CipherMind: 世界上最长的密码本

Ming Nie*, Zhixiong Yang*
Fudan University
ShangHai,China

1 引言

信息的加密传输在互联网时代已经非常常见，而大型语言模型的广泛运用让我们开始思考利用大模型进行加密在密码学领域应用的可能性。随着大模型的参数的数量级逐年增高、模型功能逐渐复杂、参数语义缺乏研究等诸多原因，大模型中间结果可解释性问题越来越难以得到解决。基于这个前提，我们认为，对于同一个大模型的两次对不同上下文的推理，我们每次随机抽取任意一层的中间结果作为密文，这两组密文都是不可区分的。因此，我们可以尝试以微调后模型的中间层输出作为传输信息，实现信息的加密传输。

我们提出了 CipherMind，一种基于大模型黑箱过程的信息加密手段。为实现这一目标，我们使用了 Qwen2.5-0.5B-Instruct 进行“异地孪生”的微调方法。即使用相同的数据集和随机种子进行微调，保证微调后得到的模型相同，且推理得到的内容相同。

核心贡献:

- 实现确定性微调方法，保证不同机器上的相同模型在微调之后得到完全相同的参数，实现“异地孪生”
- 提出 CipherMind 加密通讯模型，通过中间层输出接收的方式对信息进行加解密，并结合 CBC 加密过程保证安全性

2 背景和动机

2.1 背景

加密，是指将要传输的信息 (明文) 转变为另一种不可读的信息形式 (密文) 的方法。加密虽然无法防止信息被窃取和截获，但是可以使内容对攻击者不可见。理想情况下，只有被授权方可以将密文转换为明文 (解密)。在网络中的加密信息传输场景下，加密方式一般通过伪随机数、加密算法等来完成，因此理论上攻击方可以通过暴力破解密钥的方式来解密，并不存在理想的加密方式。实际的加密算法只需要满足计算安全即可。

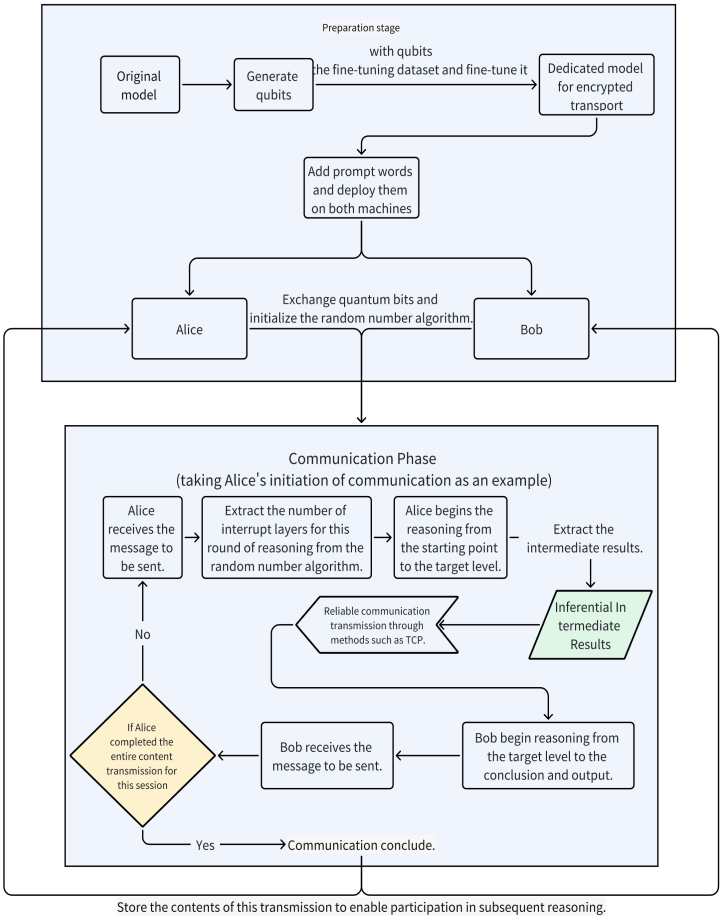

图 1: CipherMind 设计思路

2.2 动机

随着人工智能技术的兴起，算力得到了大幅的提升，这也使得过去凭借高计算量保证安全的算法变得容易受到攻击。结合大语言模型训练和使用都需要大量算力的特点，我们思考是否能够利用这一特性构造一种加密方式，其计算复杂度能够匹配上当前的高算力技术，使得攻击者即使消耗大量资源也无法破解。

