中国科学技术大学

硕士学位论文

生成式可证安全文本隐写及其应用

作者姓名： 丁锦扬

学科专业： 网络空间安全

导师姓名： 张卫明 教授 陈可江 特任副研究员

完成时间： 二〇二四年六月五日

University of Science and Technology of China

A dissertation for master's degree

Provably Secure Generative Linguistic Steganography and Its Applications

Author: Jinyang Ding

Speciality: Cyberspace Security

Supervisor: Prof. Weiming Zhang and Assoc. Res. Kejiang Chen

Finished time: June 5, 2024

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：_____ 签字日期：_____

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

控阅的学位论文在解密后也遵守此规定。

☑ 公开 ☐ 控阅（____ 年）

作者签名：_____ 导师签名：_____

签字日期：_____ 签字日期：_____

摘　要

随着互联网的迅猛发展与广泛普及，信息安全问题日益突出。传统的基于密码学的解决方案虽能保障内容保密，但可能引起攻击者的注意。因此，现代通信安全既要求内容保密又希望过程隐蔽，信息隐藏技术应运而生并得到广泛应用。隐写术作为信息隐藏的重要分支，能将秘密消息隐秘地嵌入到各类信息载体中，从而实现隐蔽通信或存储，对军事通信和个人隐私保护至关重要。传统的修改式隐写只能停留在经验安全，实现可证安全一直是隐写领域的一大追求。实现可证安全隐写的一个前提是存在一个能够严格按照载体分布进行采样的采样器。深度生成模型的快速发展和生成数据的流行为可证安全隐写带来全新的技术手段和伪装环境，促进了生成式可证安全隐写的研究。

考虑到计算能力的快速进步以及未来可能出现的新型攻击手段，现有的传统密码学很可能在几十年后就不再安全，因此需要更加强大的加密措施应对长期安全需求。蜜罐加密是一种新型密码学技术，其目标是：若尝试使用错误密钥来解密密文，会得到看似合理的诱饵明文，这样，即使攻击者通过暴力破解获得了所有可能的明文，也难以从中定位出唯一的真明文。这样的设计显著提升了加密的安全性，使得蜜罐加密成为抵御暴力破解攻击的有效手段，特别适用于那些需要长期保护和对抗强大计算能力攻击的场景。

本文从数据隐私保护的两大核心需求——隐蔽性和保密性着手，围绕生成式可证安全隐写和蜜罐加密展开研究。本文的主要工作和创新点总结如下。

1. 提出了一种基于分布副本的高效可证安全隐写构造

现有的可证安全隐写构造可分为三种：基于拒绝采样的隐写构造嵌入率低、速度慢；基于算术编码的隐写构造和基于样本空间分组的隐写构造虽然能达到较高的嵌入率和较快的速度，但是具体实现过程难以满足理论假设，因此无法达到预期的安全性。上述方法本质上都是在使用样本的索引值或样本的函数值来表达消息。为了突破这些方法的局限，本文提出了基于分布副本的高效可证安全隐写构造。通过循环移位的方式为生成模型给出的概率分布创建多个分布副本，并利用分布副本的索引值来表达消息。从分布保持的角度，给出了这种隐写构造的安全性证明。通过递归分组的思想，显著提升了这种隐写构造的嵌入率。以文本生成任务为例，对这种隐写构造进行了性能测试。实验结果表明，该隐写构造的嵌入率能接近理论极限。与正常生成相比，该隐写构造引入的额外时间相对较小，这确保了它在实际应用中的高效性。此外，将该隐写构造应用于语音合成任务上，验证了该隐写构造的通用性。

2. 提出了一种基于深度生成模型和算术编码的文本蜜罐加密方案

蜜罐加密的核心是分布转换编码器（DTE），现有的 DTE 通常采用传统的统计模型和基于累积分布函数的定长编码方案。然而，这种方案在处理复杂数据时显示出建模能力、泛化能力差，而且往往不能较好地保持原始分布。本文对生成式隐写与蜜罐加密之间的关系进行分析，并尝试将生成式可证安全隐写领域的先进思想应用于文本蜜罐加密系统的设计中，提出了一种基于深度生成模型和算术编码的 DTE：首先对明文进行分词，然后使用算术编码根据深度生成模型预测的概率分布将明文编码为种子。为了消除可区分特征，在明文和种子两个层面上设计了伪随机填充方案。实验结果表明，所提出的方案压缩率较高、编码损失较小，在使用规模较大的模型时，建模损失较小、安全性较高。

关键词：信息隐藏；隐写；可证安全隐写；蜜罐加密；生成模型；算术编码

ABSTRACT

Information security issues have become increasingly prominent as the internet undergoes rapid development and widespread popularity. While traditional cryptographic solutions secure content confidentiality, they may still attract attackers' attention. Thus, modern communication security requires not only the confidentiality of information but also the concealment of the process, leading to the emergence and widespread application of information hiding techniques. Steganography, a significant branch of information hiding, enables the covert embedding of secret messages within various carriers, which is crucial for military communications and personal privacy protection. Traditional modification-based steganographic algorithms can only achieve empirical security. Achieving provable security has always been an important pursuit for the steganographic community. One prerequisite for achieving provably secure steganography is the existence of a sampler that can strictly sample according to the carrier's distribution. The swift advancement of deep generative models and the prevalence of AI-generated data have provided new techniques and environments for provably secure generative steganography, advancing research in this field.

Considering the rapid advances in computational power and the potential emergence of new attack vectors, traditional encryption techniques will likely become insecure in a few decades, necessitating more secure encryption measures to meet long-term security needs. Honey encryption is a novel technique that aims to produce a plausible-looking decoy plaintext when an attacker attempts to decrypt the ciphertext with a wrong key. Thus, even if an attacker obtains all possible plaintexts through brute-force attacks, identifying the true plaintext among them is still challenging. Such a design significantly enhances encryption security, making honey encryption an effective tool against brute-force attacks and particularly suitable for scenarios requiring long-term protection against powerful attacks.

The thesis addresses the two core needs of data privacy protection—concealment and confidentiality—and revolves around researching provably secure generative steganography and honey encryption. The main contributions and innovations of the thesis can be summarized as follows.

1. Efficient and Provably Secure Steganographic Algorithm Based on Distribution Copies

The existing provably secure steganographic algorithms can be categorized into

three types: those based on rejection sampling suffer from low embedding rates and slow speed; those based on arithmetic coding and those based on sample space grouping are capable of achieving higher embedding rates and speed, but their implementations are challenging to meet the theoretical assumptions, therefore they cannot achieve the expected security. Essentially, all the aforementioned algorithms use the index values or function values of samples to express the message. To overcome the limitations of existing algorithms, this thesis proposes an efficient and provably secure steganographic algorithm based on distribution copies. By creating multiple distribution copies by rotating the probability distribution provided by the generative models, the index values of the distribution copies are employed to express the message. Security proof is given from the perspective of distribution preservation. The embedding rate of the algorithm is significantly enhanced by recursive grouping. Taking the text generation task as an example, performance tests are conducted. Experimental results show that the embedding rate of this steganographic algorithm can approach the theoretical limit. Compared with the normal generation, the introduced additional time is relatively small, ensuring the algorithm's efficiency in practical applications. Furthermore, the algorithm is deployed on the text-to-speech task, verifying its versatility.

