的一些非线性运算无法直接表示为加法或乘法门。具体来说，卷积层、全连接层和平均池化层中的所有运算都是线性的，可以直接表示为加法和乘法门。对于激活函数中包含的非线性运算，如指数运算和比较操作，主要包括两种解决方法。第一种是采用多项式近似方法将其近似为多项式表示，这些多项式可以直接转换为 QAP 形式。第二种方法针对 ReLU 激活函数，通过添加一个辅助参数完成 ReLU 的 QAP 关系构造。具体的，在推理过程中，除了输出 ReLU 的结果外，还会产生一个额外的布尔值 tmp 来表示输入数据与 0 的大小关系。如果 x 是 ReLU 函数的输出，并且 $x > 0$ ，则输出 x 和 $tmp = 1$ 。如果 $x \leq 0$ ，则输出 0 和 $tmp = 0$ 。因此，ReLU 中的比较操作可以被表示为一个乘法门 $x' = tmp \cdot x$ ，从而转换为 QAP 形式。

4.3.2 可信设置

在构造了神经网络推理的 QAP 关系后，需要对 Groth16 和 KZG 承诺进行可信设置生成对关系 R_{f_cm} 的公共参，具体描述如下：

- 1) $ZK_Setup(1^\lambda, R) \rightarrow \{pk, sk\}$: 可信第三方随机生成秘密元素 $\alpha, \beta, \gamma, \delta, \varepsilon \leftarrow \mathbb{Z}_p^*$ ，并计算

$$pk_1 = \begin{pmatrix} \alpha, \beta, \delta, \{\varepsilon^i\}_{i=0}^{m-1} \\ \{\gamma^{-1}(\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon))\}_{i=0}^l \\ \{\delta^{-1}(\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon))\}_{i=l+1}^n \\ \{\delta^{-1}(\varepsilon^i t(\varepsilon))\}_{i=0}^{m-2} \end{pmatrix} \quad (4-2)$$

$$pk_2 = (\beta, \gamma, \delta, \{\varepsilon^i\}_{i=0}^{m-1}) \quad (4-3)$$

$$sk = \alpha, \beta, \gamma, \delta, \varepsilon \quad (4-4)$$

CRS 可以表示为 $pk = ([pk_1]_1, [pk_2]_2)$, $[pk_1]_1$ 表示对 pk_1 中的每个元素 x 计算 $[x]_1 = x \cdot g$ ，同样的， $[pk_2]_2$ 表示对 pk_2 中的每个元素 y 计算 $[y]_2 = y \cdot h$ 。生成 CRS

的主要目的是为证明系统提供隐藏证明者隐私信息的方式，同时允许验证者在不获得相关隐私信息的前提下，验证证明的正确性。

2) $KZG_Setup(1^\lambda) \rightarrow PK$: 可信第三方随机生成秘密元素 $\theta \leftarrow \mathbb{Z}_p^*$ ，并计算 $PK = (\lceil \theta^k \rceil, [\theta]_2)$ ，其中 k 是后续可以被承诺的多项式的度

4.3.3 准确率证明

在向用户提供神经网络模型推理服务之前，首先要求模型所有者提供公开的模型准确率证明。具体流程如下：

1) **将模型参数 W 转化为多项式**: 模型所有者声称其持有的模型 W 的准确率为 e 。他首先将模型参数 W 转换为多个向量，每个向量包含 $k+1$ 个元素，其中， k 是在可信设置阶段预设的多项式的度。然后通过拉格朗日插值法将每个向量 $v = (v_1, \dots, v_{k+1})$ 表示为一个度为 k 的多项式 g ，满足 $g(i) = v_i$ 。 g_W 表示由模型参数 W 转换的所有多项式。

2) **生成承诺**: 模型所有者执行 $Commit(PK, g_W)$ 以获得模型参数 W 对应的承诺 cm_W 。具体的，对每个多项式 $g = \sum_{i=0}^k c_i X^i$ ，承诺的计算方式为 $cm = \sum_{i=0}^k c_i \lceil \theta^i \rceil$ 。设 cm_W 为模型 W 转换的所有多项式的承诺，模型所有者向所有用户公开承诺 cm_W 。

3) **准确率测试**: 任何用户（模型测试者）都可以向模型所有者发送不包含敏感信息的测试数据集 D ，以验证承诺的模型 W 是否实现了模型所有者宣称的准确率 e 。模型所有者执行推理计算 $Y' = f(D, W)$ ，并执行证明生成算法 $Prove(R_{f_cm}, pk, D, W)$ 以获得对推理计算的证明 π 。推理结果 Y' 和证明 π 均被发送给模型测试者进行验证。

4) **验证**: 模型测试者执行 $Verify(R_{f_cm}, pk, D, Y', cm_W, \pi) \rightarrow \{0,1\}$ 以验证 Y' 是否是承诺的模型 W 对数据集 D 的正确推理结果。随后，模型测试者将数据集 D 的标签 Y 与推理结果 Y' 对比，以确定承诺的模型是否在测试数据集 D 上实现了所宣称的准确率 e 。

4.3.4 多方推理

在神经网络推理服务中，为了保证模型和数据的隐私，通常采用基于 MPC 的多方安全推理方法，其中算术秘密共享是安全推理中常用的协议，我们将该协议表示为 Π ，则基于协议 Π 的安全推理方案可表示为：

$$\Pi \left(f \left(\left\{ \langle d \rangle_j, \langle W \rangle_j \right\}_{j=1}^N \right) \right) \rightarrow \left\{ \langle y \rangle_j \right\}_{j=1}^N \quad (4-5)$$

首先，模型参数 W 和待推理数据 d 由 N 个服务器 $P_j (j=1, \dots, N)$ 以秘密共

享的方式持有，每个服务器 P_j 持有模型参数份额 $\langle W \rangle_j$ 和输入数据的份额 $\langle d \rangle_j$ 。前向传播算法 f ，包括卷积层和全连接层的线性运算以及激活函数等非线性运算，由 N 个服务器基于协议 Π 合作完成，最后各服务器将各自计算的推理结果份额 $(\langle y \rangle_1, \dots, \langle y \rangle_N)$ 发送给数据所有者。

如 2.3.1 节所述，基于算术秘密共享的安全多方计算协议只能高效支持线性操作，因为无法高效的实现前向传播算法。现有研究采用混合 MPC 协议解决这个问题^[56, 91]，具体的，在线性操作阶段采用算术秘密共享协议如 SPDZ (Smart-Pastro-Damgård-Zakarias)^[152]，在非线性操作阶段则转换为布尔共享或混淆电路。由于非线性操作无法直接被转换为 QAP 形式，因此需要对神经网络的计算电路进行必要的修改。如果对采用多项式近似方法，将非线性激活函数转换为多项式表达，由于此时的计算为近似计算，为了保证验证成功，需要在多方推理阶段对激活函数也采用相同的多项式近似方法。如果激活函数为 ReLU 函数，则在过程期间只需要输出额外的布尔值 tmp ，不需要对推理过程进行额外的修改。

对于多项式近似方法，已有研究^[95]采用平方函数 $sqr(x) = x^2$ 代替非线性激活函数。虽然采用平方函数的方法可以使得浅层模型达到与使用 ReLU 函数的模型相当的准确率。然而，随着模型层数的增加，模型的准确率会大大降低。在本章方案中，我们通过下列切比雪夫多项式近似方法来近似 ReLU 函数：

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad (4-6)$$

并对多项式系数进行微调。表 4-1 展示了在 ShallowNet（包含两层全连接的神经网络）和 LeNet 中测试不同数据集，不同激活函数的准确率对比。结果表明，与平方函数相比，切比雪夫多项式近似方法能保证更高的准确率。

表 4-1 采用不同激活函数的准确率对比
Table 4-1 Accuracy of different activation functions

模型	ReLU	平方函数	切比雪夫多项式
ShallowNet-MNIST	97.24%	96.65%	96.97%
LeNet-Cifar10	59.45%	9.5%	58.27%

4.3.5 多方证明生成

N 个服务器在合作计算前向传播算法 $y = f(d, W)$ 之外，还需要生成零知识证明。因此，为了满足这个需求，需要构造一个多方证明生成方法。该证明可用于验证安全多方计算任务是否被 N 方正确执行。首先，声明和见证 $\{a_i\}_{i=1}^n$ 由 N 个服务器以秘密份额 $\{\langle a_i \rangle_1, \dots, \langle a_i \rangle_N\}_{i=1}^n$ 的形式进行管理。具体的，声明的份额 $\{\langle a_i \rangle_1, \dots, \langle a_i \rangle_N\}_{i=1}^l$ 包括推理数据 d 的份额、承诺值 cm_W 的份额以及在多方推理

阶段生成的推理结果的份额。见证的份额 $\{\langle a_i \rangle_1, \dots, \langle a_i \rangle_N\}_{i=l+1}^n$ 包括模型参数 W 的份额以及 N 个服务器在多方推理阶段计算的中间值的份额。因此，对于多方推理计算 $\Pi(f)$ 的多方证明需要满足 $R_{f_cm}\left(\{\langle a_i \rangle_1, \dots, \langle a_i \rangle_N\}_{i=1}^n\right) = 1$ ，证明生成过程 $M_Prove\left(R_{f_cm}, pk, \{\langle a_i \rangle_1, \dots, \langle a_i \rangle_N\}_{i=1}^n\right) \rightarrow m\pi$ 的具体描述如下：

- 每个服务器 P_j 持有声明的份额 $\{\langle a_i \rangle_j\}_{i=1}^l$ 、见证的份额 $\{\langle a_i \rangle_j\}_{i=l+1}^n$ 、关系 R_{f_cm} 以及 pk 。
- 每个服务器 P_j 随机生成两个元素 $\langle r \rangle_j$ 、 $\langle s \rangle_j$ ，使得 $r = \sum_{j=1}^N \langle r \rangle_j$ ，
 $s = \sum_{j=1}^N \langle s \rangle_j$ 。
- 服务器 P_1 计算：

$$\langle A \rangle_1 = \alpha + u_0(\varepsilon) + \sum_{i=1}^n \langle a_i \rangle_1 u_i(\varepsilon) + \langle r \rangle_1 \delta \quad (4-7)$$

$$\langle B \rangle_1 = \beta + v_0(\varepsilon) + \sum_{i=1}^n \langle a_i \rangle_1 v_i(\varepsilon) + \langle s \rangle_1 \delta \quad (4-8)$$

$$\begin{aligned} \langle C \rangle_1 = & \frac{\sum_{i=l+1}^n \langle a_i \rangle_1 (\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon)) + h(\varepsilon) t(\varepsilon)}{\delta} \\ & + \langle A \rangle_1 \langle s \rangle_1 + \langle B \rangle_1 \langle r \rangle_1 - \langle r \rangle_1 \langle s \rangle_1 \delta \end{aligned} \quad (4-9)$$

- 其他服务器 P_j 计算：

$$\langle A \rangle_j = \sum_{i=1}^n \langle a_i \rangle_j u_i(\varepsilon) + \langle r \rangle_j \delta \quad (4-10)$$

$$\langle B \rangle_j = \sum_{i=1}^n \langle a_i \rangle_j v_i(\varepsilon) + \langle s \rangle_j \delta \quad (4-11)$$

$$\begin{aligned} \langle C \rangle_j = & \frac{\sum_{i=l+1}^n \langle a_i \rangle_j (\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon)) + h(\varepsilon) t(\varepsilon)}{\delta} \\ & + \langle A \rangle_j \langle s \rangle_j + \langle B \rangle_j \langle r \rangle_j - \langle r \rangle_j \langle s \rangle_j \delta \end{aligned} \quad (4-12)$$

- 各服务器将 $[\langle A \rangle_j]_1, [\langle B \rangle_j]_2, [\langle C \rangle_j]_1$ 发送给 P_1 ， P_1 计算：

$$[A]_1 = \sum_{j=1}^N [\langle A \rangle_j]_1 \quad (4-13)$$

$$[B]_2 = \sum_{j=1}^N [\langle B \rangle_j]_2 \quad (4-14)$$

$$[C]_1 = \sum_{j=1}^N [\langle C \rangle_j]_1 \quad (4-15)$$

随后， P_1 将证明 $m\pi = ([A]_1, [B]_2, [C]_1)$ 发送给数据所有者。该证明仅包含椭圆曲线上的三个点。

构建上述多方证明生成过程最原始的方法即直接使用多方推理采用的算术秘密共享协议，多方在该 MPC 协议下合作执行 Groth16 的证明生成算法。通常来说，用于多方安全推理的 MPC 协议通常用于有限域 F_p 上的算术电路计算，

而 Groth16 是基于椭圆曲线群构造的。计算 $[A]_1, [B]_2, [C]_1$ 需要在 MPC 协议下执行椭圆曲线的标量乘法。然而，在基于有限域的 MPC 协议下进行椭圆曲线加法运算时，需要将椭圆曲线群上的运算映射到有限域运算上，大约需要进行六次有限域加法运算和三次有限域乘法运算。因此，即使是简单的椭圆曲线加法，也需要在 MPC 下执行三次有限域乘法运算，该运算需要各方交互完成。而椭圆曲线中的标量乘法则可能会导致数千次有限域中的运算。计算 $[A]_1, [B]_2, [C]_1$ 所需要的三个标量乘法要求 $|A+B+C|$ 椭圆曲线加法，其中， A, B, C 是有限域中的大整数，它们的大小随着关系 R_{f_cm} 的复杂度增加而线性增加。因此，在基于有限域的 MPC 协议下，多方合作生成一个复杂的神经网络推理过程的证明，可能需要数百万次有限域的运算。而 MPC 协议下有限域的运算相比于底层运算本身就慢了上千倍，最后会导致巨大的计算开销。

本方案在证明生成过程中，将基于椭圆曲线群的 MPC 协议替代基于有限域上 MPC 协议。在推理和证明过程中，采用基于有限域的 MPC 协议实现安全多方推理，采用基于椭圆曲线群的 MPC 协议实现多方证明生成。最后将有限域的 MPC 运算扩展为 $Add_F, Mul_F, Add_G, Mul_GP, Mul_GS$ ，具体如下：

$$Add_F\left(\left\{\langle a \rangle_i^F, \langle b \rangle_i^F\right\}_{i=0}^N\right) \rightarrow \langle a+b \rangle_i^F \quad (4-16)$$

$$Mul_F\left(\left\{\langle a \rangle_i^F, \langle b \rangle_i^F\right\}_{i=0}^N\right) \rightarrow \langle a \cdot b \rangle_i^F \quad (4-17)$$

$$Add_G\left(\left\{\langle Q \rangle_i^G, \langle R \rangle_i^G\right\}_{i=0}^N\right) \rightarrow \langle Q+R \rangle_i^G \quad (4-18)$$

$$Mul_GP\left(\left\{\langle a \rangle_i^F, R\right\}_{i=0}^N\right) \rightarrow \langle a \cdot R \rangle_i^G \quad (4-19)$$

$$Mul_GS\left(\left\{\langle a \rangle_i^F, \langle R \rangle_i^G\right\}_{i=0}^N\right) \rightarrow \langle a \cdot R \rangle_i^G \quad (4-20)$$

其中， $\langle \cdot \rangle_i^F$ 和 $\langle \cdot \rangle_i^G$ 分别表示由 P_i 持有的有限域 F_p 中元素的秘密份额和椭圆曲线群 G 上的元素的秘密份额。 Mul_GP 和 Mul_GS 的不同点在于，群元素 R 是公开值还是共享值。上述操作都是由多方合作进行的，其中 Mul_F 和 Mul_GS 需要交互并使用离线阶段生成的 Beaver 乘法三元组。

4.3.6 验证

数据所有者从各服务器收到推理结果的秘密份额 $\langle y \rangle_j$ 和证明 $m\pi$ ，计算 $y = \sum_{j=1}^N \langle y \rangle_j$ 以恢复推理结果 y 。数据所有者持有关系 R_{f_cm} 、 pk 、证明 $m\pi$ ，由输入数据 d ，模型的承诺 cm_W 以及推理结果 y 组成的声明 $\{a_i\}_{i=1}^l$ ，执行

$\text{Verify}\left(R_{f_cm}, pk, \{a_i\}_{i=1}^l, m\pi\right) \rightarrow \{0,1\}$ 以验证该证明。

设 $TMP = \sum_{i=0}^l a_i [\gamma^{-1}(\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon))]_1 \cdot [\gamma]_2$ ，数据所有者检查 $[A]_1 \cdot [B]_2$ 是否等于 $[\alpha]_1 \cdot [\beta]_2 + TMP + [C]_1 \cdot [\delta]_2$ 。如果验证成功，则返回 1。若验证不成功，意味着至少有一个服务器没有遵守约定的计算协议，或者在推理过程中使用了不正确的模型参数。

4.4 隐私保护和安全性分析

本方案的隐私保护和安全性主要从多方推理阶段和多方证明生成阶段进行分析。在多方推理阶段，主要包括推理数据、模型和推理结果的隐私保护，主要依赖于现有的秘密共享机制的安全性，其安全性已在相关研究中得到了证明 [133, 152]，因此不再详细描述。在多方证明生成阶段，主要包括模型的隐私保护和验证的正确性，其依赖于多方证明协议的安全性。

因此，本节主要从多方证明协议 $\Gamma: (\text{ZK-Setup}, M_Prove, Verify)$ 在完备性（Completeness）、知识合理性（Knowledge Soundness）和 N -零知识性（ N -zero-knowledge）三个方面进行安全性分析。