大语言模型最大的特点之一即其参数与中间输出的不可解释性，我们认为这种不可解释的中间输出可以作为一种传输信息的密文，结合大语言模型的推理过程将要传递的信息按照一定格式以 prompt 输入，“随机地”截取中间层作为加密结果，传递给接收方进行解析，完成加密过程。这种方法目前缺少有效的方式对它们进行解读、模仿乃至修改，因此可能有广泛的应用前景。

因此，我们的 CipherMind 有以下动机:

- 在进行调研和学习的过程中，我们发现，数据加密领域正在出现许多和 AI 技术深度结合的案例。但是尚未出现和 AI 推理中间过程结合进行加密的类似工作。

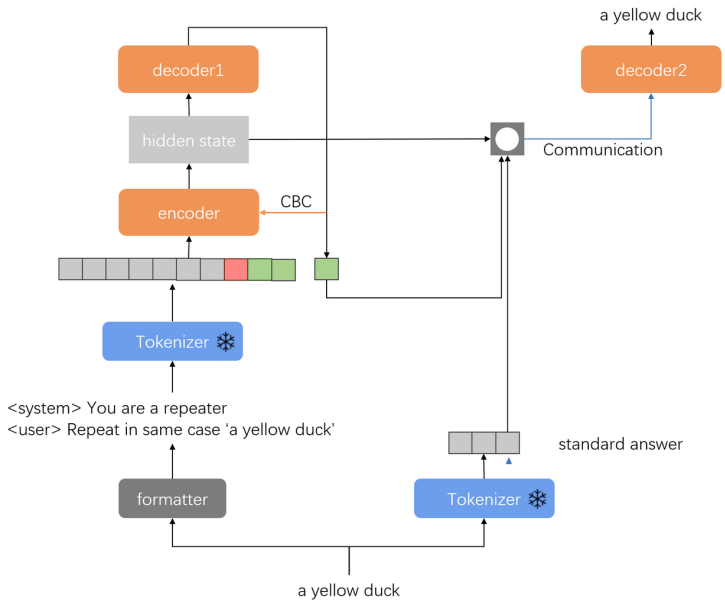

图 2: 针对输入的文本，先将其添加到预定义的模板中，要求模型复读指定内容，之后逐 token 地进行生成。每生成一个 token，便于希望传输的内容进行比较，如果匹配则将中间层输出进行传输，直到下一个 token 为 <eos>。同时，每次输出地 token 还会以一种 CBC 加密的方式更新下一次中间层的选择

- 随着算力不断膨胀，需要新的高算力加密方式；传统的算法有着严谨、自洽等优势，但依然有可解释性强的弱点，传统算法有过时的可能。
- 大模型推理中间结果具有可解释性弱，随机性强，不易模仿、篡改等优势，因此有着与加密算法深度结合的可能性。

3 设计

本节中我们将介绍 CipherMind 的设计，随后我们将介绍几个关键的技术部分，包括 (1) 微调设计 (2) 异地孪生 (3)CBC 加密。在介绍这些关键部分的同时，我们会提出相关的威胁模型，以此明确这些技术的必要性。

3.1 CipherMind 总览

CipherMind 的结构框架和具体运行方式如下图所示。

CipherMind Workflow CipherMind 的设计理念可以总结为以下三点:

1. 微调设计

   CipherMind 在设计上会使用小参数模型进行确定性微调，使得攻击者即使知道加密使用的是什么模型，将微调后模型的中间层输出放入原模型中也很难得到有意义的结果 [5.3]。具体可以构造一个 128 个小数据集构成的集合，并在通讯开始前用随机数挑选子集；最终攻击者必须通过暴力穷举 $2^{1}28$