2. Honey Encryption Scheme for Natural Language Text Based on Deep Generative Models and Arithmetic Coding

The core of honey encryption is the distribution-transforming encoder (DTE). Existing DTEs typically leverage traditional statistical models and a fixed-length encoding scheme based on the cumulative distribution function (CDF). However, these schemes fall short in modeling capability and generalization when dealing with complex data and struggle to maintain the original distribution. Starting with the minimization of modeling and encoding losses and applying ideas from the domain of provably secure generative steganography, this thesis designs a DTE based on deep generative models and arithmetic coding. It begins by tokenizing the plaintext, then encodes the tokens into a seed using arithmetic coding, leveraging the probability distribution provided by a deep generative model. To eliminate distinguishable characteristics, pseudo-random padding schemes are conceived at the plaintext and seed levels. The experimental results demonstrate that the proposed scheme achieves a high compression ratio and little encoding loss. The scheme can achieve little modeling loss and high security when using large models.

Key Words: information hiding; steganography; provably secure steganography; honey encryption; generative model; arithmetic coding

目　录

插 图 清 单

表 格 清 单

第 1 章　绪　　论

本章将介绍隐写术和蜜罐加密技术的研究背景与意义、研究现状和发展趋势，并总结本文的主要工作。其中第 1.1 节介绍了研究背景与意义；第 1.2 节介绍了国内外研究现状；第 1.3 节总结了本文的研究内容与创新点；第 1.4 节概述了本文的体系结构安排。

1.1　研究背景与意义

1.1.1　隐写术在网络空间安全中的意义

计算机网络已成为最常用、最便捷的信息传输方式，每时每刻都有海量信息通过互联网传播，人们可以很方便地在网络空间中发布作品、聊天通讯、进行电子交易。然而，随之而来的问题也层出不穷，使网络空间安全面临巨大挑战。如有敏感信息（如国家机密、商业机密、法庭证据）遭到恶意泄漏或篡改，极有可能给个人或社会带来严重损失，甚至会对国家安全造成威胁。

为了保障网络空间安全，传统的解决方案是基于密码学的，这是一个具有坚实数学理论基础的成熟领域[1-2]。尽管加密能够使得不具有正确密钥的非法用户无法正确读取消息，但是也明确提示攻击者存在秘密通信行为，从而引起攻击者的注意甚至攻击。所以，现代通信安全不仅要求信息内容保密，还希望通信过程隐蔽。如今，信息隐藏技术作为隐私保护、隐蔽通信、知识产权保护等领域的重要手段，正得到广泛应用。

隐写术（steganography）是信息隐藏的重要分支，其目的是将秘密消息隐秘地嵌入到各类信息载体数据（如文本、图像、音频、视频）中，从而以一种掩盖秘密消息的存在性、不引起攻击者注意的方式实现隐蔽通信或隐蔽存储。隐写术不仅能应用于军事通信中，还对个人隐私保护也有重要意义。隐写术可以作为密码学的补充，也可以与密码学相结合。

隐写术是追求行为安全的技术，其目标是尽可能使得隐写行为与正常行为不可区分。与隐写术相对的概念称为隐写分析（steganalysis）或隐写检测，其主要研究的是如何检测某个对象中是否使用隐写术嵌入了秘密消息。只要检测到秘密消息存在，就可以认为该隐写术被攻破了，这是因为隐写术的主要目的就是隐藏隐蔽通信本身。隐写术和隐写分析技术如同矛与盾，相互博弈并不断发展。

如今，人工智能技术的高速发展为隐写术带来了新机遇——生成式人工智能为隐写术提供了新的伪装环境和技术手段，也带来了新挑战——隐写分析技术的性能也得到了大幅提升。面对更为先进的隐写分析者，如何设计出安全性更

高甚至可证安全的隐写方法，对于保障网络空间安全乃至国家安全尤为重要。

1.1.2　文本隐写的意义

文本作为日常生活中使用最为广泛的数字载体，具有高度凝练、数据量小、易于传输等独特优势，且文本在传输过程中通常是无损的，从而具有较强的鲁棒性，以文本作为载体的隐写术有着巨大的理论价值和实用价值。然而，相较于图像、音频、视频等，文本作为一种信息编码程度更高的载体，能提供的冗余十分有限，使得将信息隐藏在其中非常困难。对于文本而已，哪怕是非常细微的修改（比如，只修改一个字符）也可能给文本语义带来巨大变化，甚至引发重大错误。如何高效、安全地将秘密消息嵌入文本中，仍是一个颇具挑战性的课题。

1.1.3　文本蜜罐加密的意义

文本是人们在日常生活中最为常用的信息载体之一，其中包含了大量的隐私数据。因此，如何安全地存储和传输文本数据是一个重要的研究课题。目前，一般的做法是通过传统密码学方法保护其机密性：在存储或传输数据之前，先使用密钥（key）将数据明文（plaintext）加密为密文（ciphertext）。这样，只有拥有正确密钥的用户才能正确地读取数据明文。

然而，传统加密方法可能会受到暴力破解攻击（brute-force attack）：攻击者截获密文后，只要他拥有足够的计算能力和时间，就能离线地枚举出所有可能的密钥，并通过解密算法获得所有可能的明文，即候选明文（candidate plaintext），真明文（true plaintext）一定包含于其中。对于使用传统加密方法得到的密文而言，找出真明文通常是容易的，因为其通常具有某些结构特征，例如，自然语言文本通常由有意义的可打印字符（对应的 ASCII 编码在 32 到 126 之间）组成，而其他候选明文通常是不可读的乱码。根据这些特征，攻击者能够以很大的把握找出真明文。当密钥空间较小时（例如，在依赖于低熵的用户选择口令的密码系统中），这个问题会加剧。另外，有些数据须在数十年内保持机密。考虑到计算能力的快速进步以及未来可能出现的新型攻击手段，现有的传统密码学很可能过几十年就不再安全，因此我们需要更强大的加密措施以应对长期安全需求。

蜜罐加密（honey encryption，HE）是 Juels 和 Ristenpart[3] 在 2014 年的欧密会上提出的一种新型密码学技术，该技术旨在提供一种超越传统暴力破解攻击界限的安全防护。蜜罐加密引入了一种称为分布转换编码器（distribution-transforming encoder, DTE）的模块，它能够实现明文和伪随机比特序列之间的相互转换，文献 [3] 中将这种伪随机比特序列称为种子（seed）。DTE 的一个设计目标是：使得对任意伪随机比特序列进行解码得到的结果的分布接近于明文分布。蜜罐加密的加密算法采用一种称为 DTE-then-encrypt 的两阶段过程：先使用 DTE 的编

码算法将明文转换为种子，再使用密钥 K 和传统加密算法（如 AES）将种子加密为密文。这样，若攻击者尝试使用错误密钥 K' 来解密密文，会得到看似合理的假明文（fake plaintext）或称诱饵明文（decoy plaintext）。因此，即使攻击者通过暴力破解获得了所有候选明文，也难以定位出其中的真明文。这样的设计显著提升了加密的安全性，使得蜜罐加密成为抵御暴力破解攻击的有效手段，特别适用于那些需要长期保护和对抗强大计算能力攻击的场景。