定理 1：如果由 N 个诚实的服务器为 $(x, w) \in R_{f_cm}$ 生成了一个证明 $m\pi$ ，该证明能够以概率 Pr 说服一个诚实的验证者，则协议 Γ 被认为是完备的，且有：

$$\Pr \left[\begin{array}{l} pk \leftarrow \text{ZK-Setup}(1^\lambda, R_{f_cm}) \\ m\pi \leftarrow M_Prove\left(R_{f_cm}, pk, \{\langle a \rangle_i\}_{i=1}^n\right) : \\ 1 \leftarrow \text{Verify}\left(R_{f_cm}, pk, \{a_i\}_{i=1}^l, m\pi\right) \end{array} \right] \geq 1 - negl(\lambda) \quad (4-21)$$

证明：对于 $(x, w) \in R_{f_cm}$ 的证明 $m\pi = ([A]_1, [B]_2, [C]_1)$ ，只有满足等式 $[A]_1 \cdot [B]_2 = [\alpha]_1 \cdot [\beta]_2 + TMP + [C]_1 \cdot [\delta]_2$ 该证明才会被验证者接受。设 $u(\varepsilon) = \sum_{i=0}^n a_i u_i(\varepsilon)$, $v(\varepsilon) = \sum_{i=0}^n a_i v_i(\varepsilon)$, $w(\varepsilon) = \sum_{i=0}^n a_i w_i(\varepsilon)$ ，上述等式左侧和右侧的计算分别如下：

左侧：

$$\begin{aligned} [A]_1 \cdot [B]_2 &= e(Ag, Bh) = ABe(g, h) \\ AB &= \left(\langle A \rangle_1 + \sum_{j=2}^N \langle A \rangle_j \right) \cdot \left(\langle B \rangle_1 + \sum_{j=2}^N \langle B \rangle_j \right) \\ &= \left(\alpha + \sum_{i=0}^n a_i u_i(\varepsilon) + r\delta \right) \cdot \left(\beta + \sum_{i=0}^n a_i v_i(\varepsilon) + s\delta \right) \\ &= u(\varepsilon)v(\varepsilon) + (\alpha\beta + \alpha v(\varepsilon) + \beta u(\varepsilon) + s\delta u(\varepsilon) + r\delta v(\varepsilon) + \beta r\delta + \alpha s\delta + rs\delta^2) \\ &= u(\varepsilon)v(\varepsilon) + Z \end{aligned}$$

右侧：

$$\begin{aligned}
& [\alpha]_1 \cdot [\beta]_2 + TMP + [C]_1 \cdot [\delta]_2 \\
&= e(\alpha g, \beta h) + \sum_{i=0}^l a_i e(\gamma^{-1}(\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon))g, \gamma h) + e(Cg, \delta h) \\
&= \left(\alpha\beta + \sum_{i=0}^l a_i (\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon)) + C\delta \right) e(g, h) \\
&= \alpha\beta + \sum_{i=0}^l a_i (\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon)) + C\delta \\
&= \alpha\beta + \sum_{i=l+1}^n a_i (\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon)) + h(\varepsilon)t(\varepsilon) + As\delta + Br\delta - rs\delta^2 \\
&= w(\varepsilon) + h(\varepsilon)t(\varepsilon) + (\alpha\beta + \alpha v(\varepsilon) + \beta u(\varepsilon) + s\delta u(\varepsilon) + r\delta v(\varepsilon) + \beta r\delta + \alpha s\delta + rs\delta^2) \\
&= w(\varepsilon) + h(\varepsilon)t(\varepsilon) + Z
\end{aligned}$$

因此，如果验证者在验证阶段对一个正确的证明 $m\pi$ 输出 0，则意味着 $u(\varepsilon)v(\varepsilon) \neq w(\varepsilon) + h(\varepsilon)t(\varepsilon)$ ，这与 4.3.1 节中构造的关系 R_{f_cm} 条件不符。因此，协议 $\Gamma: (ZK_Setup, M_Prove, Verify)$ 是完备的。

定理 2：如果对于任意 $t \leq N$ 个计算受限的敌手 \mathcal{A}_i ，和一个计算受限且可以完全访问 t 个敌手状态的提取器 E ，且满足：

$$Pr \left[\begin{array}{l} pk \leftarrow ZK_Setup(1^\lambda, R_{f_cm}) \\ (x, m\pi', w) \leftarrow (A|E)(R_{f_cm}, pk): \\ (x, w) \notin R_{f_cm} \wedge \\ 1 \leftarrow Verify(R_{f_cm}, pk, \{a_i\}_{i=1}^l, m\pi') \end{array} \right] \leq negl(\lambda) \quad (4-22)$$

则协议 Γ 被认为是知识合理的。

证明：假设有一个模拟器可以利用在可信设置中生成的陷门 $sk = (\alpha, \beta, \gamma, \delta, \varepsilon)$ 计算一个模拟的证明 $m\pi$ ，具体的，该模拟器可以随机选择 $A, B \in \mathbb{Z}_p$ ，并计算：

$$C = \frac{A \cdot B - \alpha\beta - \sum_{i=0}^l a_i (\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon))}{\delta}$$

模拟的证明可以被表示为 $m\pi = ([A]_1, [B]_2, [C]_1)$ ，这个证明可以在不包含见证的前提下说服验证者，使得包含见证的证明与不包含见证的模拟证明对于验证者不可区分。

假设有 t 个恶意敌手和 $N-t$ 个诚实服务器共同生成一个证明 $m\pi'$ ，并且恶意敌手不知道见证的具体信息。如果提取器 E 能从恶意敌手生成的证明中提取出见证 w' ，则说明证明 $m\pi'$ 和由模拟器生成的模拟证明对于提取器是可区分的。这意味着，对于可区分的 $m\pi$ 与 $m\pi'$ ，如果验证者接受了模拟证明 $m\pi$ ，那

么他一定会拒绝证明 $m\pi'$ 。因此，提取器 E 无法从一个由不知道见证的恶意敌手生成的可接受的证明 $m\pi'$ 中计算出一个见证 w' 使得 $(x, w') \in R_{f_cm}$ 。因此概率 Pr 可以忽略不计，协议 Γ 是知识合理的。

定理 3：如果一个计算受限的敌手 \mathcal{A} 无法从 N 个诚实的服务器生成的证明的 N 个秘密份额中得到任何关于见证的秘密份额的信息，则协议 Γ 被认为是 N -零知识的。正式的，对于所有 $(x, w) \in R_{f_cm}$ 和一个计算受限的敌手 \mathcal{A} ，满足：

$$\begin{aligned} & Pr \left[\begin{array}{l} pk \leftarrow ZK_Setup(1^\lambda, R_{f_cm}) \\ \langle m\pi \rangle_j \leftarrow M_Prove(R_{f_cm}, pk, \{\langle a_i \rangle_j\}_{i=1}^n) : \\ 1 \leftarrow \mathcal{A}(R_{f_cm}, pk, sk, m\pi) \end{array} \right] \\ & = Pr \left[\begin{array}{l} pk \leftarrow ZK_Setup(1^\lambda, R_{f_cm}) \\ \langle m\pi \rangle_j \leftarrow Sim(R_{f_cm}, sk, \{\langle a_i \rangle_j\}_{i=1}^l) : \\ 1 \leftarrow \mathcal{A}(R_{f_cm}, pk, sk, m\pi) \end{array} \right] \end{aligned} \quad (4-23)$$

证明：假设有 N 个模拟器，每个模拟器可以利用陷门 $sk = (\alpha, \beta, \gamma, \delta, \varepsilon)$ 执行 $\langle m\pi \rangle_j \leftarrow Sim(R_{f_cm}, sk, \{\langle a_i \rangle_j\}_{i=1}^l)$ ，以获得 N 个模拟的证明份额 $\{\langle m\pi \rangle_j\}_{j=1}^N$ ，计算过程如下：

每个模拟器 \mathcal{B}_j 随机选择 $\langle A \rangle_j, \langle B \rangle_j \in \mathbb{Z}_p$ ，使得 $A = \sum_{j=1}^N \langle A \rangle_j$ ， $B = \sum_{j=1}^N \langle B \rangle_j$ ，随后计算：

$$\langle C \rangle_j = \frac{\langle A \rangle_j \cdot \langle B \rangle_j - \alpha\beta - \sum_{i=0}^l \langle a_i \rangle_j (\beta u_i(\varepsilon) + \alpha v_i(\varepsilon) + w_i(\varepsilon))}{\delta}$$

因此，模拟器 \mathcal{B}_j 可以生成模拟证明的份额 $\langle m\pi \rangle_j = ([\langle A \rangle_j]_1, [\langle B \rangle_j]_2, [\langle C \rangle_j]_1)$ ，继而可以在不适用见证的前提下生成模拟证明 $m\pi = \sum_{j=1}^N \langle m\pi \rangle_j = ([A]_1, [B]_2, [C]_1)$ 。这个证明同样可以被诚实的验证者所接受。

记底层 Groth16 协议为 Γ_G ，该协议已被证明具有零知识性。假设有一个可以完全访问敌手 \mathcal{A} 的状态的模拟器 \mathcal{B} ，被用于区分 Groth16 中由见证构造的证明和模拟证明。如果敌手 \mathcal{A} 可以从证明的秘密份额 $\langle m\pi \rangle_j$ 中获得见证的秘密份额 $\{\langle a_i \rangle_j\}_{i=l+1}^n$ ，从而计算出见证 $\{a_i\}_{i=l+1}^n$ 。那么模拟器 \mathcal{B} 则可以利用敌手 \mathcal{A} 的信息区分 Groth16 中的证明与模拟证明。

因此，如果 Groth16 协议满足零知识性，那么协议 Γ 中的模拟证明份额和证明份额一定是不可区分的，计算受限的敌手 \mathcal{A} 无法从 N 个证明份额中获得任何关于见证份额的信息。因此协议 Γ 满足 N -零知识性。

4.5 实验评估

4.5.1 实验设置

在实现 VSecNN 时，主要考虑了以下几个方面：

1) 神经网络通常涉及大量针对浮点数的运算。而基于有限域和椭圆曲线群的 MPC 和 zk-SNARK 协议只能高效支持定点数的运算。针对这种不兼容的问题，本方案采用合理的量化方法^[153]，将基于 PyTorch 训练的参数为浮点数的神经网络模型转换为参数为定点数的神经网络模型。

2) 基于 arkworks 库实现基础的 Groth16 算法，该库是一个 Rust 库，为各种 zk-SNARKs 协议提供了高效且安全的实现。我们选择了一个配对友好的椭圆曲线，BLS12-381，该曲线为基于椭圆曲线群的 Gorth16 和 MPC 协议提供 128 位（即 $\lambda=128$ ）的安全级别。

3) 选择算术秘密共享协议作为底层 MPC 协议，并基于 Rust 语言和 SPDZ 协议实现了 4.3.5 节中的五个操作，以实现：① 量化神经网络模型的安全多方推理，② 实现 Groth16 的多方证明生成。这些操作同样可以选择其他用于多方推理的 MPC 协议，如 SecureML^[54]、Cheetah^[79]、DELPHI^[72]。

4) 将量化后的神经网络模型基于 arkworks 库转换为 QAP 关系，然后将该关系应用于多方证明生成过程，从而实现可验证的多方神经网络推理。

实验配置：本章的实验是在 1 台或 2 台配备了英特尔酷睿 (TM) i7-10700k @ 3.8 GHz CPU、128GB 内存的 Ubuntu 20.04 操作系统电脑上进行的。

数据集和模型：本章方案的实验在四个具有不同规模的数据集上进行：WINE (Wine Data Set)、MNIST (Modified National Institute of Standards and Technology database)、CIFAR10 (Canadian Institute for Advanced Research, 10 classes)、ORL (Our Database of Faces)。相应的，在上述数据集上分别考虑了不同规模的神经网络架构，以测试具有不同数量级参数的神经网络模型的性能。其中，对 WINE 和 MNIST 数据集采用全连接神经网络；对 CIFAR10 和 ORL 数据集，则采用了 LeNet 的变体。数据集和模型架构的详细信息见表 4-2 和表 4-3。

表 4-2 数据集详情
Table 4-2 Details of the datasets

数据集	类别	特征数	样本数
WINE	3	13	178
MNIST	10	28*28*1	70000
Cifar10	10	32*32*3	60000
ORL	40	46*56*3	400

表 4-3 模型架构
Table 4-3 Model architectures

数据集	WINE	MNIST	CIFAR10	ORL
卷积层 1	/	/	卷积核 5×5 通道数 6	卷积核 5×5 通道数 6
池化层	/	/	卷积核 2×2	卷积核 2×2
卷积层 2	/	/	卷积核 5×5 通道数 16	卷积核 5×5 通道数 16
池化层	/	/	卷积核 2×2	卷积核 2×2
卷积层 3	/	/	卷积核 4×4 通道数 120	卷积核 4×4 通道数 120
全连接层 1	13×32	784×128	480×84	4800×84
全连接层 2	32×3	128×10	84×10	84×40

表 4-4 证明生成时间与验证时间对比
Table 4-4 Comparisons on proof generation time and verification time

模型	证明生成 (s)			验证时间(s)		
	ZEN	VNN (1)	VSecNN (2) ^a	ZEN	VNN (1)	VSecNN (2)
WINE	36.560	2.056	3.420	0.084	0.071	0.070
MNIST	714.381	108.590	227.044	0.386	0.165	0.169
CIFAR10	817.010	733.506	1401.045	0.317	7.129	7.057
ORL	6554.729	2752.191	4882.097	1.444	24.296	24.451

4.5.2 实验结果

由于本方案的主要工作在于对多方安全推理方法实现可验证性，因此本节性能测试主要集中在多方证明的生成和验证上。

4.5.2.1 证明生成时间与验证时间对比

首先，本节在通过局域网连接的两台电脑中模拟两方计算设置下的 VSecNN，后文称为 VSecNN(2)。为了与具有相同设置的单方证明生成相比较，在与 VSecNN 具有相同设置的基础上实现了一种可验证的单方神经网络推理方案，用 VNN(1)表示。随后对 VSecNN (2)、VNN (1) 和 ZEN^[116] 进行了比较。ZEN 基于 Groth16 实现了可验证的神经网络推理，这一点与 VSecNN 相似。但 ZEN 与 VNN(1) 的推理和证明生成过程是由单方完成的，因此无法保护推理数据的隐私。表 4-4 展示了 VSecNN(2)、VNN(1) 和 ZEN 在不同数据集和模型上的证明生成时间和验证时间的比较。结果表明，相比于 ZEN，VSecNN(2) 在提供了对推理数据隐私保护的前提下，证明生成的性能依然具有显著的提高。而相比于 VNN(1)，VSecNN(2) 在两个服务器合作生成证明的情况下，效率仍然在可接受的范围内，计算时间并没有大幅增加。这得益于 VSecNN 实现过程中的优化。例如，将常用的基于有限域的 MPC 协议转换为基于椭圆曲线群的协议，以使其与 Groth16 协议更加兼容。此外，相比于 Pedersen 承诺，本方案采用了更适

合处理批量参数的 KZG 多项式承诺，因此更加适用于神经网络推理过程。在验证时间方面，VSecNN 中多方生成的证明的验证时间相比于 ZEN 有着小幅的增加，与 VNN(1)单方生成证明的验证时间相当。

4.5.2.2 服务器数量对效率的影响

本节探讨了服务器数量对 VSecNN 证明生成时间的影响。如表 4-5 所示，服务器数量的增加对证明生成时间的影响可以忽略不计。然而，随着服务器数量的增加，通信开销会显著增加。主要原因是在证明生成阶段前，服务器需要合作构造约束系统，其中通信为该过程的主要成本。因此，虽然可以通过增加服务器数量来提高方案抵抗合谋攻击的能力，但需要在安全性与效率之间进行权衡。因此，在未来的工作中需要对服务器数量增加导致的通信开销增长问题进行研究。

表 4-5 服务器数量的影响

Table 4-5 Impact of number of servers

模型	WINE		MNIST		CIFAR10		ORL	
	证明生成 (s)	通信开销 (MB)	证明生成 (s)	通信开销 (MB)	证明生成 (s)	通信开销 (GB)	证明生成 (s)	通信开销 (GB)
VSecNN (2)	3.420	0.619	227.044	124.038	1401.045	0.901	4882.097	3.407
VSecNN (3)	3.484	1.856	227.902	372.115	1442.405	2.704	4903.579	10.220
VSecNN (4)	3.478	3.713	223.932	744.231	1411.739	5.407	4895.861	20.439
VSecNN (5)	3.558	6.188	228.275	1240.384	1417.225	9.012	4901.937	34.065

表 4-6 约束数量 (K)

Table 4-6 Number of constraints (K)

层次	WINE	MNIST	CIFAR10	ORL
卷积层 1	/	/	1415.904	1323.504
ReLU	/	/	9.408	26.208
池化层	/	/	3.528	9.828
卷积层 2	/	/	961.600	3596.384
ReLU	/	/	3.200	11.968
池化层	/	/	1.200	4.224
卷积层 3	/	/	492.000	4920.000
ReLU	/	/	0.960	9.600
全连接层 1	1.696	401.536	161.364	1612.884
ReLU	0.064	0.256	0.168	0.168
全连接层 2	0.387	5.130	3.370	13.480