2. 异地孪生

   为了避免传输微调后参数遭到截获的问题，这里采用异地同步微调的方法。通过保证使用的数据集相同，随机种子相同，使得通讯双方最终微调得到的参数完全一致。除此之外，在推理过程中通过随机种子设置保证中间过程的完全一致，确保发送方的中间结果能够在接收方得到与预期中相同的结果，实现“异地孪生”。

3. CBC 加密

   通讯过程中，除了模型中间层输出本来的不可解释性之外，这里还引入中间层数的“随机性”。准确地说，这里是使用基于 token 的 CBC 密码的方式对中间层的选择进行加密。对于一个句子的生成，每输出一个 token，下一次输出的层数便于根据之前输出的所有 token 改变，且“异地孪生”保证接收方和发送方的调整一致，进一步保证加密算法的安全性。

4 实验

本节中，我们将根据我们提出的理论和架构，介绍我们的模拟实验及结果。

4.1 实验设置

本节所用的所有基底模型均为 Qwen2.5-0.5B-Instruct 模型，均使用 lora 微调方法进行微调。在进行微调时，我们发现微调过程中的微调步数对模型最终的性能有较大的影响，因此我们使用不同的微调步数，对基底模型进行了不同程度的微调并实验，以展示 CipherMind 设计的有效性和稳定性。微调用数据集均为定制化修改过的 squad_v2 数据集 (为了实现我们的" 重复" 任务，按照 10:1 的比例向其中注入了基于我们任务的提示词-输出对)。具体的代码细节可以在我们的 github 仓库中找到。

4.2 正确传输能力实验

为了探究对模型而言的最佳微调参数，并测试模型的传输能力，进行以下实验。实验当中全部使用随机字符串进行测试，若最终输出包含目标字符串且成功输出 <eos>, 就视为传输成功。

实验结果显示，当微调参数设置为 20 以内时，所有的模型与原模型相比都几乎没有损失，甚至对长度较长的内容具有更强的传输能力。但随着传输长度的增加，正确性传输变得越来越困难。我们认为，这可能是由于:

- 模型本身参数较小，造成了模型能力的限制;
- 数据集质量一般，微调手段也有一定的优化空间;

4.3 碰撞实验

本实验意在模拟在具有 CPA 能力的攻击者攻击下，该架构的稳定性。为了不让语义信息对模型做出影响,这里依然使用和正确传输同一种实现的随机字符串进行传输。sender 和 attacker 均使用 CipherMind 实现，但是 attacker 获得的信息每次都有所不同；且 sender 使用微调之后的模型，而攻击者使用原始模型。