1.2　国内外研究现状

1.2.1　文本隐写术

根据秘密消息嵌入域的不同，文本隐写术通常可以分为基于文本格式的和基于文本内容的两类，如图 1.1所示。

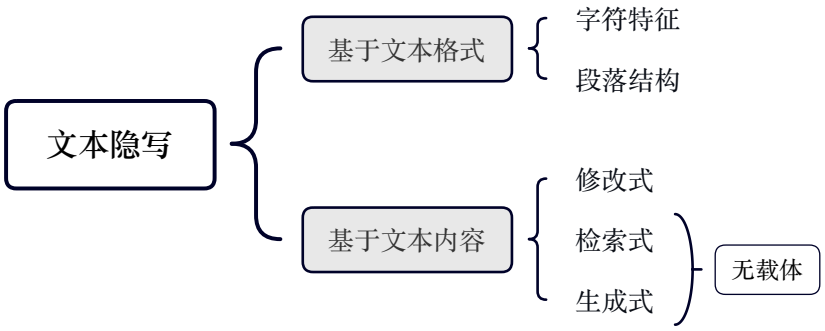


图 1.1　文本隐写分类

1. 基于文本格式的隐写术

为提升阅读体验，在一些文本文件格式（如 DOC、PDF）中，文本通常需要经过排版处理，如修改字体、段落结构等，这些处理赋予了文本丰富的格式特征，也为实施信息隐藏创造了冗余空间，基于格式的文本隐写术孕育而生。例如，Low、Brassil、Maxemchuk 等人[4-8]基于人类视觉系统的掩蔽效应，提出了字移编码、行移编码等算法，通过微调字符间距、行间距、段间距等来实现信息隐藏，利用质心检测算法提取秘密消息；Tan 等人[9]通过对字符结构进行局部微调来嵌入秘密消息；Rizzo 等人[10]利用相似字体难以分辨的现象，通过改变文本指定位置的字体样式来表达不同的消息。然而，这类基于文本格式的方法依赖于特定的文本文件格式，而且容易受到重新排版等处理影响，导致无法提取秘密消息。

2. 基于文本内容的隐写术

基于文本内容的隐写术也称为语言隐写术（linguistic steganography），有效融合了自然语言处理（natural language processing，NLP）领域的研究进展，将秘密消息嵌入到词汇、句法、语义等语言特征中，通常可以划分为修改式、检索式和生成式三种[11]。

修改式隐写方法在维持语义不变、语法结构合理的前提下，通过句法变换[12-16]（如主被动语态变换、主语后置、宾语前置、形式主语添加）、同义表述替换[17-22]等操作嵌入秘密消息。这类方法充分挖掘了自然语言语义和语法等方面的冗余，但算法需依托于现有的句法分析、词义消歧、语义抽象提取等自然语言处理技术。囿于现有技术的局限，载密文本仍存在统计失真和语义偏离现象，难以抵抗基于统计特征的隐写检测，而且嵌入容量通常不高（每句话只能嵌入几个比特）。而且，基于冗余信息修改的传统隐写方法的安全性完全依赖于算法的保密性，无疑违背了柯克霍夫原则。

为克服传统信息隐藏方法的固有局限，语言隐写术转向无载体隐写，即以秘密消息和密钥为驱动，从文本大数据集中检索或生成符合映射的文本。这种全新的思路一经提出，便吸引了该领域众多研究者的关注。根据实现机制的不同，文本无载体隐写可分为文本检索式隐写和文本生成式隐写。

检索式隐写方法[23-25]也称为选择式隐写方法，首先对载体集上的所有载体进行编码，然后根据秘密消息选择不同的载体进行传输。这类方法的优点是载体总是绝对自然的（100% natural），但是缺点也很明显——它的隐藏容量远小于传统隐写方法。

随着自然语言处理技术的持续进步，基于传统统计语言模型和神经网络语言模型的相关研究不断取得新的进展，生成式隐写方法开始涌现。不同于检索式隐写方法，生成式隐写方法根据秘密消息直接生成相应的隐写载体（即载密），秘密消息在生成过程中被嵌入。1992 年，为了使生成的文本在字符级别的概率分布上与正常文本接近一致，Wayner[26]提出了模仿函数（mimic function），首先统计大量文本数据集中各字符出现概率，再根据概率对各字符分配 Huffman 编码的码字。消息嵌入算法是通过 Huffman 编码的解码过程将秘密消息转为字符序列，消息提取算法则是通过 Huffman 编码的编码过程将字符序列转为秘密消息。然而，该方法仅考虑了文本中字符级别的统计分布，没有考虑语义关系，生成的文本没有自然意义，隐蔽性很差。1997 年，Chapman 和 Davida[27]试图使用语法模板生成文本，希望生成的文本可以符合这些语法规则。这种方法生成的文本质量虽有很大提升，但是一般遵循非常单一的模式，也缺乏对语义的考虑，容易被识别。这些早期工作告诉我们，不仅应考虑字符级别的概率分布一致性或者遵循单一的语法模板，还应从更高级的语义角度考虑，以生成更符合正常文本语义表达模式的隐写文本。

之后，随着自然语言处理技术的发展，研究人员试图将这些技术用于生成式文本隐写上。2010 到 2016 年，文献 [28-31] 利用 n-gram 语言模型即 $(n-1)$ 阶马尔可夫模型（Markov model）对文本进行建模，基于条件概率对候选词编码，根据秘密信息控制文本的生成过程。在统计自然语言处理领域，通常使用条件概率

的链式乘法来计算整段文本的概率：

$$P(\mathbf{x}) = P\left(x_1\right) P\left(x_2 \mid x_1\right) \cdots P\left(x_m \mid x_1, x_2, \cdots, x_{m-1}\right) = \prod_{t=1}^{m} P\left(x_t \mid x_{<t}\right) \tag{1.1}$$

其中，$\mathbf{x}$ 是一个由 m 个 token 组成的句子，x_i 表示其中的第 i 个 token。这里的 token 是自然语言处理领域的一个专业术语，一般翻译为“词元”，指的是文本中的最小语义单位，可能是单词、子词（subword）、标点符号等。马尔可夫模型未考虑之前所有 $t-1$ 个词元，而只考虑其中的最后 $n-1$ 词元，即

$$P_{\text{Markov}}(\mathbf{x}) = \prod_{t=1}^{m} P\left(x_t \mid x_{t-n}, \cdots, x_{t-1}\right) \tag{1.2}$$

总结来说，马尔可夫模型存在以下缺点：1) 未考虑之前所有词元，这限制了其语言建模的能力，尤其是对长期依赖的建模能力；2) 只是使用频率来近似概率，这使得预测的概率分布可能很稀疏且泛化能力不强；3) 存储一个 n-gram 数据库的成本关于 n 呈指数增长，即使 n 不是很大（如 $n=5$ 时），成本也是难以接受的。正是因为马尔可夫模型有这些缺点，基于它的隐写方法生成的文本质量一般。