4.5.2.3 约束数量

约束的数量是评估证明计算复杂度的关键指标，它描述了计算的输入、输出和中间结果之间的约束关系，同时将这些关系融入到证明中。因此，证明生成过程的效率主要取决于约束的数量和每个约束在证明中的计算成本。如表 4-6 所示，本节给出了不同模型和数据集下神经网络每一层的约束数量。研究结果

为进一步研究优化可验证神经网络推理方案约束数量研究奠定了基础，特别是对于卷积层的约束数量。例如，在 LeNet-CIFAR10 中，第一层卷积（Con1）需要 1415.904K 个约束，而随后的池化层只需要 3.528K 个约束。此外，约束数量主要受底层计算电路与约束构造方式的影响，因此在服务器数量不同的情况下约束数量保持一致。

4.5.2.4 CRS 和证明大小

本节对比了 VSecNN、vCNN^[115]和 pvCNN^[118]的 CRS 和证明大小。vCNN 是一种基于 zk-SNARK 协议的可验证神经网络推理方案，采用二次多项式程序（Quadratic Polynomial Program, QPP），但缺乏对推理数据的隐私保护。pvCNN 提供了一种隐私保护和可验证的 CNN 合作推理方案，但是，pvCNN 将部分模型设置为非敏感信息，外包给服务器进行计算，并采用零知识证明验证外包计算的正确性。而被设定为敏感信息的部分模型则在本地执行推理，基于同态加密实现数据的隐私保护。因此，该方案的证明生成仅限于推理计算的部分过程，且缺少对整体模型的隐私保护。由于实现基准不同，导致很难对上述方案进行整体实现并进行对比。因此，本节主要比较 CRS 和证明的大小，因为该部分独立于实现方法和测试环境。如表 4-7 所示，在 LeNet-CIFAR10 的推理测试表明，VSecNN 在证明大小方面具有较大的优势，这有利于构建用户友好的验证系统。在 CRS 大小方面，VSecNN 远小于 pvCNN；但相比于 vCNN 具有较大的劣势。然而，vCNN 仅实现了单方证明生成，无法保证推断数据和推理结果的隐私保护。

表 4-7 CRS 和证明大小
Table 4-7 CRS and Proof Size

方案	CRS 大小	Proof 大小
vCNN	40.07 MB	2803 B
pvCNN	14.30 TB	343.19 MB
VSecNN	5.48 GB	432 B

4.6 本章小结

本章提出了一种支持隐私保护和可验证的神经网络推理方案-VSecNN。首先结合 MPC、Groth16 协议以及 KZG 多项式承诺，实现了多方协作生成 Groth16 的证明。为了兼容基于椭圆曲线群的 Groth16 证明生成过程，本方案通过将基于有限域的 MPC 协议替换为基于椭圆曲线群的 MPC 协议，从而实现了更高效的多方证明生成。随后，本方案将上述多方证明生成过程融入到安全多方推理中，以实现具有隐私保护及可验证的神经网络安全推理。最后，本章开展了仿真实验以评估 VSecNN 的性能。

本方案展示了在保护神经网络模型、推理数据、推理结果隐私的前提下，实现推理过程可验证的可能性。然而，将神经网络推理与 Groth16 和 MPC 协议进行结合会造成较大的计算开销及内存开销，仍然需要进一步的研究与优化。后续的研究重点在于如何通过优化约束数量、每个约束在证明生成中的计算时间等，以进一步提高效率。

第 5 章 可验证神经网络隐私保护加密推理方案

5.1 问题描述

在神经网络推理过程中，已有方案大多采用安全多方计算与同态加密实现推理过程中数据的隐私保护，以适应不同的场景需求。由于有限级数同态加密的计算深度限制以及转换为全同态加密（Fully Homomorphic Encryption, FHE）的产生的昂贵的计算开销等问题，使其难以应用于大型神经网络模型的推理中因此在当前的研究进展与实际应用中均滞后于基于安全多方计算的神经网络安全推理。但在一些特定场景或需求下，需要采用全同态加密（Fully Homomorphic Encryption, FHE）或有限级数同态加密（Leveled Fully Homomorphic Encryption, LHE）而非安全多方计算作为底层技术^[106, 109]。首先，基于安全多方计算的神经网络安全推理需要较大的通信开销，在网络环境较差的场景中无法稳定运行；其次如果选择 N 个第三方服务器执行多方推理，则无法抵抗 N 个服务器的合谋攻击。而当数据所有者和模型所有者也作为参与方执行多方推理，则无法很好的适用于资源受限的用户。而同态加密不需要多方交互计算，避开了多方合谋的问题，以提供更高的安全性。因此需要采用同态加密技术，实现神经网络加密推理，以更加稳定、安全、用户友好的方式，实现神经网络推理中数据和模型的隐私保护。

以第 4 章中所提到的 AI 医生场景为例：为了确保 AI 医生诊断过程的隐私保护和可验证性，患者的隐私数据及 AI 医生的模型需要以秘密份额的方式发送给 N 个服务器，并通过安全多方计算的方式执行推理和零知识证明过程。因此首先需要确定 N 个服务器的来源，由于患者可能并不持有执行模型推理和证明生成所需的计算和通信资源，因此往往选择额外的服务器来完成计算过程。但安全多方计算仅能保证 $N-1$ 个服务器的合谋攻击，当 N 个服务器合谋时，患者的隐私信息及模型信息就可以被完全恢复。因此在这种场景下，如果患者没有足够的计算和通信资源参与到推理过程，其隐私数据仍然面临着泄露的威胁。

在基于同态加密的安全推理中，通常设置执行加密推理的服务器为被动敌手模型（也称半诚实敌手模型），在这个假设中，服务器虽然会试图挖掘用户的隐私信息，但也会按照既定的协议执行加密推理过程，并返回正确的计算结果。然而，在实际应用中，可能存在服务器为恶意或受到恶意攻击的情况，从而破坏推理结果的正确性或带来安全威胁。因此，与第 4 章中提出的问题类似，在基于同态加密的神经网络加密推理中，同样也面临着推理结果不可验证的问题。因此，实现加密推理的可验证性，可以使得服务器满足主动敌手模型（也称恶

意敌手模型), 对于解决下列挑战至关重要:

1) 正确性: 在基于同态加密的神经网络推理中, 服务器可能会返回不正确的推理结果。推理结果的正确性主要包括两个方面, 一方面, 恶意服务器没有正确的执行推理过程或在推理中采用了不一致的模型, 从而导致错误的推理结果; 另一方面, 服务器由于计算失误(如超出预期的噪声溢出)产生了错误的密文, 导致无法正确解密。这种错误可以通过按照计算电路合理的设置安全参数来进行规避。

2) 机密性: 恶意服务器可能会利用针对同态加密的密钥恢复攻击^[154-156]破坏用户数据的机密性。这超出了现有基于同态加密的神经网络安全推理方案中假定的被动敌手模型设置的范围。密钥恢复攻击通过构造错误的特殊密文, 并利用用户对特殊密文的解密失败后做出的反应(例如, 通过请求重新运行计算或终止进一步交互)恢复部分或全部用户私钥, 从而造成用户数据的隐私泄露。

5.1.1 存在的挑战

为了实现基于同态加密的神经网络推理的可验证性, 关键点在于构造满足主动敌手模型的可验证同态加密方案。当前已有针对同态加密可验证的研究, 主要包括基于消息认证码(Message Authentication Code, MAC)、可信执行环境(Trusted Execution Environment, TEE)、零知识证明(Zero-Knowledge Proof, ZKP)三类底层技术的方法。消息认证码通常被用于验证传统对称加密的完整性。在同态加密中, 需要采用同态 MAC, 使得服务器能够在对密文进行同态操作时, 将输入密文的有效 MAC 转换为输出密文的有效 MAC。然而, 现有基于 MAC 的解决方案仅支持半同态加密或部分同态加密(如仅支持一次同态乘法操作), 尚不清楚目前是否存在完全支持常用的全同态加密操作及密文维护操作(如重新线化)的同态 MAC 方法^[157-159]。TEE 提供了一个受保护的执行环境, 允许在硬件级别执行同态计算, 因此通常具有较高的计算效率。然而基于 TEE 的安全性对执行硬件有较强的依赖性。此外, 同态加密较高的计算复杂度, 特别是在包含大量同态操作的神经网络推理中, 相比于不可信的底层硬件, 对 TEE 在内存和计算能力等方面具有更高的要求^[160]。零知识证明作为实现可验证同态加密的潜在解决方案, 近年来得到了研究者的关注。然而, 现有基于 ZKP 的可验证同态加密方案仅支持加法同态加密如 Paillier 同态加密^[161]或简单的有限级数同态加密如 BV(Brakerski-Vaikuntanathan) 同态加密^[162, 163], 无法很好的应用于神经网络推理中。

因此, 本章方案的目标是探索 ZKP 与神经网络推理中常用的同态加密方案如 BGV(Brakerski-Gentry- Vaikuntanathan)、BFV(Brakerski-Fan-Vercauteren)、

CKKS (Cheon-Kim-Kim-Song) 结合的可能性，以实现适用于神经网络加密推理的可验证同态加密方案。然而，在构造可验证神经网络隐私保护加密推理方案时，存在一些挑战亟待解决：1) 构造可验证同态加密方案。在基于同态加密的计算中，复杂的同态操作会导致计算电路的改变，原有的底层加法和乘法需要由基于环多项式的同态加法和同态乘法实现，并且需要复杂的重线性化、密钥交换等操作。而 zk-SNARKs 协议当前大多面向基于有限域的算术电路或布尔电路，因此基于 zk-SNARKs 为基于环多项式算术电路的同态加密计算生成零知识证明成为一大挑战。2) 可验证同态加密方案与神经网络推理的结合。由于神经网络推理中存在一些非线性运算，无法将其直接构造为算术电路，因此将可验证同态加密方法用于神经网络推理过程中面临着不兼容的问题，需要对神经网络加密推理过程进行适应性的优化。3) 计算效率问题。由于 zk-SNARKs 非交互式的特点，在计算针对神经网络推理过程的零知识证明时，需要对推理过程中所有的计算生成约束并包含在证明中。而神经网络推理过程又包含了大量基础运算，因此证明中会包含大量约束，造成较大的计算开销。这是当前基于 zk-SNARKs 方案的可验证神经网络推理普遍面临的问题，而结合同态加密后，每个基础运算都会扩展为同态运算，计算效率问题会更为突出。

5.1.2 设计目标与方法

本章方案首先结合同态加密和零知识证明，构造可验证同态加密方案。出于对用户友好的角度，本章方案依旧选择 Groth16 作为底层的 ZKP 协议，由于 CKKS 支持浮点数运算，而 ZKP 协议通常支持定点数运算，二者结合可能会更加困难，因此选择另一种神经网络推理中常用的有限级数同态加密方案，BGV，作为底层方案。由于 Groth16 协议只能高效地支持基于有限域的算术电路的证明生成，而基于 BGV 的神经网络推理过程通常被表示为基于环多项式的电路，因此无法直接基于 Groth16 生成环多项式的电路的证明。

本方案的设计参考 Rinocchio^[123]，一种用于环上电路的简洁非交互知识论证，将 Groth16 转换为支持环上计算的协议（后续称为 R-Groth16）。具体的，在 Groth16 基础构造中，需要首先将待证明的算术电路（如由神经网络推理过程转换的计算电路）转换为 QAP 关系。而为了支持环多项式上的计算，Rinocchio 中提出了二次环程序（Quadratic Ring Program, QRP）和环上的安全编码。通过将环上的电路转换为 QRP 关系，并基于该 QRP 关系生成与基于 QAP 关系的 Groth16 协议相似的证明，从而实现基于 BGV 的神经网络推理过程的可验证。随后，将可验证同态加密方案与神经网络推理相结合，以构造可验证的加密推理方案。

5.1.3 本章贡献

本章主要贡献总结如下：

- 1) 基于环上电路的 zk-SNARKs 协议，提出了同态加密方案中的乘法、比特分解等重要运算到 QRP 的转换方法，从而构造可验证同态加密方案。
- 2) 将可验证同态加密方案结合到神经网络加密推理中，并对推理中的非线性计算进行了适应性的调整，实现了满足模型、推理数据、推理结果隐私保护以及模型真实性和推理正确性可验证的神经网络加密推理方案 VHENN。
- 3) 在实验评估过程中，通过采用 SIMD 技术，降低 zk-SNARKs 证明系统的约束数量，从而提升可信设置、证明生成和验证的效率。

5.2 方案概述

5.2.1 系统模型

本方案采用 Commit-and-Prove 证明系统，首先由证明方通过承诺协议对所持有的“秘密”进行承诺，在神经网络推理场景下，该“秘密”特指神经网络参数。随后，证明某结果是由承诺的参数经过执行特定计算得到的。也就是，证明推理结果是由所承诺的神经网络经过前向传播算法所得。本文从神经网络加密推理和证明系统的构造两方面进行描述，主要由三类实体组成，包括可信第三方、模型所有者、数据所有者。对每个实体的描述如下：

- 1) 可信第三方：可信第三方在初始化阶段生成零知识证明协议的公共参考字符串（Common Reference String, CRS）和承诺协议的结构化参考字符串（Structured Reference String, SRS）。用于生成 CRS 和 SRS 的密钥必须保密，并在初始化过程结束后销毁。
- 2) 模型所有者：模型所有者首先对其持有的模型参数进行承诺，并负责生成神经网络推理计算和承诺计算对应的 QRP 关系。在接收到数据所有者发送的加密数据后，执行加密推理和证明生成过程。
- 3) 数据所有者：将持有的待推理数据利用同态加密方案进行加密，并发送给模型所有者。在接收到加密的推理结果及其对应的证明后，对推理结果进行验证，并进行解密。

5.2.2 威胁和安全模型

在传统基于同态加密的神经网络安全推理中，通常设置执行加密推理的服务器为被动敌手模型（也称半诚实敌手模型），在这个假设中，服务器虽然会试图挖掘用户的隐私信息，但也会按照既定的协议执行加密推理过程，并返回正

确的计算结果。然而，在实际应用中，可能存在服务器为恶意或受到恶意攻击的情况，从而破坏推理结果的正确性或带来安全威胁。因此，在基于同态加密的神经网络加密推理中，同样也面临着推理结果不可验证的问题。本方案将模型所有者设定为主动敌手模型，它不仅会试图挖掘用户推理数据和推理结果中的隐私信息，还可能执行错误的协议，从而得到错误的推理结果。另一方面，本方案为数据所有者设定被动敌手模型。他们可能试图从神经网络模型或推理的数据中提取隐私信息，但会遵循既定的协议，不会主动参与操纵、干扰或修改数据。

本方案主要关注两个阶段的隐私保护：加密推理阶段和加密推理证明阶段。在加密推理阶段，主要包括推理数据和推理结果的隐私保护；而在多方证明生成阶段，则侧重于模型的隐私保护和验证的正确性。

1) 在加密推理阶段，推理数据和推理结果的机密性依赖于同态加密方案的安全性。BGV 同态加密方案提供选择明文攻击（Security under Chosen Plaintext Attack, CPA）下的安全性，即使攻击者可以选择任意明文并获取其对应的密文，仍然无法从密文中恢复出与明文相关的任何信息，这是通过在加密过程中加入噪声来实现的。

2) 在加密推理证明阶段，模型的机密性与验证的正确性依赖于零知识证明协议的安全性。设 Γ 为本方案的加密推理证明生成协议，如果协议 Γ 满足以下定义的完备性（Completeness）、知识合理性（Knowledge Soundness）和零知识性（Zero-Knowledge），则被认为是安全的。

定义 5-1（完备性）。持有声明和见证的模型所有者可以生成一个证明。在验证这个证明时，验证者输出 0 的概率 $negl(\lambda)$ 是可以忽略不计的，其中 λ 是安全参数。

定义 5-2（知识合理性）。对于计算能力有限且不持有见证的敌手，存在一个计算能力有限的提取器 ε ，它可以完全访问敌手的状态。每当敌手生成一个有效的证明时，提取器 ε 就可以计算出一个相应的见证，使得 $(x, w) \notin R$ 和证明 π 说服验证者的概率可以忽略不计。

定义 5-3（零知识性）。存在一个模拟器 S ，可以在不依赖见证的情况下生成与真实证明无法区分的模拟证明。即对于一个计算能力有限的敌手，它可以以可忽略不计的概率 $negl(\lambda)$ 区分真实证明和模拟证明。