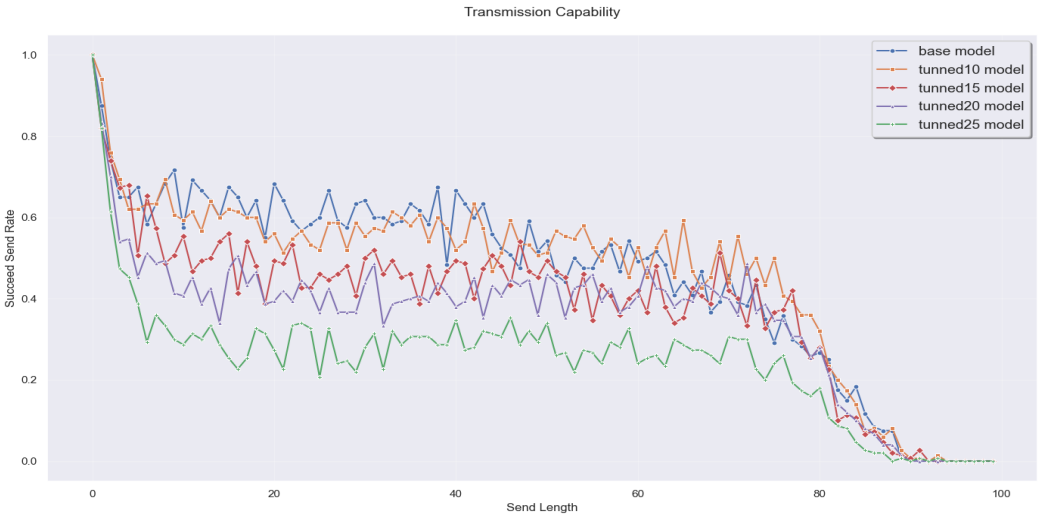

图 3: 不同微调程度模型的信息传输能力

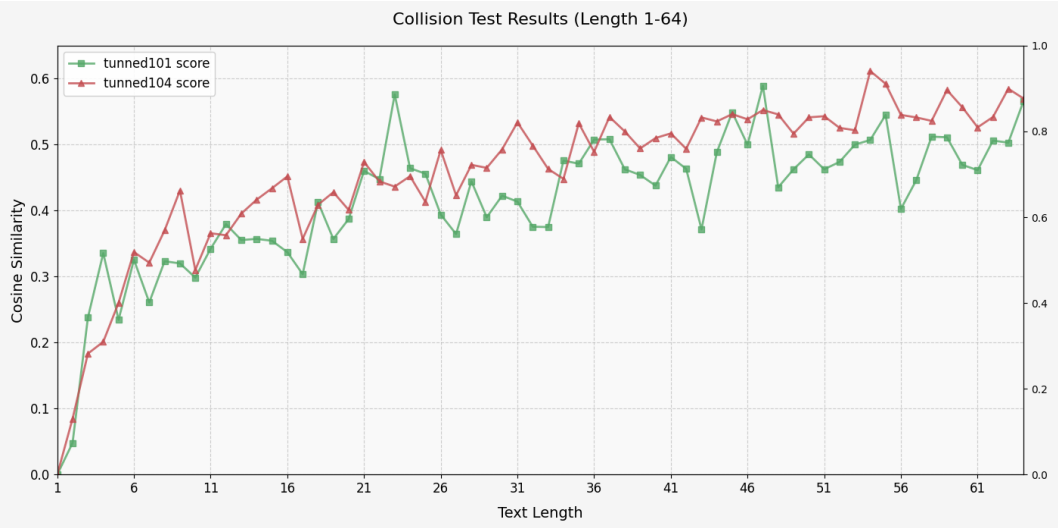

图 4: 消融随机层数之后的碰撞概率对比图

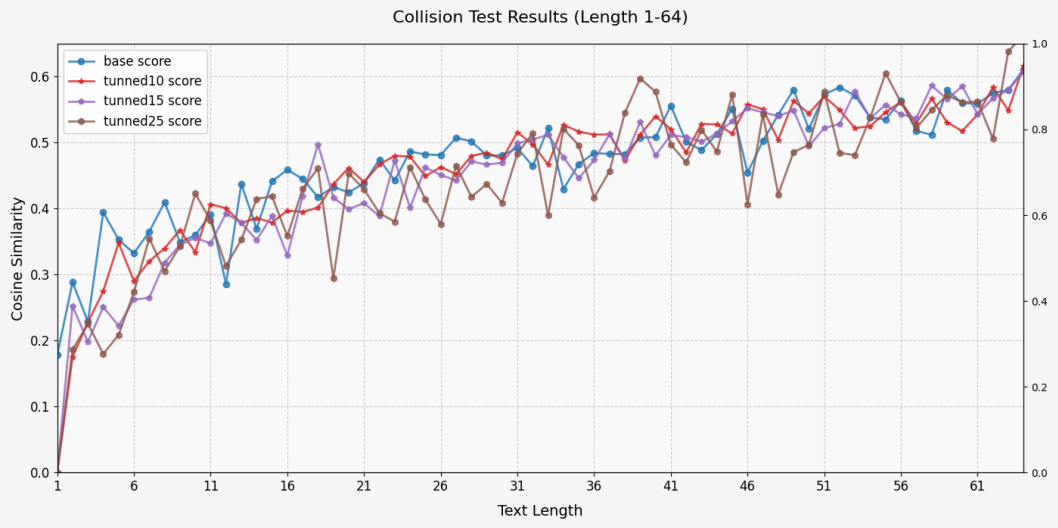

图 5: 消融微调之后的碰撞概率对比图

以正确性实验的结果作为参考，在进行碰撞实验时，我们只选择长度为 64 字符及以下的字符串进行实验。首先，我们选择消融随机层数的影响，attacker 接下来无法得知 sender 的伪随机数生成器与当前种子; 上图中，可以看到，如果对此进行消融，那么碰撞率会上升 20% 左右。我们猜测可能由于:

- 微调带来了模型中间参数的变化，使得本身消除了一部分碰撞概率;
- Qwen0.5B 模型每次推理会经过 27 个中间层，随机层数消除的碰撞概率应该会和模型层数大小有关。

其次，我们对比了不同强度微调对碰撞率的影响。在上图中，base score 表示使用原模型作为 sender，tunned10 和 tunned20 表示不同程度的微调模型。和原模型相比，微调程度越大的模型进行碰撞实验时，余弦相似度越低；但碰撞率下降并不显著，我们猜测可能由于:

- 模型参数量小，且模型可承受的微调造成的影响也较低。
- 由于注意力机制，模型在收到含疑问或祈使语气的内容会增大失败率。
- 本实验为了试图消除语义层面的影响，使用了随机字符串进行传输，最终进行余弦相似度计算时也是逐字符计算的；加入语义后，微调对于结果的影响应当更加显著。

整体而言，实际使用场景下，模型参数的大小选择和微调参数的选取之间值得根据使用场景做出权衡，以此发挥出 CipherMind 的更大潜能。

5 开放性探索

5.1 实际应用构想