2017 年以来，伴随着人工智能和深度学习技术的高速发展，生成式文本隐写使用的语言模型从原先的统计语言模型转为基于深度学习的语言模型，生成文本的质量也取得了飞跃发展。Fang 等人[32]首次将循环神经网络（recurrent neural network，RNN）引入生成式文本隐写领域，他们首先使用大规模文本语料库训练了一个长短期记忆（long short-term memory，LSTM，RNN 的一个变体）模型，使其具有预测给定上文 $x_{<t}$ 条件下下一个词元的概率分布的能力。隐写步骤如下：1) 将词表（vocabulary，预设置的所有可能的词元组成的集合）中的所有词元尽可能均匀地装入 $2^{|B|}$ 个箱子中（之后固定不变）；2) 将秘密消息（比特序列形式）切分成若干个长度为 $|B|$ 的消息分段；3) 在每一时间步 t，模型在上文条件下，预测下一个词元的概率分布，并将第 t 个消息分段转为十进制数，将其对应的箱子中概率最高的词元作为下一个词元（例如 $|B|=3$，当前消息分段为 110，则选取第 6 个箱子中概率最高的词元作为下一个词元，其中 6 为二进制 110 对应的十进制）。这种方案虽然能生成自然度较好的文本，但是显然破坏了正常文本的分布，安全性较差。

2018 年，Yang 等人[33]提出了 RNN-Stega，如图 1.2 所示。他们也训练了一个 LSTM 模型，在每个时间步，在上文条件下预测下一个词元的概率分布。不同于 Fang 等人的方法[32]，RNN-Stega 在每个时间步根据预测出的分布动态地创建 Huffman 树，消息嵌入算法是通过 Huffman 编码的解码过程将秘密消息转为词元序列，消息提取算法则是通过 Huffman 编码的编码过程将词元序列转为秘密消息。对比 [32]，RNN-Stega 能生成与正常文本的概率分布更相近的隐写文本。

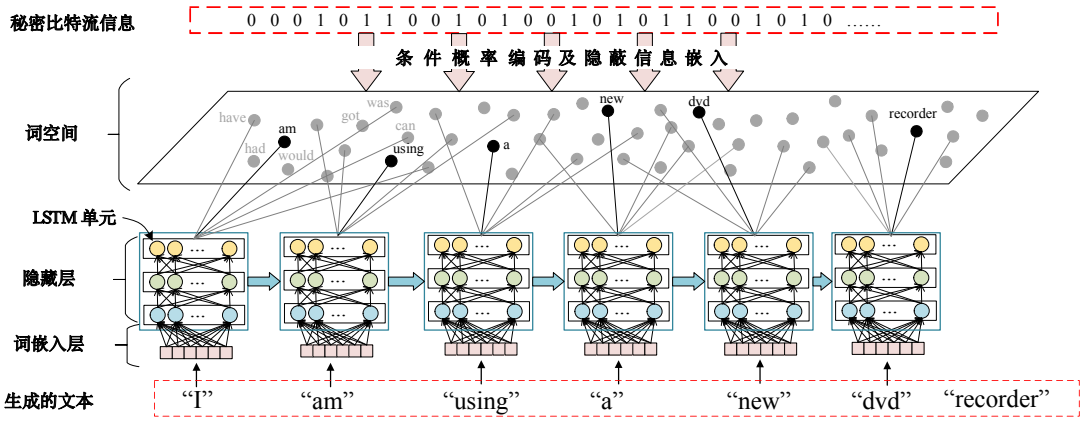


图 1.2　RNN-Stega 示意图[33]

2019 年，Dai 和 Cai[34] 与 Ziegler 等人[35] 分别从两个方面对 RNN-Stega 进行优化。Dai 和 Cai[34] 将 Huffman 编码后每个词元的隐写等效概率构成的分布称为 Huffman 分布，并预设一个阈值 δ。在某个时间步，若 Huffman 分布和原始分布的距离小于 δ，则嵌入消息（相当于从 Huffman 分布采样）；否则不嵌入消息（从原始分布采样），“耐心地”等待下一次嵌入机会。他们将这个方案称为 Patient-Huffman。Ziegler 等人[35] 则将 RNN-Stega 中使用的 Huffman 编码换成了性能更好的算术编码（arithmetic coding），提出了 Neural Linguistic Steganography（下文简称为 NLS），如图 1.3 所示，他们使用固定 k 值对每个时间步模型预测出的概率分布进行 top-k 截断。此外，他们都将 RNN-Stega 中使用的 LSTM 模型换为更先进的 Transformer 模型（具体地，是由 OpenAI 提出的在超大规模文本数据集 WebText 上进行自监督训练得到的 GPT-2 模型[36]），使得生成文本质量更优，隐写系统部署起来也更容易。

2020 年，Shen 等人[37] 指出了 NLS 使用固定的 top-k 截断带来的问题：在一个时间步，如果 k 太小，会导致隐写不可感知性较差；如果 k 太大，会降低隐写速度。鉴于此，他们将 NLS 中固定 k 值改为在每个时间步自适应地确定 k 值。他们将这个方案称为自适应算术编码（self-adjusting arithmetic coding，SAAC），实验发现 SAAC 对比 NLS 无论是安全性还是嵌入率都有所提升。

上面这些方法都没有正式地对安全性进行证明，2021 年研究者们不约而同地开始将目光投向可证安全隐写。Kaptchuk 等人[38] 指出，对于基于算术编码的隐写方案，如果在每次采样时秘密消息没有被重新加密（re-encrypt），则会引发随机性重用（randomness reuse）问题，可能有信息泄露风险。为了克服这个问题，他们提出了改进后的隐写方案 Meteor。实际上，他们只对 NLS 进行了简单的修改：第一，与 NLS 要进行一系列缩小区间操作不同，Meteor 在每一时间步都是从 $[0, 1)$ 采样；第二，每一时间步执行消息嵌入算法之前，先对消息进行重新加密，以防止随机性重用问题。然而，由于第一项修改，Meteor 在容量上比 NLS

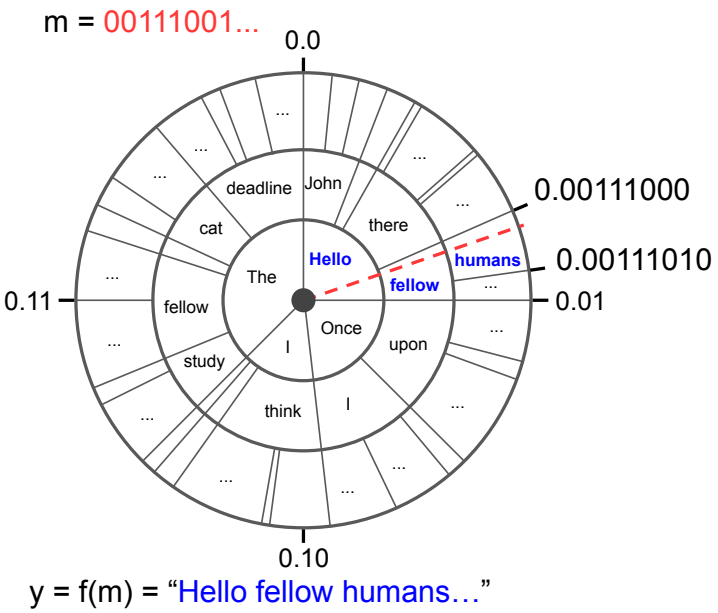


图 1.3　基于算术编码的生成式文本隐写方法[35]