5.2.3 方案流程

为了实现可验证的神经网络加密推理，本方案首先基于环上运算的 zk-SNARK 方案，构造可验证的 BGV 同态加密方案。随后，基于可验证 BGV，

完成 VHENN 的方案设计。如图 5-1 所示，VHENN 主要包括：模型承诺、初始化、加密推理、加密推理的证明生成、验证与解密。每个步骤的说明如下：

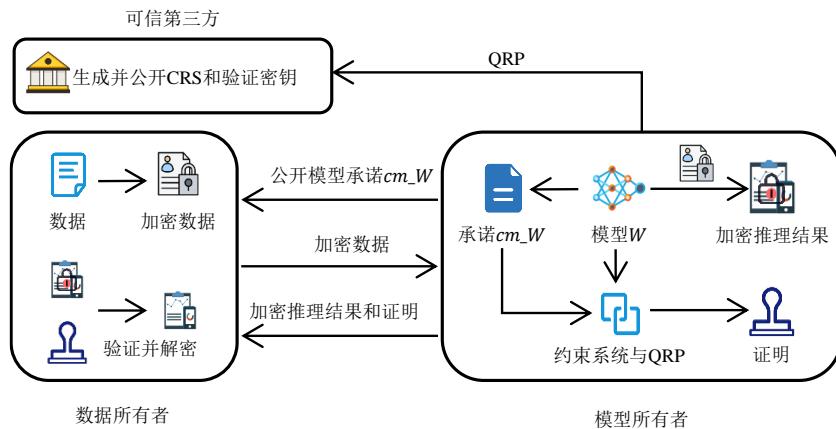


图 5-1 VHENN 方案流程
Fig.5-1 Overview of VHENN Workflow

- 模型承诺:** 模型所有者对所持有的模型进行承诺，并将生成的承诺公开。该步骤保证在后续的推理服务中，模型所有者始终采用了同一模型。
- 初始化:** 在初始化阶段，首先构造加密推理的 QRP，具体的，将神经网络前向传播中的每个基础操作转换为同态操作，并为其构造子电路，生成对应的 QRP。然后将模型参数承诺的计算过程生成 QRP。最后，将这些子电路的 QRP 进行连接，最终生成整个电路的 QRP。随后执行 R-Groth16 协议和 BGV 同态加密的设置阶段，生成必要的参数如 R-Groth16 的 CRS 和 BGV 的公共参数。其中 R-Groth16 的设置为可信设置，需要由可信第三方执行，BGV 的设置则无需额外设置可信第三方。
- 加密推理:** 数据所有者将待推理数据进行加密，发送给模型所有者。模型所有者按照 BGV 同态加密的计算方法，实现对加密数据的神经网络推理过程，得到加密的推理结果并发送给数据所有者。该过程保护了数据所有者的数据及推理结果的机密性。
- 加密推理的证明生成:** 基于神经网络加密推理的 QRP 关系，完成加密推理的证明生成，并将证明发送给数据所有者。
- 推理结果的验证与解密:** 收到加密推理结果和相应的证明后，用户对其进行验证及解密。如果验证失败，则表明模型所有者没有遵守约定的计算协议，或者在推理过程中使用了不正确的模型参数。

5.3 可验证 BGV

本节对基于 RLWE 的 BGV 中同态加法与同态乘法操作的 QRP 构造与证明

生成过程进行描述，主要涉及利用环上乘法的 QRP 构造实现环上多项式乘法的 QRP 构造过程。首先，记 BGV 中的明文为 $PT \in R_t$ ，密文为 $CT \in R_q$ 。BGV 中的同态操作主要包含密文加法 $Add(pk, CT_1, CT_2) \rightarrow CT_3$ 和密文乘法 $Mul(pk, CT_1, CT_2) \rightarrow CT_3$ 。对于密文加法，只需要将 CT_1 和 CT_2 中的对应组分相加，也就是对 $CT_1 = (c_{11}, c_{21}) \in R_q^2$ 和 $CT_2 = (c_{12}, c_{22}) \in R_q^2$ ，其中 $R = \mathbb{Z}[X]/(X^d + 1)$ ， $R_q = R/qR$ 。计算 $c_{13} = c_{11} + c_{12}$ ， $c_{23} = c_{21} + c_{22}$ ，相当于计算两个度至多为 $d - 1$ 的多项式的加法运算，只需要将各项系数相加即可。因此实现同态加密操作的可验证性，主要工作集中于密文乘法操作的证明构造。

如 2.2.2 节所述，执行密文乘法包括如下操作：

- 1) $CT_3 = CT_1 \otimes CT_2$ ，张量积操作由环多项式乘法实现。
- 2) $CT_4 \leftarrow SwitchKey(CT_3, \tau_{sk_j \rightarrow sk_{j-1}}, q_{j-1})$ ， $CT_4 = BitDecomp(CT_3)^T \cdot \tau_{sk_j \rightarrow sk_{j-1}}$ ，密钥交换计算由比特分解计算及一个高维向量 ($4\lceil \log q_j \rceil$) 与高维矩阵 ($4\lceil \log q_j \rceil \times 2$) 的乘积，包括环多项式乘法、环多项式加法，组成。
- 3) $CT_5 \leftarrow Scale(CT_4, q_j, q_{j-1})$ 。 $CT_5 = q_{j-1} / q_j \cdot CT_4$ ，该操作包含 $4(\lfloor \log q_j \rfloor + 1)$ 个常数与环多项式乘法。

如果采用支持有限域运算的零知识证明（如基于 QAP 的 Groth16 协议）对上述操作生成证明，在支持有限域的零知识证明方案中，环上多项式加法和常数乘法需要引入额外的约束，而基于 QRP 的 R-Groth16 方案支持环上的运算，加法门和常数的乘法门不需要在 QRP 中引入额外的约束，因此我们主要考虑多项式乘法和比特分解的 QRP 构造方法，以实现 BGV 中同态操作的 QRP 构造。

5.3.1 环多项式乘法 QRP 构造

直观上，多项式乘法可以简单的由多项式各项相乘得到，然而，对两个次数为 n 的多项式直接相乘时间复杂度为 $O(n^2)$ ，在处理大规模数据时，传统多项式乘法计算会变得非常低效。数论变换（Number Theoretic Transform, NTT）提供了一种快速计算多项式乘法的方法，可以将时间复杂度降低为 $O(n \log n)$ 。通过 NTT 对环多项式乘法进行优化，可以将多项式乘法转换为一系列加法门、乘法门。

首先，根据 2.4.3 节中对乘法门的 QRP 构造方法实现对单个乘法门的 QRP 构造。随后，对于两个输出线和输入线首尾相连的算术电路 \mathcal{C}_1 和 \mathcal{C}_2 ， I_1 和 I_2 为电路中线的索引， $I_1 \cap I_2$ 表示 \mathcal{C}_1 中输出线作为 \mathcal{C}_2 中输入线的部分。对于 $i \in \{1, 2\}$ ， $Q_i = (\{U^{(i)} = \{u_k^{(i)}(X) : k \in I_i\}, V^{(i)} = \{v_k^{(i)}(X) : k \in I_i\}, W^{(i)} = \{w_k^{(i)}(X) : k \in I_i\}\}, t^{(i)}(X))$ 为相应的 QRP。那么， $Q = Q_2 \circ Q_1$ 表示为电路 $\mathcal{C} = \mathcal{C}_1 \circ \mathcal{C}_2$ 的 QRP，其中，

$U = \{u_k(X) : k \in I_1 \cup I_2\}$, $V = \{v_k(X) : k \in I_1 \cup I_2\}$, $W = \{w_k(X) : k \in I_1 \cup I_2\}$ 。对于目标多项式，首先将其表示为 $t(X) = t^{(1)}(X) \cdot t^{(2)}(X)$ ，随后，对于所有的电路线索引 $\tilde{k} \in I_2 \setminus I_1$ ，设置 Q_1 中的对应多项式为 $u_{\tilde{k}}^{(1)}(X) = v_{\tilde{k}}^{(1)}(X) = w_{\tilde{k}}^{(1)}(X) = 0$ 。同样的，对于 $\tilde{k} \in I_1 \setminus I_2$ ，设置 Q_2 中的对应多项式为 $u_{\tilde{k}}^{(2)}(X) = v_{\tilde{k}}^{(2)}(X) = w_{\tilde{k}}^{(2)}(X) = 0$ 。对于 $k \in I_1 \cup I_2$ ，只要目标多项式没有共同的根，就可以满足下列模的等价性设置： $u_k(X) \equiv u_k^{(i)}(X) \bmod t^{(i)}(x), v_k(X) \equiv v_k^{(i)}(X) \bmod t^{(i)}(x), w_k(X) \equiv w_k^{(i)}(X) \bmod t^{(i)}(x)$ 。最后可通过单个乘法门的 QRP 构造出环多项式乘法对应的整个电路的 QRP。

5.3.2 比特分解 QRP 构造

在比特分解电路中，对于输入 $CT \in R_q$ ，通过比特分解得到 CT 的二进制表示 $(a_1, \dots, a_{\lceil \log q \rceil}) \in R_2^{\lceil \log q \rceil}$ ，使得 $CT = \sum_{i=1}^{\lceil \log q \rceil} 2^{i-1} \cdot a_i$ 。首先，将输出线标记为 $1, \dots, \lceil \log q \rceil$ ，输入线标记为 $\lceil \log q \rceil + 1$ 。设 $t(X) = (X - r) \prod_{i=1}^{\lceil \log q \rceil} (X - r_i)$ ，其中 $r, r_1, \dots, r_{\lceil \log q \rceil} \in A$ ， A 为特殊集。

对于 $1 \leq i \leq \lceil \log q \rceil$ ，设：

$$\begin{aligned} u_0(r) &= 0, u_i(r) = 2^{i-1}, u_{\lceil \log q \rceil + 1}(r) = 0 \\ v_0(r) &= 1, v_i(r) = 0, v_{\lceil \log q \rceil + 1}(r) = 0 \\ w_0(r) &= 0, w_i(r) = 0, w_{\lceil \log q \rceil + 1}(r) = 1 \end{aligned} \quad (5-1)$$

对于 $1 \leq j \leq \lceil \log q \rceil$ ，设

$$\forall i \neq j, \quad u_j(r_j) = 1, u_i(r_j) = 0 \quad (5-2)$$

$$\forall i \neq 0, i \neq j, \quad v_0(r_j) = 1, v_j(r_j) = -1, v_i(r_j) = 0 \quad (5-3)$$

$$\forall i, \quad w_i(r_j) = 0 \quad (5-4)$$

若 $(u_0(X) + \sum_{k=1}^{\lceil \log q \rceil} a_k \cdot u_k(X)) \cdot (v_0(X) + \sum_{k=1}^{\lceil \log q \rceil} a_k \cdot v_k(X)) - (w_0(X) + \sum_{k=1}^{\lceil \log q \rceil} a_k \cdot w_k(X))$ 可以被 $t(X)$ 整除，则该式在 r 点处的值一定为 0，因此公式 (5-1) 可以保证 $CT = \sum_{i=1}^{\lceil \log q \rceil} 2^{i-1} \cdot a_i$ 。公式 (5-2) - (5-4) 可以保证每个 r_i 都是多项式 $t(X)$ 的一个根，使得 $a_j(1 - a_j) = 0$ ，因此保证 $(a_1, \dots, a_{\lceil \log q \rceil})$ 是 CT 的二进制表示。

5.4 VHENN 方案设计

本章主要基于验证模块设计 VHENN 方案，首先为神经网络加密推理过程生成 QRP 关系；随后基于 BGV 实现神经网络加密推理，基于 R-Groth16 和可验证 BGV 实现证明的生成；最后是推理结果的验证与解密。

5.4.1 神经网络 QRP 构造

模型所有者首先对持有的模型进行承诺，并将承诺公开，并且在后续的计

算中，将承诺的运算加入到证明的约束系统中，从而保证在后续的推理过程中，采用的模型始终为最初所承诺的模型。在第四章中，介绍了基于 Groth16 和 MPC 的神经网络 QAP 的构造，将神经网络的前向传播的计算 $f(d, W)$ 和承诺协议的计算转换为算术电路，随后将算术电路转换为 R1CS，最后得到神经网络的 QAP。与第四章中神经网络 QAP 构造不同，在基于环多项式的加密推理中，神经网络的前向传播计算为加密计算，其中的算术运算是由同态操作（如同态加法、同态乘法）实现的。如 2.2.2 节所描述，每个同态乘法需要由若干复杂的环多项式操作表示。因此，首先需要将神经网络前向传播中的每个加法转换为多项式加法，乘法转换为由环多项式乘法、比特分解等操作组成的子电路 C_x ，生成对应的 QRP。随后，将这些子电路的 QRP 按照 2.4.3 节所述的方法进行连接，最终生成整个电路的 QRP。

在神经网络网络加密推理中，存在两类应用场景。一类是模型持有者与数据持有者为不同方，模型持有者为数据持有者提供推理服务，加密推理计算由模型持有者执行，在后续称为加密推理服务。另一类是模型持有者与数据持有者为同一方，加密推理被外包给云服务器执行，称为加密推理外包。在本章方案中，主要面向加密推理服务，下面给出加密推理服务场景下的神经网络 QRP 构造过程，并对加密推理外包场景下的神经网络 QRP 构造进行了简要说明。

5.5.1.1 加密推理服务

在此类场景中，模型持有者为数据所有者提供推理服务，加密推理计算由模型持有者执行，输入的加密推理数据、模型的承诺值以及输出的加密推理结果对数据持有者（即验证者）是公开的，因此被指定为声明（Statement）。剩余部分包括模型权重、偏置、加密的中间值等信息，对数据所有者是保密的，因此被指定为见证（Witness）。最终，我们可以得到一个 QRP 关系，表示为：

$$Q = (R_q, l, \{u_k(X), v_k(X), w_k(X)\}_{k=0}^m, t(X)) \quad (5-5)$$

该关系满足下列条件：

- $\{u_k(X), v_k(X), w_k(X)\}_{k=0}^m$ 是度为 $n-1$ 的 QRP 多项式，其中 n 为约束数量。QRP 多项式与神经网络模型的加密推理过程和模型参数的承诺计算过程相关联。
- $t(X)$ 是度为 n 的目标多项式。
- $x = (a_1, \dots, a_l)$ 为声明，包括输入的加密数据 $\llbracket d \rrbracket$ ，模型 W 的承诺值 cm_W 以及加密的推理结果 $\llbracket y \rrbracket$ 。其中， $\llbracket \cdot \rrbracket$ 表示加密数据， $\llbracket d \rrbracket, \llbracket y \rrbracket \in R_i$ ， $i \in [0, L]$ 为密文的层级，承诺值 $cm_W \in R_t$ ， R_t 为明文空间。
- $w = (a_{l+1}, \dots, a_m)$ 为见证，包括模型 W 的参数，如权重和偏置，以及在推

理过程中生成的加密的中间结果，即每层网络的输出。其中， W 的参数属于 R_t ，加密的中间结果属于 R_i 。

- 对于 $(x, w) \in Q$, $p(X) = \sum_{k=0}^m a_k u_k(X) \cdot \sum_{k=0}^m a_k v_k(X) - \sum_{k=0}^m a_k w_k(X)$, 其中 $a_0 = 1$, $p(X)$ 被 $t(X)$ 整除。

通过上述过程，可以得到一个满足相应声明和见证的 QRP 关系 Q 。模型所有者可以基于关系 Q 生成一个证明，以说服数据所有者，他拥有一个神经网络模型 W ，加密的推理结果 $\llbracket y \rrbracket$ 是通过对数据所有者的加密数据 $\llbracket d \rrbracket$ 在模型 W 上进行推理得到的。

5.5.1.2 加密推理外包 QRP 构造

在此类场景中，数据、模型均被加密发送给服务器。对于验证者（即模型和数据持有者），加密推理过程的数据均无需保证“零知识”。因此，输入的加密推理数据、输出的加密推理结果、模型参数和推理计算的中间值均被指定为声明，且无需对模型进行承诺。整个证明系统实际为简洁非交互知识论证。在加密推理中，推理数与模型参数均为密文形式，因此电路中大多为密文与密文的乘法运算。最终，我们可以得到一个 QRP 关系，表示为：

$$Q = (R_q, \{u_k(X), v_k(X), w_k(X)\}_{k=0}^m, t(X)) \quad (5-6)$$

该关系满足下列条件：

- $\{u_k(X), v_k(X), w_k(X)\}_{k=0}^m$ 为 QRP 多项式，其与神经网络模型的加密推理过程相关联。
- $t(X)$ 是度为 n 的目标多项式。
- $a = (a_1, \dots, a_m)$ 为声明，包括输入的加密数据 $\llbracket d \rrbracket$ ，加密的模型参数 $\llbracket W \rrbracket$ ，加密的中间计算结果，加密的推理结果 $\llbracket y \rrbracket$ 。
- 对于 $a \in Q$, $p(X) = \sum_{k=0}^m a_k u_k(X) \cdot \sum_{k=0}^m a_k v_k(X) - \sum_{k=0}^m a_k w_k(X)$, 其中 $a_0 = 1$, $p(X)$ 被 $t(X)$ 整除。

通过上述过程，外包服务器可以基于关系 Q 生成一个证明，以说服验证者，加密的推理结果 $\llbracket y \rrbracket$ 是通过对验证者的加密数据 $\llbracket d \rrbracket$ 在加密的模型 $\llbracket W \rrbracket$ 上进行推理得到的。

对于神经网络模型中的非线性运算，可以采用两种方式，一种是与 VSecNN 同样的处理方法，即采用多项式近似方法将激活函数近似为多项式表示，对多项式运算生成 QRP。另一种是基于混淆电路实现对 ReLU 函数的比较操作，并生成比较电路的 QRP。对于比较电路的 QRP 生成，可以通过添加辅助参数，在推理过程中，除了输出 ReLU 的结果外，增加一个布尔值的输出用于表示所比较的参数与 0 的大小关系，从而将比较运算转换为算术运算。例如，假设密

文 $\llbracket x \rrbracket$ 为 ReLU 函数的输入，在推理过程中，若 $x > 0$ ，则输出 $\llbracket x \rrbracket$ 及 $tmp = 1$ 。若 $x \leq 0$ ，则输出 $\llbracket 0 \rrbracket$ 及 $tmp = 0$ 。在构造 QRP 时，将运算表示为 $\llbracket x' \rrbracket = tmp \cdot \llbracket x \rrbracket$ ，即可以保证比较运算的正确性。