根据 CipherMind 的 P2P 加密特点，可以选择在网关内部实现一个加密网络: 所有的成员同意协议之后，统一部署微调前大模型；每当有新的线路要建立，都会通过 RSA 等方法交换随机数，在双方本地生成一组异地孪生大模型。不同网关之间也可以使用类似机制联络。

5.2 理想加密强度

在 CipherMind 的加密过程当中，有两个主要的加密手段: (1) 部署前微调加密 (2) 通讯时动态加密

1. 部署前微调加密

   通讯首次建立时，双方会通过 RSA 交换一系列密钥，并用于在 128 个微调用数据集中进行取舍，从而建立起大小为 2^128

2. 通讯时动态加密

为了在微调的基础上再加一次保险，CipherMind 考虑了通讯建立之后，利用通讯历史向量化存储的方法，以检索增强生成 (RAG) 等方式进行动态加密。在这个前提下，窃听者在破解密钥成功的基础上，还需要获取所有的通讯历史才能攻击成功，也就让 CipherMind 形成了前向保密性。

基于上述两种方法，我们认为，这种加密方式可以实现计算安全要求。

5.3 潜在的问题与挑战

1. 部署成本问题当前市面上最小的开源模型所占内存依然以 GB 为单位；尽管 lora 微调方法支持单独存储参数，细化到用户之间的部署依然对内存和算力是不小的负担。这些部署成本对 CipherMind 的广泛部署提出了挑战。

2. 微调数据集的选择问题当前缺乏对数据集之间语义相似性进行定量计算的研究，这种相似性对微调后结果的干扰程度也缺乏定量的数据支持；根据我们的测试，当模型出现灾难性遗忘时，性能会大幅下降，因此挑选优质数据集以缓解这一现象也是一个重要问题。如何构造通用的计算方法已选择最优的数据集集合可以作为后续的研究重点。

6 总结

在本文中，我们提出了 CipherMind，一种基于大模型推理的加密通讯框架，并且根据其中的理念进行了初步的实践。经过我们的探索和实验，这种框架有着很高加密潜力和应用前景，其实现并不困难，成本易于接受，且理想的加密强度很令人激动，尤其是在分布式的计算和加密场景下或许会有非常强大的性能。

参考文献