低了很多。此外，在 Meteor 作者提供的实现代码中，原始分布经历了一系列修改操作才成为隐写采样的实际分布，这些操作给原始分布带来了严重破坏，因此 Meteor 无法达到预期的安全性。Zhang 等人[39]提出了一种基于基于样本空间分组的隐写构造 ADG（adaptive dynamic grouping），并证明了这种方法在理想情况下的 KL 散度等于 0。他们具体是这样做的：将词表中的所有词元划分成尽可能均匀的 2^r 个分组（每个组中词元的概率之和约等于 $1/2^r$），然后读取 r 比特消息，从相应的分组随机采样出一个词元，这样就能嵌入 r 比特消息。他们还通过递归思想提升了嵌入容量。然而，证明所依赖的条件是能够将所有词元完美均匀地划分成概率之和相等的组，而在具体实现中，由于词元的概率分布是离散的，这个条件几乎不可能满足，因此 ADG 也无法达到预期的安全性。

在 RNN-Stega 之后，还有一种研究路线是关注生成符合特定应用场景下文本的整体统计分布模式的隐写文本，或者使得生成文本的语义表达更符合人类认知模型，代表工作有基于变分自编码器（variational auto-encoder，VAE）的 VAE-Stega[40]、基于图神经网络（graph neural network，GNN）的 Graph-Stega[41]、使用 Encoder-Decoder 模型提升认知不可感知性（cognitive-imperceptibility）的 [42]。这条研究路线不是本文关注的重点，所以在此不过多介绍。

本文将生成式文本隐写的发展历史总结为图 1.4，这些论文大多发表在自然语言处理或安全的顶会顶刊上。发展的大致趋势是：由只满足简单的字符级概率分布或单一的语法模板到更考虑上下文语义关联再到可证安全，由从头开始训练的 RNN（LSTM）模型到经过大规模语料库上预训练的 Transformer 模型，隐写文本的质量有了大幅提升，隐写文本与正常文本的统计特征差异在缩小。

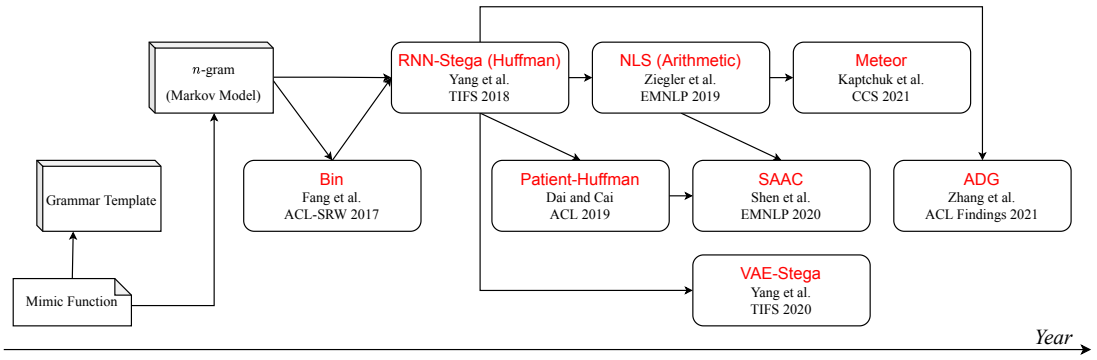


图 1.4　生成式文本隐写的发展

1.2.2　文本蜜罐加密

蜜罐加密的核心是分布转换编码器（DTE），它负责明文与种子之间的相互转换。现有的 DTE[3,43-47] 普遍采用传统的统计模型（如 *n*-gram 模型、概率上下文无关模型）来对明文分布进行建模，并通过基于累积分布函数（cumulative distribution function，CDF）的定长编码方案将明文转换为种子。这种 DTE 最初是针对较为简单的数据类型而设计的，如信用卡号、个人识别码（PIN）、银行卡安全码（CVV）等[3]。然而，传统的统计方法在处理更加复杂的数据（如自然语言文本）时显示出建模能力、泛化能力差，在这个深度生成模型盛行的时代已不再是最优选择；此外，基于 CDF 的逆变换采样编码方案是一种定长编码方案，需要使用更长的编码来表示数据，导致了存储和传输开销的增加。如何设计出适合自然语言文本这种数据的更加安全、高效的 DTE，是值得研究的课题。

1.3　论文研究内容与创新点

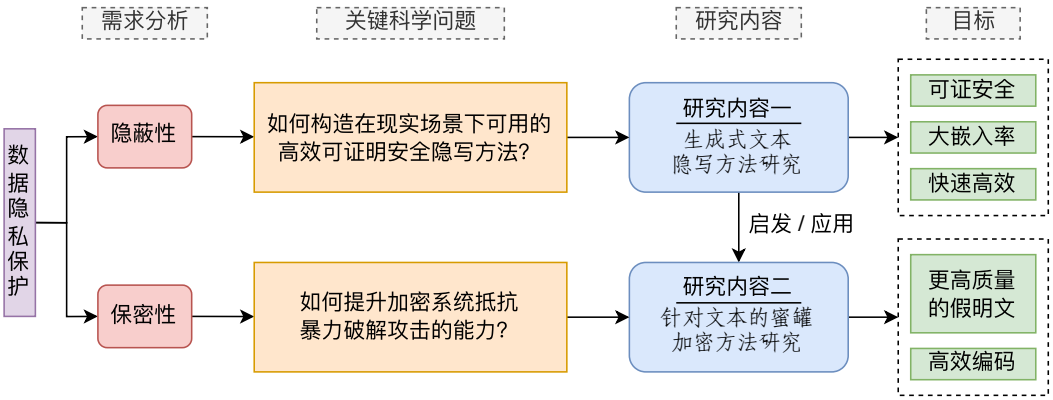


图 1.5　本文研究内容框图

图 1.5 展示了本文的研究内容框图。本文包括两项研究内容：一是设计在现实场景中仍能保持安全性的高效生成式可证安全隐写构造，二是将生成式可证安全隐写的思想应用于文本蜜罐加密系统的设计中，从而改进文本蜜罐加密系

统。这两项研究内容虽然分别是从隐蔽性和保密性两个视角考虑数据隐私保护，但是文本蜜罐加密系统的核心 DTE 实际上是生成式文本隐写的一种变化形式。具体地，如图 1.6 所示，生成式文本隐写的嵌入算法是将比特序列转换为自然语言文本的过程 f，而提取算法则是将自然语言文本转换为比特序列的过程 g；文本蜜罐加密的 DTE 编码算法和解码算法正好可以分别对应 g 和 f。此外，生成式文本隐写与文本蜜罐加密都具有分布保持的追求。

具体研究内容如下。从对随机采样中区间分配方案的观察出发，提出一种基于分布副本的高效可证安全隐写构造，通过循环移位的方式为原始分布创建多个副本，使用分布副本的索引值表达消息；通过递归分组的思想，将这种隐写构造的嵌入率提升至接近理论极限。从生成式文本隐写与文本蜜罐加密之间的关系出发，将生成式可证安全隐写的思想应用于文本蜜罐加密系统的设计中，提出一种基于深度生成模型和算术编码的文本蜜罐加密方案，能产生质量更高、更符合自然语言文本分布的诱饵明文，同时大幅降低存储和传输开销。