5.4.2 加密推理

基于同态加密的神经网络推理，主要分为加密推理服务和加密推理外包两类。在加密推理服务与加密推理外包的区别在于，加密推理服务中待推理的数据为密文形式，模型为明文形式。而加密推理外包中待推理数据与模型均为密文形式。由于这两类推理模式的过程类似，这里仅对加密推理服务过程进行描述，后续统一称为加密推理。

在神经网络加密推理中，主要的运算包括密文加法、明文与密文乘法、密文与密文乘法。我们将遵循同态加密运算的协议表示为 Π ，则基于同态加密的加密推理方案可表示为 $\Pi(f\llbracket d \rrbracket, W) \rightarrow \llbracket y \rrbracket$ ，其中函数 f 表示前向传播算法， $\llbracket d \rrbracket$ 表示加密的数据， W 表示模型参数， $\llbracket y \rrbracket$ 表示加密的推理结果。

在神经网络的线性计算层，如卷积层和全连接层，基础运算主要包括向量和矩阵点积，哈达玛(Hadamard)积等。这些基础运算均可以由加法和乘法实现。因此线性层的计算在加密推理中可以直接转换为同态加法和乘法操作。而对于神经网络的非线性计算，如激活函数的计算，则无法直接采用 BGV 同态加密进行实现。为了实现神经网络非线性层的计算，可以采用两种方式。第一种是多项式近似方法，将非线性激活函数近似为线性的多项式。第二种则是采用安全多方计算，假设通过线性层的计算得到了一个加密的中间值 $\llbracket z \rrbracket$ ，服务器首先选择一个随机数 r ，使用同态加密的公钥 pk 加密随机数 r ，得到 $\llbracket r \rrbracket$ 。随后，服务器计算 $\llbracket z \rrbracket - \llbracket r \rrbracket$ 并将结果发送给数据所有者。数据所有者解密 $\llbracket z \rrbracket - \llbracket r \rrbracket$ 得到明文 $z - r$ ，可以将 $z - r$ 和 r 视为 z 的两个加法秘密份额。因此数据所有者和模型所有者可以基于混淆电路交互完成 ReLU 函数的比较操作。线性计算与非线性计算交替进行，最终完成神经网络推理过程。由于在线性计算中选择安全多方计算需要每次乘法都进行交互，因此虽然在本章方法中激活函数的计算仍然需要交互，相比于完全基于安全多方计算的方法，节省了大量的通信开销。

5.4.3 证明生成

模型所有者在执行加密推理之外，还需要生成加密推理的零知识证明。利用 5.4.1 节中生成的 QRP 关系，记为 $Q = (R_q, l, \{u_k(X), v_k(X), w_k(X)\}_{k=0}^m, t(X))$ 。假设电路 C 具有 m 个线和 n 个乘法门，使 $I_s = 1, 2, \dots, l$ 对应满足该 QRP 关系的声

明 $x = (a_1, \dots, a_l)$, 包括推理数据 $\llbracket d \rrbracket$, 模型承诺, 以及加密的输出结果 $\llbracket y \rrbracket$; $I_w = l+1, \dots, m$ 对应满足该 QRP 关系的见证 $w = (a_{l+1}, \dots, a_m)$, 包括模型参数 \mathbf{W} 和电路的中间值。因此, 对于加密推理计算 $\Pi(f)$, 其证明需要满足 $Q(x, w) = 1$, 即满足 5.4.1 中的条件。在证明生成和验证中, 模型所有者持有声明 $x = (a_1, \dots, a_l)$ 与见证 $w = (a_{l+1}, \dots, a_m)$, 其负责生成对应 QRP 关系 Q 的证明 π , 数据所有者仅持有声明 $x = (a_1, \dots, a_l)$, 负责对生成的证明 π 进行验证。

5.4.4 验证与解密

数据所有者收到加密的推理结果 $\llbracket y \rrbracket$ 及证明 π 后, 首先执行 $Verify(Q, vk, x, \pi) \rightarrow \{0,1\}$, 其中声明 $x = (a_1, \dots, a_l)$, 包括推理数据 $\llbracket d \rrbracket$, 模型承诺, 以及加密的输出结果 $\llbracket y \rrbracket$ 。若验证成功, 则返回 1。随后解密 $\llbracket y \rrbracket$ 得到明文推理结果 y 。若验证不成功, 则意味着模型所有者没有遵守约定的计算协议, 或者在推理过程中使用了不正确的模型参数。

5.5 隐私保护和安全性分析

本方案的隐私保护主要从加密推理阶段和加密推理证明阶段进行分析。在加密推理阶段, 主要包括推理数据和推理结果的隐私保护, 其依赖于 BGV 同态加密以及加密推理证明协议的安全性。在多方证明生成阶段, 包括模型的隐私保护和验证的正确性, 其依赖于加密推理证明协议的安全性。由于 BGV 同态加密的安全性已在相关研究中得到了证明^[127], 因此本节主要对加密推理证明阶段的安全性进行分析, 从加密推理证明生成协议 $\Gamma : (\text{Setup}, \text{Prove}, \text{Verify})$ 在完备性 (Completeness)、知识合理性 (Knowledge Soundness) 和零知识性 (Zero-Knowledge) 三个方面进行描述, 以证明本章方案的隐私保护效果。

定理 1: 如果模型所有者为 $(x, w) \in Q$ 生成了一个证明 π , 该证明能够以概率 Pr 说服一个诚实的验证者, 则协议 Γ 是完备的, 且满足:

$$\Pr \left[\begin{array}{l} (CRS, vk) \leftarrow \text{Setup}(1^\lambda, Q) \\ \pi \leftarrow \text{Prove}(Q, CRS, x, w) : \\ 1 \leftarrow \text{Verify}(Q, vk, x, \pi) \end{array} \right] \geq 1 - negl(\lambda) \quad (5-7)$$

证明: 对于 $(x, w) \in Q$ 的证明 $\pi = (A, B, C)$, 只有满足等式 $AB = E(\alpha)E(\beta) + \gamma E(f_s) + \delta C$, 其中 $f_s = (\beta u_s(\varepsilon) + \alpha v_s(\varepsilon) + w_s(\varepsilon)) / \gamma$, 该证明才会被接受。设 $u(\varepsilon) = \sum_{k=0}^m a_k u_k(\varepsilon) = u_s(\varepsilon) + u_w(\varepsilon)$, $v(\varepsilon) = \sum_{k=0}^m a_k v_k(\varepsilon) = v_s(\varepsilon) + v_w(\varepsilon)$, $w(\varepsilon) = \sum_{k=0}^n a_k w_k(\varepsilon) = w_s(\varepsilon) + w_w(\varepsilon)$, 由于 Encode 算法 $E()$ 具有加法同态性, 上述等式左侧和右侧的计算分别如下:

左侧：

$$\begin{aligned}
 A \cdot B &= E(\alpha + u(\varepsilon)) \cdot E(\beta + v(\varepsilon)) \\
 &= (E(\alpha) + E(u(\varepsilon))) \cdot (E(\beta) + E(v(\varepsilon))) \\
 &= E(u(\varepsilon))E(v(\varepsilon)) + E(\alpha)E(\beta) + E(\beta)E(u(\varepsilon)) + E(\alpha)E(v(\varepsilon)) \\
 &= E(u(\varepsilon))E(v(\varepsilon)) + Z
 \end{aligned}$$

右侧：

$$\begin{aligned}
 &E(\alpha)E(\beta) + \gamma E(f_{io}) + \delta C \\
 &= E(\alpha)E(\beta) + \gamma E\left(\frac{\beta u_s(\varepsilon) + \alpha v_s(\varepsilon) + w_s(\varepsilon)}{\gamma}\right) + \delta C \\
 &= E(\alpha)E(\beta) + E(\beta u(\varepsilon) + \alpha v(\varepsilon) + w(\varepsilon)) + E(h(\varepsilon)t(\varepsilon)) \\
 &= E(w(\varepsilon)) + E(h(\varepsilon)t(\varepsilon)) + E(\alpha)E(\beta) + E(\beta)E(u(\varepsilon)) + E(\alpha)E(v(\varepsilon)) \\
 &= E(w(\varepsilon)) + E(h(\varepsilon)t(\varepsilon)) + Z
 \end{aligned}$$

因此，如果验证者在验证阶段对一个正确的证明 π 输出 0，则意味着 $E(u(\varepsilon))E(v(\varepsilon)) \neq E(w(\varepsilon)) + E(h(\varepsilon)t(\varepsilon))$ ，也就是 $u(\varepsilon)v(\varepsilon) \neq w(\varepsilon) + h(\varepsilon)t(\varepsilon)$ ，这与 5.4.1 节中构造的 QRP 关系 Q 条件不符。因此，协议 $\Gamma: (\text{Setup}, \text{Prove}, \text{Verify})$ 是完备的。

定理 2：如果对于计算受限的敌手 \mathcal{A} ，和一个计算受限且可以完全访问敌手 \mathcal{A} 状态的提取器 E ，满足：

$$Pr \left[\begin{array}{l} (CRS, vk) \leftarrow \text{Setup}(1^\lambda, Q) \\ (x, \pi', w) \leftarrow (\mathcal{A}|E)(Q, CRS): \\ (x, w) \notin Q \wedge \\ 1 \leftarrow \text{Verify}(Q, vk, x, \pi') \end{array} \right] \leq negl(\lambda) \quad (5-8)$$

则协议 Γ 是知识合理的。

证明：假设有一个模拟器可以生成一系列系数 $A_\alpha, A_\beta, A_\gamma, A_\delta, \{A_k\}_{k=0}^m$ 和两个多项式 $A(x), A_h(x)$ ，则证明 π 中的元素 A, B, C 对应的 A_u 可以表示为：

$$\begin{aligned}
 A_u &= A_\alpha \alpha + A_\beta \beta + A_\gamma \gamma + A(s) + \sum_{k=0}^l A_k \frac{\beta u_k(\varepsilon) + \alpha v_k(\varepsilon) + w_k(\varepsilon)}{\gamma} \\
 &\quad + \sum_{k=l+1}^m A_k \frac{\beta u_k(\varepsilon) + \alpha v_k(\varepsilon) + w_k(\varepsilon)}{\delta} + A_h(s) \frac{t(s)}{\delta}
 \end{aligned}$$

B_v, C_w 也可以以相同的方式构造，该构造方式使得 A_u, B_v, C_w 包含了验证阶段的验证等式中包含的所有项。

对于验证阶段的等式 $AB = E(\alpha)E(\beta) + \gamma F + \delta C$ ，可以将其视为一个多元

Laurent 多项式的一个等式。根据 Rinocchio^[123]中定义的环上的 Laurent 多项式 Schwartz-Zippel 引理，敌手 \mathcal{A} 可以忽略不计的优势构造出满足等式的 A_u, B_v, C_w ，且提取器 E 无法从 A_u, B_v, C_w 计算出一个见证 w' 使得 $(x, w') \in Q$ 。因此概率 Pr 可以忽略不计，协议 Γ 是知识合理的。

定理 3：如果一个计算受限的敌手 \mathcal{A} 无法从诚实的模型所有者生成的证明中得到任何关于见证的信息，则协议 Γ 是零知识的。正式的，对于所有 $(x, w) \in Q$ 和一个计算受限的敌手 \mathcal{A} ，满足：

$$\Pr \left[\begin{array}{l} (CRS, vk) \leftarrow \text{Setup}(1^\lambda, Q) \\ \pi \leftarrow \text{Prove}(Q, CRS, x, w) : \\ 1 \leftarrow \mathcal{A}(Q, vk, \pi) \end{array} \right] = \Pr \left[\begin{array}{l} (CRS, vk) \leftarrow \text{Setup}(1^\lambda, Q) \\ \pi' \leftarrow \text{Sim}(Q, vk, x) : \\ 1 \leftarrow \mathcal{A}(Q, vk, \pi') \end{array} \right] \quad (5-9)$$

证明：假设存在一个模拟器，可以利用陷门 $vk = (sk, CRS, \varepsilon, \gamma, \delta)$ 执行 $\pi \leftarrow \text{Sim}(Q, vk, x)$ ，以获得模拟证明 π' ，计算过程如下：

模拟器 \mathcal{B} 随机选择 $A_u, B_v \in R_q$ ，且有 $A = E(A_u)$, $B = E(B_v)$ ，随后计算：

$$C = E(C_w) = \frac{A \cdot B - E(\alpha)E(\beta) - E(\beta u_s(\varepsilon) + \alpha v_s(\varepsilon) + w_s(\varepsilon))}{\delta}$$

因此，模拟器 \mathcal{B} 可以利用陷门生成模拟证明 $\pi' = (A, B, C)$ ，使得验证等式 $AB = E(\alpha)E(\beta) + \gamma F + \delta C$ 满足，该证明同样可以被诚实的验证者所接受。由正确的见证生成的证明 π 与模拟证明 π' 对于验证者是不可区分的，而模拟证明 π' 中，不包含任何与见证相关的信息。因此，如果敌手可以从证明 π 中提取到见证的信息，则无法满足 π 与 π' 不可区分的特性。因此，计算受限的敌手 \mathcal{A} 无法从一个证明中获得任何关于见证份额的信息，协议 Γ 满足零知识性。

5.6 实验评估

5.6.1 实验设置

实验配置：本章实验是在一台配备了英特尔酷睿(TM)i7-10700k @ 3.8 GHz CPU、128GB 内存的 Ubuntu 20.04 操作系统电脑上进行的。本文实验采用 C++ 语言，零知识证明基于 libsnark 和 ringSNARK 实现，同态加密基于 SEAL 库实现。同态加密通过 SIMD 操作并行处理多个数据。

数据集和模型：本文实验首先测试了环上基础运算的证明生成与验证开销，包括环上乘法、同态加密密文乘法，同态加密密文与密文乘法、比特分解等。随后在三个具有不同规模的分类任务的数据集和不同模型上测试加密推理的证明与验证时间。采用的数据集包括：WINE (Wine Data Set)、MNIST (Modified National Institute of Standards and Technology database)、CIFAR10

(Canadian Institute for Advanced Research, 10 classes)。相应的，在上述数据集上采用的模型包括：全连接神经网络模型（ShallowNet）、卷积神经网络模型（LeNet），测试具有不同数量级参数和计算量的模型推理性能。数据集和模型架构的详细信息见表 5-1 和表 5-2。

表 5-1 数据集详情
Table 5-1 Details of the datasets

数据集	类别	特征数	样本数
WINE	3	13	178
MNIST	10	28*28*1	70000
CIFAR10	10	32*32*3	60000

表 5-2 模型架构
Table 5-2 Model architectures

数据集	WINE	MNIST	CIFAR10
卷积层 1	/	/	卷积核 5×5 通道数 6
池化层	/	/	卷积核 2×2
卷积层 2	/	/	卷积核 5×5 通道数 16
池化层	/	/	卷积核 2×2
卷积层 3	/	/	卷积核 4×4 通道数 120
全连接层 1	13×32	784×128	480×84
全连接层 2	32×3	128×10	84×10

5.6.2 实验结果

5.6.2.1 环上基础运算开销

本节测试了环上基础运算的可信设置、证明生成、验证的计算开销，以及约束数量。可信设置由可信第三方执行，相同电路的多次运算仅需要一次可信设置。如表 5-3 所示，展示了环上乘法（明文乘法（Mul-PP）、同态加密密文与明文乘法（Mul-CP）、同态加密密文乘法（Mul-CC）、比特分解（Decom）等计算的开销。由于约束系统是由乘法电路转换而来，加法电路无需额外添加约束，因此本节未给出对加法运算的开销展示。比特分解为 BGV 同态加密执行密文乘法后，重线性化所需要的基础操作。实验结果表明，随着约束数量的增长，可信设置、证明生成和验证的计算开销也在逐渐增长。因此在神经网络的实际应用中，计算复杂度会随着输入规模和模型复杂度的增加而增加。降低计算开销的重点在于减少相关高开销基础操作数量。

表 5-3 基础操作计算开销
Table 5-3 Computation cost of basic operations

操作	Mul-PP	Mul-CP	Mul-CC	Decom
可信设置(s)	0.157	0.167	0.176	0.590
证明生成(s)	0.027	0.035	0.095	1.61
验证(s)	0.039	0.039	0.041	0.360
约束数量	1	2	4	34

5.6.2.2 神经网络证明与验证开销

本节测试了三个数据集在不同神经网络模型下，实现加密推理可验证所需的各阶段操作的开销，包括可信设置、证明生成及验证。对于相同网络结构的推理验证，可信设置仅需执行一次。证明生成过程由模型所有者完成，变量包括加密后的数据、权重、计算中间值及推理结果等，对于验证者，权重和中间值为秘密值，加密数据和推理结果为公开值。验证过程由验证者（数据所有者）完成。如表 5-4 所示，随着模型和数据集的复杂度增长，可信设置、证明生成、验证时间及约束数量均随之增长。其中，主要开销为证明生成过程。若想降低计算开销，主要可以从两方面入手，一方面优化约束构造方式，降低约束数量。另一方面提高每个约束的计算时间，从而提升整体效率。本文实验主要通过采用 SIMD 技术执行加密推理，从而大大降低了约束数量。

表 5-4 神经网络推理验证的计算开销
Table 5-4 Computation cost of neural network inference verification

模型	ShallowNet-WINE	ShallowNet-MNIST	LeNet-Cifar10
可信设置(s)	3.77	42.2	169
证明生成(s)	32.6	424	1631
验证(s)	10.9	142	526
约束数量	166	660	1370

5.6.2.3 约束数量及对比

本节对采用 SIMD 技术对加密推理过程进行优化后构造约束系统的约束数量，以及未采用 SIMD 技术，对直接进行加密推理计算的电路构造约束系统的约束数量进行了对比。如表 5-5 所示，采用 SIMD 技术后约束数量大大减少。主要原因便在于 SIMD 的并行计算特性。例如，在全连接层，假设输入数据的维度为 m ，神经元数量为 n ，即该数据向量需要与 n 个 m 维的权重向量进行点积运算，每个点积运算需要 $m*m$ 次乘法，因此每个全连接层在明文状态下进行计算大约会产生 $n*m*m$ 个约束。当点积运算中包含密文时，密文与明文或密文

乘法操作以及带来的额外比特分解等操作会导致更多的约束数量。在采用 SIMD 技术执行加密推理运算时，一个或多个 m 维数据（根据设定的参数，若数据维度较大，也可以编码一个数据中的部分元素）可以被编码并加密到同一个密文中，同样的权重向量也可以被编码到同一个明文中，这样多个乘法约束被压缩为 1 个乘法约束，从而大大降低约束数量。

表 5-5 约束数量
Table 5-5 Number of constraints

模型	ShallowNet-WINE	ShallowNet-MNIST	LeNet-Cifar10
SIMD	166	660	1370
无 SIMD	1129	203678	1759440

5.6.2.4 存储要求

本节对方案的内存占用、证明密钥、验证密钥及证明大小进行了测试。如表 5-6 所示，随着模型复杂度的增加，内存占用与证明密钥的大小也随之增加，而验证密钥与证明大小保持不变。本文方案的内存占用主要是在可信设置与证明生成过程，用户（即验证者）无需承担该过程的计算，因此仍然可以满足资源受限的用户的需求。证明密钥主要用于证明生成过程，该过程由模型所有者执行，因此该过程也不涉及用户。验证密钥与证明大小保持不变，且只需要较小的存储空间，因此即便发送给用户执行验证过程，也不需要对用户的资源做过高的要求。

表 5-6 内存占用及 CRS 和证明大小
Table 5-6 Memory usage and size of CRS and proof

模型	ShallowNet-WINE	ShallowNet-MNIST	LeNet-Cifar10
内存占用(GB)	1.47	5.54	29.35
CRS(MB)	371.11	1457.93	3260.46
验证密钥(MB)	0.48	0.48	0.48
证明(B)	2.04	2.04	2.04

5.6.2.5 方案对比

本节将 VHENN 与 pvCNN^[118]进行了对比。pvCNN 同样采用同态加密和 zk-SNARKs 实现神经网络推理中的隐私保护和可验证性。然而，该方案并未将同态加密和 zk-SNARKs 进行结合，而是将模型拆分为 PriorNet 和 LaterNet，其中 PriorNet 保持私有，LaterNet 被设定为非隐私部分，委托给服务器进行计算。在 PriorNet 部分采用同态加密保护数据隐私，在 LaterNet 部分采用 zk-SNARKs

实现对服务器运算结果的可验证性。由于 pvCNN 并非在整个神经网络模型上采用 zk-SNARKs，因此为了保持相同的对比基准，本节给出了输入维度为 $32*32*3$ ，卷积核为 5×5 ，通道数为 6 的一层卷积层中，执行前向传播时，构造证明系统所需的可信设置、证明生成、验证所需的时间，内存占用，CRS 大小以及证明大小等方面对比。如表 5-7 所示，得益于同态加密可以采用 SIMD 的特性，本方案的约束数量大大降低，因此在各方面的表现均优于 pvCNN。此外，pvCNN 所需的验证时间过长，对验证者的要求过高，并不适用于神经网络推理服务的场景。更重要的，本方案结合了同态加密与 zk-SNARKs，相较于 pvCNN 的隐私保护和可验证分别针对部分模型，本方案可以同时实现神经网络推理的隐私保护与可验证。

表 5-7 与 pvCNN 的对比实验
Table 5-7 Comparison with pvCNN

方案	pvCNN	VHENN
可信设置(s)	4650.41	0.35
证明生成(s)	3997.82	0.56
验证(s)	408911	0.15
内存占用	17.30 GB	0.63GB
CRS 大小	14.29 GB	44.18 MB
证明大小	4.76 GB	0.48 MB

5.7 本章小结

本章提出了一种可验证的神经网络加密推理方案-VHENN。首先结合环上 zk-SNARKs 协议与基于环多项式的同态加密方案-BGV，实现了可验证同态加密方案。随后将可验证同态加密方案与基于 BGV 的神经网络加密推理方案相结合，实现了满足模型、推理数据、推理结果隐私保护以及模型真实性和推理正确性可验证的神经网络推理方案。最后，本文进行了实验评估，以展示 VHENN 的性能，并与相关方案进行了对比分析。

本方案首次将可验证同态加密方案应用于神经网络推理中，实现可验证的神经网络加密推理方案。然而，受限于同态加密的效率，当前对神经网络加密推理的研究与应用发展较慢，结合零知识证明会进一步降低整体方案的效率，且需要较大的内存开销，因此仍然需要进一步的研究与优化。未来研究重点在于如何通过提高同态加密的计算效率以及优化同态加密与零知识证明结合的方法等，使得可验证神经网络加密推理的效率进一步提高，使其能够在实际应用中部署。

第 6 章 总结与展望

6.1 工作总结

本文主要基于安全多方计算、同态加密、零知识证明等密码学技术，针对神经网络合作训练与推理中的隐私保护和可验证性进行了研究。针对神经网络训练中的隐私保护问题，提出基于多密钥同态加密的隐私保护合作训练方案，实现了训练数据、模型及标签的隐私保护。针对神经网络推理过程中的隐私保护及验证问题，提出了两种满足不同推理场景下的推理方案：可验证神经网络隐私保护多方推理方案和可验证同态加密神经网络推理方案，实现了推理过程中的模型、推理数据和推理结果的隐私保护，以及对模型真实性和推理结果正确性的可验证性。本文工作成果总结如下：

1) 首先，本文提出了一种具有隐私保护的合作学习方案：SecureSL。SecureSL 通过将拆分学习和多密钥同态加密相结合，使得持有不同特征数据的各参与方能够合作训练模型，并确保参与者和模型所有者的隐私，抵御合谋攻击。为了提高加密计算的效率，提出了两种优化算法，并以兼容 SIMD 的方式优化了模型训练的计算过程。实验表明 SecureSL 可以有效抵御数据重构攻击和标签推理攻击，并且相比于已有的基于噪声扰动的方法，能提供更好的隐私保护效率。在准确率方面，与基于噪声扰动的拆分学习相比，SecureSL 可以在提供相似的隐私保护效果的前提下，达到 65%-70% 的模型准确率，而基于噪声扰动的拆分学习仅达到 50% 的模型准确率。

2) 其次，本文提出了一种具有隐私保护和可验证的神经网络推理方案：VSecNN。首先将 MPC、Groth16 协议以及 KZG 多项式承诺进行结合，实现了多方证明生成方法，使得 Groth16 的证明生成过程可以以多方合作的形式完成。为了提高多方证明生成的效率，在证明生成过程，将传统基于有限域的 MPC 协议替换为基于椭圆曲线群的 MPC 协议。随后将多方证明生成方法融入到安全多方推理中，以实现具有隐私保护及可验证的神经网络多方安全推理方案。实验结果表明，相比于单方无隐私保护的可验证神经网络推理方案，VSecNN 在全连接模型推理的证明生成的时间上相较于基准方案降低了 2-9 倍，验证时间与对比方案接近。

3) 最后，本文提出了一种可验证的同态加密神经网络推理方案-VHENN。首先结合环上 zk-SNARKs 协议-RGroth16 与基于环多项式的同态加密方案-BGV，实现了可验证同态加密方案。随后将可验证同态加密方案与基于 BGV 的神经网络加密推理方案相结合，实现了满足模型、推理数据、推理结果隐私

保护以及模型真实性和推理正确性可验证的神经网络推理方案。VHENN 可以在无需多方交互的前提下，满足神经网络安全推理的可验证需求。实验结果表明，相比于对比方案，本方案在可信设置、证明生成和验证等环节的计算时间缩短了超过 4 个数量级。

6.2 未来工作展望

本文工作主要聚焦于安全多方计算、同态加密、零知识证明等密码学方案与神经网络的结合。虽然本文方案均致力于提升效率以及神经网络与上述密码学方案之间的兼容性，但目前仍然存在一定的局限性且主要集中于简单的全连接和卷积神经网络模型。主要原因在于：1) 即使进行了多种优化，基于上述密码学方案的计算相较于底层计算效率仍然较低，而神经网络训练与推理过程又包含大量基础运算，因此整体方案效率仍然未达到实际应用的需求。2) 本文采用的多为基于算术电路的密码学方案，无法很好的支持神经网络中的非线性运算，只能对神经网络中的激活函数采取多项式近似、修改激活函数等方法以适应密码学方案的计算。因此会带来使用范围的局限性以及准确率的牺牲等问题。因此，在当前工作的基础上，未来研究主要从以下方面着手：

1) 计算与通信效率的提升：安全多方计算的主要开销在于多方之间的通信，降低通信开销的主要方式是通过降低交轮次。因此未来工作拟研究基于函数秘密共享的安全多方计算协议，实现具有常数轮交互的神经网络安全训练和推理方案。同态加密的主要开销在于同态运算，当前方案主要采用 SIMD 的方式提高效率，并构造 SIMD 友好的神经网络计算形式。然而采用 SIMD 虽然可以通过并行运算降低加解密和同态乘法等运算的数量，但其引入旋转操作开销巨大，当并行运算的数据较少时，采用 SIMD 并不一定高效。未来工作拟研究如何通过将快速数论变换（Number Theoretic Transforms, NTT）中的剩余数系统模替换为费马数的方式，从根本上提高同态加密的计算效率，而无需进一步的改变神经网络的计算方式。

2) 算术电路与布尔电路的转换：目前基于安全多方计算的神经网络安全计算中，已广泛采用混合多方计算协议，即在神经网络的线性计算部分采用基于算术电路的 MPC 协议，在神经网络的非线性计算部分则转换为基于布尔电路的 MPC 协议。在同态加密和零知识证明中，同样存在类似的解决方式（如 CKKS 同态加密和 TFHE 同态加密间的转换）。但由于转换效率问题，尚未有研究将其有效的与神经网络计算相结合。为了提高方案适用性，减少对神经网络计算方式的修改，未来工作拟研究算术电路与布尔电路的同态加密和零知识证明转换技术，以实现其在神经网络计算中的高效应用。

3) 更广泛的适用范围: 为了适应当前神经网络的发展, 已有部分研究将基于安全多方计算的神经网络安全推理方案扩展至大语言模型领域, 但当前研究仍然处于初期阶段。对于同态加密和零知识证明在除全连接和卷积神经网络模型之外更复杂的模型中的应用研究几乎空白。因此, 未来工作拟从同态加密和零知识证明入手, 针对不同复杂模型的特点, 研究更为普适的密码学方案与神经网络结合方式。

参考文献

- [1] 刘俊旭, 孟小峰. 机器学习的隐私保护研究综述[J]. 计算机研究与发展, 2020, 57(2): 1-17
- [2] 谭作文, 张连福. 机器学习隐私保护研究综述[J]. 软件学报, 2020, 31(7): 1-30
- [3] Konečný J., McMahan H. B., Yu F. X., et al. Federated learning: Strategies for improving communication efficiency[A]. Proceedings of the NIPS Workshop on Private Multi-Party Machine Learning (2016)[C]. San Diego, California, 2016:1-10
- [4] Fu F., Wang X., Jiang J., et al. ProjPert: Projection-Based Perturbation for Label Protection in Split Learning Based Vertical Federated Learning [J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(7): 3417-3428
- [5] Ng L. K. L., Chow S. S. M. SoK: Cryptographic Neural-Network Computation[A]. Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP)[C]. San Francisco, CA, USA, 2023:497-514
- [6] 魏立斐, 陈聪聪, 张蕾, 等. 机器学习的安全问题及隐私保护[J]. 计算机研究与发展, 2020, 57(10): 2066-2085
- [7] 韩伟力, 宋鲁杉, 阮雯强, 等. 安全多方学习:从安全计算到安全学习[J]. 计算机学报, 2023, 46(7): 1494-1512
- [8] 胡奥婷, 胡爱群, 胡韵, 等. 机器学习中差分隐私的数据共享及发布:技术, 应用和挑战[J]. 信息安全学报, 2022, 7(4): 1-16
- [9] 马俊明, 吴秉哲, 余超凡, 等. S3ML:一种安全的机器学习推理服务系统[J]. 软件学报, 2022, 33(9): 1-19
- [10] Garg S., Goel A., Jain A., et al. zkSaaS: Zero-Knowledge SNARKs as a Service[A]. Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23)[C]. Anaheim, CA, USA, 2023:4427-4444
- [11] Na H., Oh Y., Lee W., et al. Systematic Evaluation of Robustness Against Model Inversion Attacks on Split Learning[A]. Proceedings of the Information Security Applications[C]. Singapore, 2024:107-118
- [12] Zhang L., Gao X., Li Y., et al. Functionality and Data Stealing by Pseudo-Client Attack and Target Defenses in Split Learning [J]. IEEE Transactions on Dependable and Secure Computing, 2024: 1-16
- [13] Gao X., Zhang L. PCAT: Functionality and data stealing from split learning by Pseudo-Client attack[A]. Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23)[C]. Anaheim, CA, USA, 2023:5271-5288

-
- [14] 任奎, 孟泉润, 闫守琨, 等. 人工智能模型数据泄露的攻击与防御研究综述[J]. 网络与信息安全学报, 2021, 7(01): 1-10
 - [15] Shokri R., Stronati M., Song C., et al. Membership inference attacks against machine learning models[A]. Proceedings of the 2017 IEEE symposium on security and privacy (SP)[C]. San Jose, CA, USA, 2017:1-16
 - [16] 婷 高. 机器学习成员推理攻击研究进展与挑战[J]. 运筹与模糊学, 2022, 12(1): 1-15
 - [17] Ganju K., Wang Q., Yang W., et al. Property inference attacks on fully connected neural networks using permutation invariant representations[A]. Proceedings of the 2018 ACM SIGSAC conference on computer and communications security[C]. Toronto Canada, 2018:619-633
 - [18] Mao Y., Xin Z., Li Z., et al. Secure Split Learning Against Property Inference, Data Reconstruction, and Feature Space Hijacking Attacks[A]. Proceedings of the European Symposium on Research in Computer Security–ESORICS 2023[C]. The Hague, The Netherlands, 2023:23-43
 - [19] Yu F., Wang L., Zeng B., et al. SIA: A sustainable inference attack framework in split learning [J]. Neural Networks, 2024, 171: 396-409
 - [20] Huang H., Li X., He W. Pixel-Wise Reconstruction of Private Data in Split Federated Learning[A]. Proceedings of the Information and Communications Security[C]. Singapore, 2023:435-450
 - [21] Gong X., Chen Y., Yang W., et al. InverseNet: Augmenting Model Extraction Attacks with Training Data Inversion[A]. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)[C]. Montreal, 2021:2439-2447
 - [22] Wei W., Liu L., Loper M., et al. A framework for evaluating client privacy leakages in federated learning[A]. Proceedings of the European Symposium on Research in Computer Security-ESORICS 2020[C]. Guildford, UK, 2020:545-566
 - [23] Lam M., Wei G.-Y., Brooks D., et al. Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix[A]. Proceedings of the International Conference on Machine Learning[C]. Virtual, 2021:5959-5968
 - [24] Lyu L., Yu H., Ma X., et al. Privacy and Robustness in Federated Learning: Attacks and Defenses [J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(7): 8726-8746
 - [25] Liu J., Lyu X., Cui Q., et al. Similarity-Based Label Inference Attack Against Training and Inference of Split Learning [J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 2881-2895
 - [26] Wan X., Sun J., Wang S., et al. PSLF: Defending Against Label Leakage in