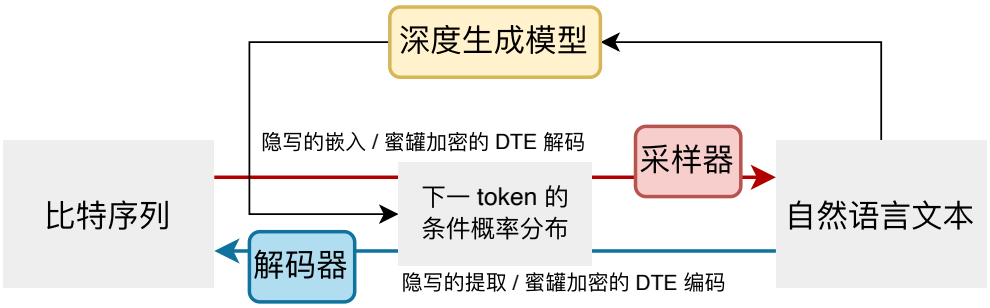


图 1.6　隐写与蜜罐加密的关系

1.4　论文体系结构

本学位论文共分为五章，论文结构如下：

第 1 章 绪论：介绍了本学位论文的研究背景与意义、国内外研究现状与发展趋势，对本学位论文的研究内容与创新点、体系结构进行总结。

第 2 章 相关理论与方法：介绍了隐写模型、修改式隐写、隐写安全性、经典可证安全隐写构造、高效可证安全隐写构造，分析了现有的追求可证安全的隐写构造在实践中的问题，介绍了蜜罐加密的相关理论与方法。

第 3 章 基于分布副本的生成式可证安全隐写构造：从对随机采样中区间分配方案的观察出发，提出了基于分布副本的可证安全隐写构造，通过递归分组的思想，将这种隐写构造的嵌入率提升至接近理论极限。

第 4 章 基于生成式可证安全隐写的蜜罐加密方案：将生成式可证安全隐写领域的思想应用于蜜罐加密，从减小 DTE 的建模损失和编码损失入手，设计了基于深度生成模型和算术编码的文本蜜罐加密方案。

第 5 章 总结与展望：对本学位论文的研究工作进行了总结，并对未来研究方向进行了展望。

第 2 章　相关理论与方法

本章主要介绍隐写和蜜罐加密的相关理论与方法。其中第 2.1 节介绍隐写模型；第 2.2 节以经典载体图像为例介绍了传统的修改式隐写方法；第 2.3 节介绍隐写安全性定义；第 2.4 节介绍经典可证安全隐写构造；第 2.5 节介绍深度生成模型及其对可证安全隐写的意义；第 2.6 节介绍基于深度生成模型的高效可证安全隐写构造；第 2.7 节介绍蜜罐加密的相关理论与方法。

2.1　隐写模型

隐写术是信息隐藏技术的一个重要分支，可以通过 Simmons 于 1984 年提出的囚徒问题（Prisoners' Problem）模型[48]定义：Alice 和 Bob 被关押在监狱的不同牢房中，试图策划一起越狱计划。然而，他们相互交流的唯一信道被典狱长 Eve 严格审查，一旦 Eve 察觉到他们通信中存在任何异常，如非法词语、加密数据或异常符号，她就会切断信道，并将他们单独监禁。因此，Alice 和 Bob 必须找到一种方法将秘密消息（message）嵌入到“看似正常”的载体中。在这个模型中，Alice 和 Bob 被称为隐写者（steganographers），Eve 被称为隐写分析者（steganalyzer）。在下文中，统一称呼秘密消息的发送方为 Alice，接收方为 Bob，攻击者（隐写分析者）为 Eve。注意，Eve 需要达到的目标仅仅是判断秘密消息是否存在，而不需要知道消息的具体内容。换言之，当 Eve 发现 Alice 和 Bob 在进行秘密通信，该隐写系统就被破解了。一般将未嵌入秘密消息的对象称为载体（cover），将嵌入秘密消息的对象称为载密（stego），使用 P_{c} 和 P_{s} 分别表示载体分布和载密分布。载体分布 P_{c} 也可以称为信道分布 $\mathcal{D}$。攻击者 Eve 只需判别通信中是否嵌入了秘密消息，而不需要知道消息的具体内容。

柯克霍夫原则（Kerckhoffs's principle）是密码学的一条基本原则[49]，可归纳为：密码系统的安全性应当取决于密钥，而非加密算法本身。若将其扩展到隐写术中，可以得到类似的原则：隐写系统应该以算法公开为基本前提，利用密钥确保系统安全。在囚徒问题中，假设 Eve 掌握了 Alice 和 Bob 可能使用的隐写算法的所有知识，除了 Alice 和 Bob 在进监狱之前就商定好的隐写密钥。

我们可以将信道分布 $\mathcal{D}$ 上的隐写系统 $\Sigma_{\mathcal{D}}$ 形式化为一个由概率多项式时间（probabilistic polynomial time，PPT）算法组成的三元组 $\Sigma_{\mathcal{D}} = \left(\mathsf{KeyGen}_{\mathcal{D}}, \mathsf{Encode}_{\mathcal{D}}, \mathsf{Decode}_{\mathcal{D}}\right)$[38,50]。

- **密钥生成算法** $\mathsf{KeyGen}_{\mathcal{D}}\left(1^{\lambda}\right)$：以任意 λ 长比特序列为输入，生成一个长度为 k 的共享密钥 $K \in \{0,1\}^{k}$，该密钥将在其他两个算法中使用。其中，λ

为安全参数（security parameter），密钥长度 k 与安全参数 λ 相关。

- **消息嵌入算法** $\mathsf{Encode}_D(K, \mathbf{m}, \mathcal{H})$：以密钥 K、秘密消息 $\mathbf{m} \in \{0,1\}^*$ 和信道历史 $\mathcal{H}$ 作为输入，返回载密，即长度为 l 的符号序列 $\mathbf{s} = s_1 \left| s_2 \right| \cdots |s_l$。
- **消息提取算法** $\mathsf{Decode}_D(K, \mathbf{s}, \mathcal{H})$：以密钥 K、载密 $\mathbf{s}$ 和信道历史 $\mathcal{H}$ 作为输入，返回从 $\mathbf{s}$ 中提取出的消息 $\mathbf{m}$。

2.2　修改式隐写

隐写领域发展最为成熟的是图像隐写，其他信息载体上的隐写的很多思想也是从图像隐写领域迁移得到的。为了更全面地理解隐写的相关基础理论，我们以图像隐写为例梳理传统的修改式隐写方法设计思想的发展历程，包括图像空域和 JPEG 域的非自适应隐写和自适应隐写。