-
- Split Learning[A]. Proceedings of the 32nd ACM International Conference on Information and Knowledge Management[C]. Birmingham, United Kingdom, 2023:2492-2501
- [27] Fu C., Zhang X., Ji S., et al. Label inference attacks against vertical federated learning[A]. Proceedings of the 31st USENIX Security Symposium (USENIX Security 22)[C]. Boston, MA, USA, 2022:1397-1414
- [28] Liu Y., Zou T., Kang Y., et al. Batch label inference and replacement attacks in black-boxed vertical federated learning [J]. arXiv preprint arXiv:211205409, 2021: 1-14
- [29] Li O., Sun J., Yang X., et al. Label Leakage and Protection in Two-party Split Learning[A]. Proceedings of the Tenth International Conference on Learning Representations (ICLR 2022)[C]. Virtual, 2021:1-27
- [30] Pasquini D., Ateniese G., Bernaschi M. Unleashing the tiger: Inference attacks on split learning[A]. Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security[C]. Virtual Event, Republic of Korea, 2021:2113-2129
- [31] Jin X., Chen P.-Y., Hsu C.-Y., et al. CAFE: Catastrophic data leakage in vertical federated learning[A]. Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021)[C]. Virtual, 2021:994-1006
- [32] Shokri R., Shmatikov V. Privacy-Preserving Deep Learning[A]. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security[C]. Denver, Colorado, USA, 2015:1310–1321
- [33] Vaidya J., Kantarcıoglu M., Clifton C. Privacy-preserving naive bayes classification [J]. The VLDB Journal, 2008, 17(4): 879-898
- [34] Gascón A., Schoppmann P., Balle B., et al. Secure Linear Regression on Vertically Partitioned Datasets [J]. IACR Cryptol ePrint Arch, 2016, 2016: 345-364
- [35] 宋蕾, 马春光, 段广晗, 等. 基于数据纵向分布的隐私保护逻辑回归[J]. 计算机研究与发展, 2019, 56(10): 2243-2249
- [36] Rubinstein B., Bartlett P., Huang L., et al. Learning in a large function space: Privacy-preserving mechanisms for SVM learning [J]. Journal of Privacy and Confidentiality, 2012, 4(1): 65-100
- [37] Chaudhuri K., Monteleoni C. Privacy-preserving logistic regression[A]. Proceedings of the Advances in Neural Information Processing Systems 21 (NIPS 2008)[C]. Vancouver British Columbia Canada, 2008:1-8
- [38] Bonawitz K., Ivanov V., Kreuter B., et al. Practical Secure Aggregation for Privacy-Preserving Machine Learning[A]. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security[C]. Dallas,

-
- Texas, USA, 2017:1175–1191
- [39] So J., Güler B., Avestimehr A. S. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning [J]. IEEE Journal on Selected Areas in Information Theory, 2021, 2(1): 479-489
- [40] Wang Z., Yang G., Dai H., et al. Privacy-Preserving Split Learning for Large-Scaled Vision Pre-Training [J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 1539-1553
- [41] Gu C., Cui X., Zhu X., et al. FL2DP: Privacy-Preserving Federated Learning Via Differential Privacy for Artificial IoT [J]. IEEE Transactions on Industrial Informatics, 2024, 20(4): 5100-5111
- [42] Tang X., Shen M., Li Q., et al. PILE: Robust Privacy-Preserving Federated Learning Via Verifiable Perturbations [J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(6): 5005-5023
- [43] Mugunthan V., Polychroniadou A., Byrd D., et al. Smpai: Secure multi-party computation for federated learning[A]. Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS 2019)[C]. Vancouver, Canada, 2019:1-9
- [44] Zhang C., Ekanut S., Zhen L., et al. Augmented Multi-Party Computation Against Gradient Leakage in Federated Learning [J]. IEEE Transactions on Big Data, 2022: 1-10
- [45] Liu S., Luo J., Zhang Y., et al. Efficient privacy-preserving Gaussian process via secure multi-party computation [J]. Journal of Systems Architecture, 2024, 151: 103-134
- [46] Yang Z., Chen Y., Huangfu H., et al. Dynamic Corrected Split Federated Learning With Homomorphic Encryption for U-Shaped Medical Image Networks [J]. IEEE Journal of Biomedical and Health Informatics, 2023, 27(12): 5946-5957
- [47] Khan T., Nguyen K., Michalas A., et al. Love or Hate? Share or Split? Privacy-Preserving Training Using Split Learning and Homomorphic Encryption[A]. Proceedings of the 2023 20th Annual International Conference on Privacy, Security and Trust (PST)[C]. Copenhagen, Denmark, 2023:1-7
- [48] Zhang C., Li S., Xia J., et al. BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning[A]. Proceedings of the 2020 USENIX annual technical conference (USENIX ATC 20)[C]. Virtual, 2020:493-506
- [49] Sav S., Pyrgelis A., Troncoso-Pastoriza J. R., et al. POSEIDON: Privacy-preserving federated neural network learning[A]. Proceedings of the 28Th Annual Network And Distributed System Security Symposium (Ndss 2021)[C]. ELECTR NETWORK, 2021:1-18
- [50] Liu Z., Guo J., Yang W., et al. Privacy-Preserving Aggregation in Federated

-
- Learning: A Survey [J]. IEEE Transactions on Big Data, 2022: 1-20
- [51] Phong L. T., Aono Y., Hayashi T., et al. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption [J]. IEEE Transactions on Information Forensics and Security, 2018, 13(5): 1333-1345
- [52] Lou Q., Feng B., Charles Fox G., et al. Glyph: Fast and accurately training deep neural networks on encrypted data[A]. Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020)[C]. Virtual, 2020:9193-9202
- [53] Li T., Li J., Chen X., et al. NPMML: A Framework for Non-Interactive Privacy-Preserving Multi-Party Machine Learning [J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(6): 2969-2982
- [54] Mohassel P., Zhang Y. Secureml: A system for scalable privacy-preserving machine learning[A]. Proceedings of the 2017 IEEE symposium on security and privacy (SP)[C]. San Jose, CA, USA, 2017:19-38
- [55] Corrigan-Gibbs H., Boneh D. Prio: Private, robust, and scalable computation of aggregate statistics[A]. Proceedings of the 14th USENIX symposium on networked systems design and implementation (NSDI 17)[C]. 2017:259-282
- [56] Mohassel P., Rindal P. ABY3: A mixed protocol framework for machine learning[A]. Proceedings of the 2018 ACM SIGSAC conference on computer and communications security[C]. Toronto Canada, 2018:35-52
- [57] Chen Z., Zhang F., Zhou A. C., et al. ParSecureML: An efficient parallel secure machine learning framework on GPUs[A]. Proceedings of the 49th International Conference on Parallel Processing-ICPP[C]. Edmonton AB Canada, 2020:1-11
- [58] Patra A., Suresh A. BLAZE: blazing fast privacy-preserving machine learning[A]. Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2020[C]. San Diego, CA, USA, 2020:1-18
- [59] Chaudhari H., Rachuri R., Suresh A. Trident: Efficient 4pc framework for privacy preserving machine learning[A]. Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2020[C]. San Diego, CA, USA, 2019:1-18
- [60] Attrapadung N., Hamada K., Ikarashi D., et al. Adam in private: Secure and fast training of deep neural networks with adaptive moment estimation[A]. Proceedings of the 22th Privacy Enhancing Technologies Symposium[C]. Sydney, Australia, 2022:746-767
- [61] Wagh S., Tople S., Benhamouda F., et al. Falcon: Honest-Majority Maliciously Secure Framework for Private Deep Learning[A]. Proceedings of the Proceedings on Privacy Enhancing Technologies[C]. On the Internet, 2021:188-208

-
- [62] Zheng W., Deng R., Chen W., et al. Cerebro: A Platform for Multi-Party Cryptographic Collaborative Learning[A]. Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)[C]. Virtual Event, 2021:2723-2740
 - [63] Chang I., Sotiraki K., Chen W., et al. HOLMES: Efficient Distribution Testing for Secure Collaborative Learning[A]. Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23)[C]. Anaheim, CA, USA, 2023:4823-4840
 - [64] Rathee D., Bhattacharya A., Gupta D., et al. Secure Floating-Point Training[A]. Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23)[C]. Anaheim, CA, USA, 2023:6329-6346
 - [65] Agrawal N., Shahin Shamsabadi A., Kusner M. J., et al. QUOTIENT: two-party secure neural network training and prediction[A]. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security[C]. London United Kingdom, 2019:1231-1247
 - [66] Liu J., Juuti M., Lu Y., et al. Oblivious neural network predictions via minionn transformations[A]. Proceedings of the 2017 ACM SIGSAC conference on computer and communications security[C]. Dallas Texas USA, 2017:619-631
 - [67] Rouhani B. D., Riazi M. S., Koushanfar F. Deepsecure: Scalable provably-secure deep learning[A]. Proceedings of the 55th annual design automation conference[C]. San Francisco California, 2018:1-6
 - [68] Riazi M. S., Weinert C., Tkachenko O., et al. Chameleon: A hybrid secure computation framework for machine learning applications[A]. Proceedings of the 2018 on Asia conference on computer and communications security[C]. Incheon Republic of Korea, 2018:707-721
 - [69] Demmler D., Schneider T., Zohner M. ABY-A framework for efficient mixed-protocol secure two-party computation[A]. Proceedings of the Network and Distributed System Security (NDSS) Symposium 2015[C]. San Diego, California, 2015:1-15
 - [70] Rathee D., Rathee M., Kumar N., et al. CrypTFlow2: Practical 2-party secure inference[A]. Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security[C]. Virtual Event 2020:325-342
 - [71] Ryffel T., Tholoniat P., Pointcheval D., et al. Ariann: Low-interaction privacy-preserving deep learning via function secret sharing[A]. Proceedings of the 22th Privacy Enhancing Technologies Symposium[C]. Sydney, Australia, 2022:291–316
 - [72] Mishra P., Lehmkuhl R., Srinivasan A., et al. Delphi: A cryptographic inference service for neural networks[A]. Proceedings of the 29th USENIX Security Symposium (USENIX Security 20)[C]. Virtual, 2020:2505-2522

-
- [73] Jha N. K., Ghodsi Z., Garg S., et al. DeepReDuce: Relu reduction for fast private inference[A]. Proceedings of the International Conference on Machine Learning[C]. Virtual, 2021:4839-4849
 - [74] Knott B., Venkataraman S., Hannun A., et al. Crypten: Secure multi-party computation meets machine learning[A]. Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021)[C]. Virtual, 2021:4961-4973
 - [75] Ng L. K., Chow S. S. GForce: GPU-Friendly Oblivious and Rapid Neural Network Inference[A]. Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)[C]. Virtual Event, 2021:2147-2164
 - [76] Lehmkuhl R., Mishra P., Srinivasan A., et al. Muse: Secure inference resilient to malicious clients[A]. Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)[C]. Virtual Event, 2021:2201-2218
 - [77] Chandran N., Gupta D., Obbattu S. L. B., et al. SIMC: ML Inference Secure Against Malicious Clients at Semi-Honest Cost[A]. Proceedings of the 31st USENIX Security Symposium (USENIX Security 22)[C]. Boston, MA, USA, 2022:1361-1378
 - [78] Rathee D., Bhattacharya A., Sharma R., et al. SECFLOAT: Accurate Floating-Point meets Secure 2-Party Computation (Full Version)[A]. Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP)[C]. San Francisco, CA, USA, 2022:576-595
 - [79] Huang Z., Lu W.-j., Hong C., et al. Cheetah: Lean and Fast Secure Two-Party Deep Neural Network Inference[A]. Proceedings of the 31st USENIX Security Symposium[C]. Boston, MA, USA, 2022:809-826
 - [80] Dalskov A., Escudero D., Keller M. Secure evaluation of quantized neural networks[A]. Proceedings of the 20th Privacy Enhancing Technologies Symposium[C]. On the Internet, 2020:355-375
 - [81] Hou X., Liu J., Li J., et al. Ciphergpt: Secure two-party gpt inference [J]. Cryptology ePrint Archive, 2023: 1-16
 - [82] Dong Y., Lu W.-j., Zheng Y., et al. Puma: Secure inference of llama-7b in five minutes [J]. arXiv preprint arXiv:230712533, 2023: 1-13
 - [83] 刘伟欣, 管晔玮, 霍嘉荣, 等. 一种基于安全多方计算的快速 Transformer 安全推理方案[J]. 计算机研究与发展, 2024, 61(5): 1218-1229
 - [84] Li D., Shao R., Wang H., et al. MPCFormer: fast, performant and private transformer inference with mpc[A]. Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023)[C]. Kigali Rwanda, 2023:1-16
 - [85] Wagh S., Gupta D., Chandran N. SecureNN: 3-Party Secure Computation for Neural Network Training[A]. Proceedings of the 25th Privacy Enhancing

-
- Technologies Symposium[C]. Stockholm, Sweden, 2019:26-49
- [86] Shen L., Chen X., Shi J., et al. An efficient 3-party framework for privacy-preserving neural network inference[A]. Proceedings of the European Symposium on Research in Computer Security-ESORICS 2020 [C]. Guildford, UK, 2020:419-439
- [87] Kumar N., Rathee M., Chandran N., et al. Cryptflow: Secure tensorflow inference[A]. Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP)[C]. San Francisco, CA, USA, 2020:336-353
- [88] Tan S., Knott B., Tian Y., et al. CryptGPU: Fast privacy-preserving machine learning on the GPU[A]. Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP)[C]. San Francisco, CA, USA, 2021:1021-1038
- [89] Byali M., Chaudhari H., Patra A., et al. FLASH: fast and robust framework for privacy-preserving machine learning[A]. Proceedings of the 20th Privacy Enhancing Technologies Symposium[C]. On the Internet, 2020:459-480
- [90] Koti N., Patra A., Rachuri R., et al. Tetrad: actively secure 4pc for secure training and inference[A]. Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2022[C]. San Diego, CA, USA, 2021:1-18
- [91] Dalskov A., Escudero D., Keller M. Fantastic Four: Honest-Majority Four-Party Secure Computation With Malicious Security[A]. Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)[C]. Virtual Event, 2021:2183-2200
- [92] Koti N., Pancholi M., Patra A., et al. SWIFT: Super-fast and Robust Privacy-Preserving Machine Learning[A]. Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)[C]. Virtual Event, 2021:2651-2668
- [93] 阎允雪, 马铭, 蒋瀚. 基于秘密分享的高效隐私保护四方机器学习方案[J]. 计算机研究与发展, 2022, (10): 2338-2347
- [94] Bost R., Popa R. A., Tu S., et al. Machine learning classification over encrypted data [J]. Cryptology ePrint Archive, 2014: 1-34
- [95] Gilad-Bachrach R., Dowlin N., Laine K., et al. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy[A]. Proceedings of the International conference on machine learning[C]. New York, USA, 2016:201-210
- [96] Hesamifard E., Takabi H., Ghasemi M. CryptoDL: Deep Neural Networks over Encrypted Data [J]. arXiv preprint arXiv:171105189, 2017: 1-21
- [97] Juvekar C., Vaikuntanathan V., Chandrakasan A. GAZELLE: A low latency framework for secure neural network inference[A]. Proceedings of the 27th USENIX Security Symposium (USENIX Security 18)[C]. Baltimore, MD,

USA, 2018:1651-1669

- [98] Dathathri R., Saarikivi O., Chen H., et al. CHET: an optimizing compiler for fully-homomorphic neural-network inferencing[A]. Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation[C]. Phoenix AZ USA, 2019:142-156
- [99] Xu G., Li H., Ren H., et al. Secure and verifiable inference in deep neural networks[A]. Proceedings of the Annual Computer Security Applications Conference[C]. Austin USA, 2020:784-797
- [100] Boemer F., Cammarota R., Demmler D., et al. MP2ML: A mixed-protocol machine learning framework for private inference[A]. Proceedings of the 15th International Conference on Availability, Reliability and Security[C]. Virtual Event Ireland, 2020:1-10
- [101] Zhang Q., Xin C., Wu H. GALA: Greedy Computation for linear algebra in privacy-preserved neural networks[A]. Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2021[C]. Virtual, 2021:1-16
- [102] Chen H., Dai W., Kim M., et al. Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference[A]. Proceedings of the ACM SIGSAC Conference on Computer and Communications Security[C]. London United Kingdom, 2019:395-412
- [103] Lu W.-j., Huang Z., Hong C., et al. PEGASUS: bridging polynomial and non-polynomial evaluations in homomorphic encryption[A]. Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP)[C]. San Francisco, CA, USA, 2021:1057-1073
- [104] Jovanovic N., Fischer M., Steffen S., et al. Private and reliable neural network inference[A]. Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security[C]. Los Angeles CA USA, 2022:1663-1677
- [105] Cong K., Das D., Park J., et al. SortingHat: Efficient Private Decision Tree Evaluation via Homomorphic Encryption and Transciphering[A]. Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security[C]. Los Angeles CA USA, 2022:1-27
- [106] Kim D., Guyot C. Optimized privacy-preserving cnn inference with fully homomorphic encryption [J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 2175-2187
- [107] Aharoni E., Adir A., Baruch M., et al. HeLayers: A Tile Tensors Framework for Large Neural Networks on Encrypted Data[A]. Proceedings of the 23th Privacy Enhancing Technologies Symposium[C]. Lausanne, Switzerland and Online, 2023:325–342

- [108] Sanyal A., Kusner M., Gascon A., et al. TAPAS: Tricks to accelerate (encrypted) prediction as a service[A]. Proceedings of the 35th International Conference on Machine Learning[C]. Stockholm, Sweden, 2018:4490-4499
- [109] Folkerts L., Gouert C., Tsoutsos N. G. REDsec: Running encrypted discretized neural networks in seconds[A]. Proceedings of the Network and Distributed System Security (NDSS) Symposium 2023[C]. San Diego, CA, USA, 2023:1-17
- [110] Xing Z., Zhang Z., Liu J., et al. Zero-knowledge proof meets machine learning in verifiability: A survey [J]. arXiv preprint arXiv:231014848, 2023: 1-23
- [111] Ghodsi Z., Gu T., Garg S. Safetynets: Verifiable execution of deep neural networks on an untrusted cloud[A]. Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017)[C]. Long Beach, CA, USA, 2017:1-10
- [112] Liu T., Xie X., Zhang Y. ZkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy[A]. Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security[C]. Virtual Event, Republic of Korea, 2021:2968-2985
- [113] Weng C., Yang K., Xie X., et al. Mystique: Efficient Conversions for Zero-Knowledge Proofs with Applications to Machine Learning[A]. Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)[C]. Virtual Event, 2021:501-518
- [114] Hao M., Chen H., Li H., et al. Scalable Zero-knowledge Proofs for Non-linear Functions in Machine Learning[A]. Proceedings of the 33rd USENIX Security Symposium[C]. Philadelphia, PA, USA, 2024:3819-3836
- [115] Lee S., Ko H., Kim J., et al. vCNN: Verifiable convolutional neural network based on zk-snarks [J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(4): 4254 - 4270
- [116] Feng B., Qin L., Zhang Z., et al. ZEN: An optimizing compiler for verifiable, zero-knowledge neural network inferences [J]. Cryptology ePrint Archive, 2021: 1-25
- [117] Zhao L., Wang Q., Wang C., et al. VeriML: Enabling integrity assurances and fair payments for machine learning as a service [J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(10): 2524-2540
- [118] Weng J., Weng J., Tang G., et al. pvCNN: Privacy-preserving and verifiable convolutional neural network testing [J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 2218-2233
- [119] Chen B.-J., Waiwitlikhit S., Stoica I., et al. ZKML: An Optimizing System for ML Inference in Zero-Knowledge Proofs[A]. Proceedings of the Nineteenth

- European Conference on Computer Systems[C]. Athens Greece, 2024:560-574
- [120] Niu C., Wu F., Tang S., et al. Toward verifiable and privacy preserving machine learning prediction [J]. IEEE Transactions on Dependable and Secure Computing, 2020, 19(3): 1703-1721
- [121] Li X., He J., Vijayakumar P., et al. A verifiable privacy-preserving machine learning prediction scheme for edge-enhanced HCPSs [J]. IEEE Transactions on Industrial Informatics, 2021, 18(8): 5494-5503
- [122] Dong C., Weng J., Liu J.-N., et al. Fusion: Efficient and secure inference resilient to malicious servers[A]. Proceedings of the Network and Distributed System Security (NDSS) Symposium 2023[C]. San Diego, CA, USA, 2023:1-18
- [123] Ganesh C., Nitulescu A., Soria-Vazquez E. Rinocchio: SNARKs for ring arithmetic [J]. Journal of Cryptology, 2023, 36(4): 1-50
- [124] Paillier P. Public-key cryptosystems based on composite degree residuosity classes[A]. Proceedings of the International conference on the theory and applications of cryptographic techniques[C]. Prague, Czech Republic, 1999:223-238
- [125] Shamir A. How to share a secret [J]. Communications of the ACM, 1979, 22(11): 612-613
- [126] Acar A., Aksu H., Uluagac A. S., et al. A survey on homomorphic encryption schemes: Theory and implementation [J]. ACM Computing Surveys (Csur), 2018, 51(4): 1-35
- [127] Brakerski Z., Gentry C., Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping [J]. ACM Transactions on Computation Theory (TOCT), 2014, 6(3): 1-36
- [128] Cheon J. H., Kim A., Kim M., et al. Homomorphic encryption for arithmetic of approximate numbers[A]. Proceedings of the Advances in Cryptology—ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security[C]. Hong Kong, China, 2017:409-437
- [129] Lindell Y. Secure multiparty computation [J]. Communications of the ACM, 2020, 64(1): 86-96
- [130] Yao A. C. Protocols for secure computations[A]. Proceedings of the 23rd annual symposium on foundations of computer science (sfcs 1982)[C]. Chicago, IL, USA, 1982:160-164
- [131] Huang Y., Evans D., Katz J., et al. Faster secure Two-Party computation using garbled circuits[A]. Proceedings of the 20th USENIX Security Symposium (USENIX Security 11)[C]. SAN FRANCISCO, CA, 2011:35-51
- [132] Yadav V. K., Andola N., Verma S., et al. A survey of oblivious transfer

-
- protocol [J]. ACM Computing Surveys (Csur), 2022, 54(10s): 1-37
- [133] Keller M. MP-SPDZ: A versatile framework for multi-party computation[A]. Proceedings of the 2020 ACM SIGSAC conference on computer and communications security[C]. Virtual Event, 2020:1575-1590
- [134] Li F., McMillin B. A survey on zero-knowledge proofs [M]. Advances in computers. Elsevier. 2014: 25-69
- [135] Kate A., Zaverucha G. M., Goldberg I. Constant-Size Commitments to Polynomials and Their Applications[A]. Proceedings of the Advances in Cryptology - ASIACRYPT 2010[C]. Singapore, 2010:177-194
- [136] Nielsen M. A. Neural networks and deep learning [M]. Determination press San Francisco, CA, USA, 2015: 1-211
- [137] Chang Y., Zhang K., Gong J., et al. Privacy-Preserving Federated Learning via Functional Encryption, Revisited [J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 1855-1869
- [138] Halevi S., Shoup V. Algorithms in HElib[A]. Proceedings of the Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference[C]. Santa Barbara, CA, USA, 2014:554-571
- [139] Jang J., Lee Y., Kim A., et al. Privacy-preserving deep sequential model with matrix homomorphic encryption[A]. Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security[C]. Nagasaki Japan, 2022:377-391
- [140] Balla S., Koushanfar F. HELiKs: HE Linear Algebra Kernels for Secure Inference[A]. Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security[C]. Copenhagen Denmark, 2023:2306-2320
- [141] Kim T., Kwak H., Lee D., et al. Asymptotically Faster Multi-Key Homomorphic Encryption from Homomorphic Gadget Decomposition[A]. Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security[C]. Copenhagen Denmark:726–740
- [142] Brutzkus A., Gilad-Bachrach R., Elisha O. Low latency privacy preserving inference[A]. Proceedings of the International Conference on Machine Learning[C]. Long Beach, California, USA, 2019:812-821
- [143] Jawalkar N., Gupta K., Basu A., et al. Orca: FSS-based Secure Training and Inference with GPUs[A]. Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP)[C]. San Francisco, CA, USA, 2024:597-616
- [144] Liu J., Li X., Liu X., et al. Privacy-Preserving and Verifiable Outsourcing Linear Inference Computing Framework [J]. IEEE Transactions on Services Computing, 2023, 16(6): 4591-4604
- [145] Fan Y., Xu B., Zhang L., et al. psvCNN: A Zero-Knowledge CNN

-
- Prediction Integrity Verification Strategy [J]. IEEE Transactions on Cloud Computing, 2024, 12(2): 359-369
- [146] Kanjalkar S., Zhang Y., Gndlur S., et al. Publicly Auditable MPC-as-a-Service with succinct verification and universal setup[A]. Proceedings of the 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)[C]. Vienna, Austria, 2021:386-411
- [147] Rivinius M., Reisert P., Rausch D., et al. Publicly Accountable Robust Multi-Party Computation[A]. Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP)[C]. San Francisco, CA, USA, 2022:2430-2449
- [148] Ozdemir A., Boneh D. Experimenting with collaborative zk-SNARKs: Zero-Knowledge proofs for distributed secrets[A]. Proceedings of the 31st USENIX Security Symposium (USENIX Security 22)[C]. Boston, MA, USA, 2022:4291-4308
- [149] Boneh D., Boyle E., Corrigan-Gibbs H., et al. Zero-knowledge proofs on secret-shared data via fully linear PCPs[A]. Proceedings of the Annual International Cryptology Conference[C]. Santa Barbara, CA, USA, 2019:67-97
- [150] Cui H., Zhang K., Chen Y., et al. MPC-in-Multi-Heads: A Multi-Prover Zero-Knowledge Proof System[A]. Proceedings of the European Symposium on Research in Computer Security-ESORICS 2021 [C]. Darmstadt, Germany, 2021:332-351
- [151] Groth J. On the size of pairing-based non-interactive arguments[A]. Proceedings of the Advances in Cryptology-EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques[C]. Vienna, Austria, 2016:305-326
- [152] Damgård I., Pastro V., Smart N., et al. Multiparty computation from somewhat homomorphic encryption[A]. Proceedings of the Annual Cryptology Conference[C]. Berlin, Heidelberg, 2012:643-662
- [153] Jacob B., Kligys S., Chen B., et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[A]. Proceedings of the IEEE conference on computer vision and pattern recognition[C]. SALT LAKE CITY, 2018:2704-2713
- [154] Zhang Z., Plantard T., Susilo W. Reaction attack on outsourced computing with fully homomorphic encryption schemes[A]. Proceedings of the Information Security and Cryptology-ICISC 2011: 14th International Conference[C]. Seoul, Korea, 2012:419-436
- [155] Chaturvedi B., Chakraborty A., Chatterjee A., et al. A practical full key recovery attack on tfhe and fhew by inducing decryption errors [J]. Cryptology ePrint Archive, 2022: 1-18
- [156] Chenal M., Tang Q. On key recovery attacks against existing somewhat

-
- homomorphic encryption schemes[A]. Proceedings of the Third International Conference on Cryptology and Information Security[C]. Latin America Florianópolis, Brazil, 2015:239-258
- [157] Gennaro R., Wichs D. Fully homomorphic message authenticators[A]. Proceedings of the 19th International Conference on the Theory and Application of Cryptology and Information (ASIACRYPT 2013)[C]. Bengaluru, India, 2013:301-320
- [158] Fiore D., Gennaro R., Pastro V. Efficiently verifiable computation on encrypted data[A]. Proceedings of the 2014 ACM SIGSAC conference on computer and communications security[C]. Scottsdale Arizona USA 2014:844-855
- [159] Chatel S., Knabenhans C., Pyrgelis A., et al. Verifiable encodings for secure homomorphic analytics [J]. arXiv preprint arXiv:220714071, 2022: 1-26
- [160] Natarajan D., Loveless A., Dai W., et al. Chex-mix: Combining homomorphic encryption with trusted execution environments for two-party oblivious inference in the cloud [J]. Cryptology ePrint Archive, 2021: 1-19
- [161] Gong B., Lau W. F., Au M. H., et al. Efficient Zero-Knowledge Arguments For Paillier Cryptosystem[A]. Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP)[C]. San Francisco, CA, USA, 2024:96-96
- [162] Fiore D., Nitulescu A., Pointcheval D. Boosting verifiable computation on encrypted data[A]. Proceedings of the Public-Key Cryptography–PKC 2020: 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, [C]. Edinburgh, UK, 2020:124-154
- [163] Bois A., Cascudo I., Fiore D., et al. Flexible and efficient verifiable computation on encrypted data[A]. Proceedings of the 24th IACR International Conference on Practice and Theory of Public Key Cryptography[C]. Label Leakage and Protection in Two-party Split Learning, 2021:528-558

攻读博士学位期间发表的论文及其它成果

(一) 发表的学术论文

已发表：

- [1] Yang W., Wang N., Guan Z., Wu L., Du X. and Guizani M. A Practical Cross-Device Federated Learning Framework over 5G Networks[J]. IEEE Wireless Communications, 2022, 29(6): 128-134. (第一作者, 收录号: 000917339700018, 中科院1区Top期刊, 影响因子: 10.9)
- [2] Yang W., Wang X., Guan Z., Wu L., Du X. and Guizani M. SecureSL: A Privacy-Preserving Vertical Cooperative Learning Scheme for Web 3.0[J]. IEEE Transactions on Network Science and Engineering, 2024, 11(5): 3983-3994. (第一作者, 收录号: 001294586400075, 中科院小类1区, 影响因子: 6.7)
- [3] Yang W., Guan Z., Wu L. and He Z. A Secure Neural Network Inference Framework for Intelligent Connected Vehicles[J]. IEEE Network, 2024, 38(6): 120-127. (第一作者, 收录号: 001360457500012, 中科院小类1区, 影响因子: 6.8)
- [4] Wang N., Yang W., Wang X., Wu L., Guan Z., Du X. and Guizani M. A blockchain based privacy-preserving federated learning scheme for Internet of Vehicles[J]. Digital Communications and Networks, 2024, 10(1): 126-134. (第二作者, 收录号: 001223673300001, 中科院2区, ESI高被引, 影响因子: 7.5)
- [5] Wang R., Wang X., Yang W., Yuan S., Guan Z. Achieving fine-grained and flexible access control on blockchain-based data sharing for the Internet of Things[J]. China Communications, 2022, 19(6): 22-34. (第三作者, 收录号: 000818877000007, 中科院3区, 影响因子: 3.1)
- [6] Wang N., Yang W., Wu L., Guan Z., Du X. and Guizani M. BPFL: A Blockchain Based Privacy-Preserving Federated Learning Scheme[A]. Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM)[C]. Madrid, Spain, 2021:1-6. (第二作者, 收录号: 20221311872328, CCF C类会议)

在审：

- [1] 杨文梯, 何朝阳, 李萌, 张子剑, 关志涛, 祝烈煌. VHENN: 基于环上零知识证明协议的可验证同态加密神经网络推理方案[J]. 计算机学报. (第一作者, 大修, 一级学报, CCF A类中文科技期刊)

-
- [2] He Z., Yang W., Wu L., Guan Z. ScureBadger: A Homomorphic Encryption-based Framework for Secure Medical Inference[J]. Digital Communications and Networks. (第二作者, 小修, 中科院 2 区)

(二) 申请及已获得的专利

- [1] 关志涛, 王霄东, 杨文梯. 一种基于区块链的隐私保护机器学习训练与推理方法及系统: 中国, 202111207606.9 [P]. 2021-10-18. (第三作者, 已授权)

攻读博士学位期间参加的科研工作

- [1] 面向神经网络推理的隐私保护技术研究 (No. 62372173), 国家自然科学基金面上项目, 2024.1-2027.12.
- [2] 面向能源互联网电力交易的数据安全访问控制方法研究 (No. 61972148), 国家自然科学基金面上项目, 2020.1-2023.12.
- [3] 基于联邦学习的跨区域数据安全流通和数据共享机制研究 (No. 2021110045003279), 中国电力科学研究院有限公司实验室开放基金, 2021.6-2022.12.
- [4] 面向电力数据可信共享的隐私计算协作保护关键技术与应用 (No. 521207230002), 国网安徽电力公司科技项目, 2023.1-2023.12.

致 谢

这已经是我在华电待的第 11 个年头了，占了我 1/3 还多的人生。

本科四年，我读着不知道是做什么、高考完胡乱报的专业，迷茫的度过了四年的光阴。没有见识、没有兴趣、没有目标、也没有理想，连心血来潮想努力的时候都不知道要做什么。宿舍所有人都在准备考研的时候，我在实习准备工作。有一天实习回来的晚上，突然觉得人活着应该要有目标的，决定开始考研只用了一个晚上。

感谢自己，在没有人支持和理解，甚至家里人觉得我肯定考不上的情况下，顶住了压力，做出了正确的选择并为之努力。

硕博六年半，在机缘巧合下选择了关志涛教授当我的导师，又在自己的努力下得到了他的青睐成为了他的学生。这几年学习到了很多东西，去了很多地方，见识了更宽广的世界，也开始有了自己感兴趣的事情。实现了很多大大小小的目标，熬过的夜，起过的早都开始有了意义。我开始体会到靠自己努力拼搏然后获得成功的快乐，当然也体会了很多努力拼搏之后却什么也没得到的挫败。回过头看，选择是大于努力的，在正确的选择下，付出的努力与收获也是成正比的。

感谢关志涛教授，在我毕业那年评上博导，让我有机会再次做出了正确的选择。关老师曾提到，有位师姐在致谢中对他的评价是温润如玉，我一点也不能理解。不过，虽然他对我不温润也不如玉，甚至脾气还有些大，我的感恩依然是发自内心的。感谢他对我的指引，带我见识到的这些更宽广的世界，带我做出的选择，提供的各种相对优越的条件，以及感谢他没在我跟他吵架的时候把我逐出实验室。

博士三年半，压力很大，又因为亲人的离世时常陷入到很差的情绪中。真的很幸运在 BAIS 遇到了很多很好的人们，王乃玉，张千一，王霄东，李轩，何朝阳（好像那次机场差点没赶上飞机的点名）等等，感谢你们的友好和容忍，感谢那些一起快乐一起疯癫一起发狂的日子，这个团体是我遇到过的最快乐的团体，支撑我走过了读博生涯最痛苦的时光，没有你们我可能早都抑郁了。另外，特别鸣谢李轩，何朝阳在我的论文上做出的帮助。

感谢吕泽芳，性格跟我最同频的好朋友，倾听了我太多的负面情绪，当我的科研搭子。

感谢陈冲，性格跟我完全相反的好朋友，听我每天碎碎念，满足我的分享欲，当我的聊天搭子。

致 谢

最后，再次感谢自己，在度过了黑暗的时光之后，依然努力找到了能把我照亮的这些可爱的人们。

马上就要成为 Dr. Yang 了，我的姥姥，路焕永，一定会为我感到骄傲的，如果她能知道的话。当然，即便没有成为 Dr. Yang，我也一定是她的骄傲。

作者简介

1992年7月28日出于河北省南宫市。

2013年9月考入华北电力大学可再生能源学院新能源科学与工程专业，
2017年6月本科毕业并获得工学学士学位。

2018年9月——2021年6月在华北电力大学控制与计算机工程学院计算机
科学与技术学科学习并获得工学硕士学位。

2021年9月——2024年12月在大学华北电力大学控制与计算机工程学院
控制科学与工程学科学习并获得工学博士学位。

获奖情况：国家奖学金、优秀博士生、优秀研究生（2023-2024, 2022-2023）。