2.2.1　空域非自适应隐写

最不重要比特替换（least significant bit replacement，简称 LSB 替换）可能是最简单、最为人所知的隐写算法，其利用了人眼对相近颜色不敏感这个特点。以灰度图像为例，每个像素通常使用一个 8 比特数值来表示灰度信息，总共有 $2^8 = 256$ 种不同的灰度级别；而在 RGB 图像中，每个像素通常使用 3 个 8 比特数值来表示颜色，总共有 256^3 种不同的颜色。对于人眼来说，相邻的灰度或颜色在视觉上难以区分。在 LSB 替换算法中，发送方需要准备一张图像作为载体图像，隐写嵌入过程就是将这张图像上的所有像素的 LSB 替换为秘密消息比特序列，得到载密图像。提取方在接收到载密图像后，只需按照相同的顺序提取载密图像上所有像素的 LSB，即可恢复出嵌入的消息。消息通常是先被加密成伪随机比特序列再被嵌入，所以对于 LSB 替换来说，消息密文比特有 50% 的可能与原始 LSB 一致，所以平均来说每嵌入 1 比特消息需要修改 0.5 个像素，嵌入效率为 2 比特/次。尽管载体图像和载密图像在视觉上很接近，但 LSB 替换算法存在“值对效应”[51]，容易被基于统计的隐写分析方法如卡方检测[52]、RS 检测[53]有效检测。如图 2.1 所示，“值对效应”的出现是因为 LSB 替换过程中，像素值 $2n$ 只会与 $2n+1$ 相互转换，但并不会与 $2n-1$ 转换，其中 $n \in [0, 127] \cap \mathbb{Z}$，这会使得“值对”样点数量上会更加接近。

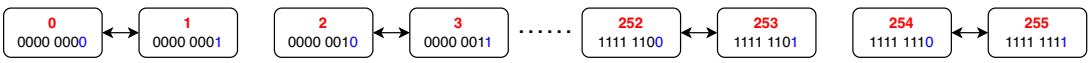


图 2.1　LSB 替换的“值对效应”

为了打破“值对效应”，一种改进算法是 LSB 匹配（LSB matching）[54]。在 LSB 匹配中，同样是利用 LSB 来表达秘密消息，当需要修改某个像素的 LSB 值

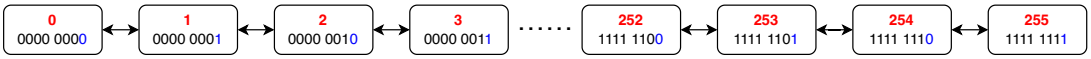


图 2.2　LSB 匹配打破了“值对效应”

时，对这个像素值进行随机加减 1。在数学上，LSB 匹配可以表示为

$$y_i = \begin{cases} x_i, & \mathrm{LSB}(x_i) = m_i \\ x_i \pm 1, & \mathrm{LSB}(x_i) \neq m_i \end{cases} \tag{2.1}$$

其中，±1 表示各有 50% 的概率进行 +1 或是 −1。在 LSB 匹配过程中，每个像素值与左右相邻的值都可能发生转换，从而解决了“值对效应”问题，如图 2.2 所示。LSB 匹配的嵌入效率也是 2 比特/次，与 LSB 替换相同。LSB 匹配开创了加减 1 嵌入（±1 embedding）这类隐写方法，直到今天它依然是最重要的、使用最广泛的隐写方法。2006 年，Mielikainen[55] 观察到，LSB 匹配算法不仅可能修改 LSB 位，还可能修改次 LSB 位。他将像素两两成对，通过利用次 LSB 信息，将嵌入效率提升至 2.67 比特/次。

加减 1 嵌入使得像素值有三种修改方向：不修改、+1 或 −1，理论上最多能承载 $\log_2 3$ 比特消息。基于这个观察，Zhang 和 Wang[56] 于 2006 年提出称为利用修改方向（exploiting modification direction，EMD）的隐写编码：以 n 个像素为一组，每组至多允许对其中 1 个像素进行 ±1 修改（也可能不修改），那么每组有 $(2n+1)$ 种修改方向，可以嵌入 $\log_2(2n+1)$ 比特消息，将嵌入效率提升至 $\frac{(2n+1)\cdot\log_2(2n+1)}{2n}$ 比特/次。之后，Fridrich 和 Lisonek[57] 基于图着色理论给出了与 EMD 相同的算法。

Crandall[58] 最早发现隐写编码与线性分组纠错码的内在联系。Westfeld[59] 根据这一性质，基于汉明码设计了经典的隐写矩阵编码算法 F5。Fridrich 等人[60] 通过结合矩阵编码和随机线性码设计了编码方案，并基于低密度生成矩阵（low-density generator matrix，LDGM）构造了具有大嵌入效率的隐写编码[61]。Zhang 等人[62] 通过结合汉明码和湿纸编码，给出了可逼近率失真界的编码方法 ZZW。

2.2.2　空域自适应隐写

第 2.2.1 小节中所介绍的算法都基于这样一个假设：修改图像中不同像素所引入的失真（或称代价）都是相同的，基于这种假设的隐写模型称为常数失真模型。然而，这种假设并没有充分利用载体不同区域的特征来提升安全性。例如，对于图像来说，在相同的修改幅度下，修改纹理复杂区域比修改光滑区域更难被检测出来。基于这个观察，我们应该为图像不同区域的像素定义不同的隐写失真，这种模型称为自适应失真模型。自适应隐写追求在嵌入容量一定时最小化总失真，或者在总失真一定时最大化嵌入容量。

对自适应隐写的研究主要可以分为两类：一是设计隐写编码，追求以一种最小化总失真的方式嵌入消息；二是定义合理的隐写失真函数，为载体上的每个元素定义不同的隐写代价，鼓励优先修改不易被检测到的区域上的元素。

在设计隐写编码方面，Filler 等人于 2010 年提出了可以逼近隐写率失真界的基于卷积码的校验子格编码（syndrome-trellis codes，STC）[63-64]。Li 等人于 2020 年提出了基于极化码的极化隐写码（steganographic polar codes，SPC）[65]，能以更低的嵌入时耗逼近隐写率失真界。Yao 等人于 2023 年建立了从信源编码到自适应隐写码的通道，提出了基于 LDGM 码的逼近隐写率失真界的自适应隐写编码[66]。

在 STC 编码问世以来，自适应隐写逐渐成为隐写的主流，研究者开始关注如何定义能合理反映隐写修改代价的隐写失真函数。Pevný 等人[67]提出了最早采用 STC 编码的自适应隐写算法 HUGO（highly undetectable stego），它的思想是：因为隐写的目标是尽可能避免被隐写分析方法检测到，所以直接针对隐写分析特征设计失真函数——如果一个像素在修改后导致的 SPAM（subtractive pixel adjacency model）隐写分析特征[68]变化小，则认为这个像素的修改代价就小。这种失真函数使得隐写修改更加集中于轮廓附近区域。得益于这种内容的自适应性，HUGO 能够有效地抵御基于 SPAM 特征的隐写分析。然而，攻击者可以通过其他隐写分析特征来检测 HUGO。考虑到小波系数可以反映载体在局部的空域和频率特性，WOW（wavelet obtained weights）[69] 和 S-UNIWARD（spatial-universal wavelet relative distortion）[70]通过使用方向小波滤波器定义隐写失真，并能支持采用二元或三元 STC 嵌入，它们倾向于选择纹理复杂区域进行隐写嵌入。使用 SRM（spatial rich model）[71]进行隐写分析实验的结果表明，WOW 和 S-UNIWARD 的安全性好于 HUGO。随后，Li 等人对隐写失真分布给隐写抗检测性带来的影响进行了探讨，提出三个隐写失真定义原则[72]：复杂度优先原则、扩散原则和修改聚集原则，并提出一种简单有效的隐写失真函数 HILL（high-pass, low-pass, and low-pass）[73]。HILL 通过使用一个高通滤波器的滤波残差来刻画载体图像的纹理复杂程度，并使用两个低通滤波器来实现失真扩散，取得了比 HUGO、WOW 和 S-UNIWARD 更好的抗隐写分析性能。

以上隐写失真函数的设计都是启发式的，它们缺乏对隐写失真和统计可检测性之间关系的理论探究，与之相对应的是基于统计模型的隐写失真函数，其试图最小化载体与载密之间的差异或最优检测器的性能，从统计模型的意义上保证隐写性能。Fridrich 和 Kodovský[74]构建了载体和载密分布的高斯模型，通过最小化两者之间的 KL 散度，设计了基于多元高斯模型的隐写算法 MG（multivariate gaussian）。使用 SRM 进行隐写分析实验的结果表明，MG 的安全性与 HUGO 基本相当。虽然 MG 没有显著提升安全性，但它提供了一种新的隐写失真函数设计

视角，为后续 Sedighi 等人设计的更加完善的 MiPOD 算法（minimizing the power of optimal detector）[75]打下基础。

上述隐写失真函数假设载体的每个元素的隐写失真都是相互独立、互不影响的，整体失真等于各元素失真的累加和，这类失真估计模型被称为加性模型。事实上，载体的不同元素（尤其是相邻元素）之间是有相关性的，而加性模型忽略了这种相关性。考虑到载体不同元素隐写修改之间的相互影响，Li 等人[76]以及 Denemark 和 Fridrich[77]分别提出了 CMD 和 Synch 非加性隐写算法，这些算法将载体图像分割成多个部分，第一部分按照原始失真进行消息嵌入，其余部分的失真则会根据先前已发生的隐写修改进行动态调整，促使邻近像素修改方向尽可能一致。使用 SRM 进行隐写分析实验的结果表明，相比加性模型，它们的抗检测性能有显著提高。Zhang 等人[78]考虑同时修改多个相邻像素的联合失真，提出了非加性隐写编码 DeJoin（Decompose Joint distortion），将非加性编码问题等效地分解成多个加性编码问题，这一策略使得嵌入过程更为高效。

2.2.3　JPEG 域隐写

JPEG 是由联合图像专家小组（Joint Photographic Experts Group）开发的一种广泛使用的有损压缩标准。针对 JPEG 图像的隐写一般将消息嵌入到 JPEG 压缩编码过程中量化后的 DCT 系数中。与空域隐写类似，JPEG 域隐写也可根据载体每个元素的隐写失真是否相互独立划分为加性模型和非加性模型。

Upham 将 LSB 替换算法拓展应用到 JPEG 域，编写了隐写软件 JSteg[79]，通过将载体图像除 0 值和 1 值外的 DCT 系数的 LSB 替换成秘密消息比特序列来实现隐写嵌入。与空域图像上的 LSB 替换类似，JSteg 也会引起“值对效应”，可以被基于统计的隐写分析方法有效检测。随后，一系列试图保持统计特征的隐写算法相继被提出。Provos 提出了 OutGuess[80]算法，主要包含两个步骤：第一步，根据隐写密钥从所有 DCT 系数中选择一个伪随机子集，通过将这个子集中除 0 值和 1 值外的 DCT 系数的 LSB 替换成秘密消息比特序列来实现隐写嵌入；第二步，对这个子集之外的 DCT 系数进行校正，使得隐写修改前后 DCT 系数直方图一致。类似的算法还有 MB（model-based）[81]和 Steghide[82]。然而，它们仅仅能在一定程度上保持一阶直方图，在高阶统计量上仍然存在差异，因此抗检测性能弱。

在 JPEG 图像的自适应隐写方面，需要为每个可嵌入的 DCT 系数估计修改代价。J-UNIWARD（JPEG-universal wavelet relative distortion）[70]通过 DCT 系数的修改对方向小波滤波残差的影响定义隐写失真。Guo 等人[83]观察到，由于小值 DCT 系数较多，随机地选择隐写修改位置会使得 DCT 系数分布曲线发生形变，从而容易被检测，基于这个观察，他们提出了 UED（uniform embedding distortion）[83]，

优先选择大值 DCT 系数进行修改，并压制在小值 DCT 系数修改。由于 UED 构造相对简单，其速度明显优于 J-UNIWARD，但是安全性不如 J-UNIWARD。之后，Guo 等人提出了 UERD（uniform embedding revisited distortion）[84]，通过综合考虑 DCT 系数和分块所处频率等因素进一步提升了 UED。Li 等人[85]从抑制隐写修改的块效应着手，提出一种适用于 JPEG 域自适应隐写的块边界连续性（block boundary continuity，BBC）原则，通过鼓励相邻块间相同频率 DCT 系数联合修改来保证图像的空域连续性和邻域相关性，进一步提升了隐写安全性。

然而，上面介绍的所有隐写算法都停留在经验安全。对于经验安全隐写算法而言，攻击者总是存在一个不可忽略的优势。即使这个优势可能看起来很小，但是攻击者总能通过一定数量的样本，以很高的置信度识别出隐写者[86]。因此，实现可证安全隐写一直是隐写领域的一大追求。

2.3　隐写安全性

在隐写领域，可证安全的追求由来已久。1949 年，Shannon 在其经典论文 Communication Theory of Secrecy Systems[87]中开宗明义地指出了构建隐蔽系统的困难性。

直到二十世纪九十年代，学界开始模仿密码学中的安全性定义来定义隐写的安全性。1998 年，Cachin[88]首先从信息论的角度建模隐写系统的安全性。直觉上，载体分布 P_{c} 和载密分布 P_{s} 越接近，攻击者 Eve 做出错误判断的概率就越大，那么可以将隐写系统的安全性量化为 P_{c} 和 P_{s} 之间的相对熵（relative entropy）即 KL 散度（Kullback-Leibler divergence，KLD），计算公式为

$$D_{\mathrm{KL}}\left(P_{\mathrm{c}} \| P_{\mathrm{s}}\right)=\sum_{\mathbf{x}} P_{\mathrm{c}}(\mathbf{x}) \log_2 \frac{P_{\mathrm{c}}(\mathbf{x})}{P_{\mathrm{s}}(\mathbf{x})} \tag{2.2}$$

其衡量了两个分布之间的差异。如果 $D_{\mathrm{KL}}\left(P_{\mathrm{c}} \| P_{\mathrm{s}}\right)=0$，则该隐写系统被认为是绝对安全的（perfectly secure），在这种情况下，P_{c} 与 P_{s} 严格相等，隐写分析器判别一个对象 $\mathbf{x}$ 是载体还是载密的能力无法好于随机猜测。如果 $D_{\mathrm{KL}}\left(P_{\mathrm{c}} \| P_{\mathrm{s}}\right) \leqslant \epsilon$，则称隐写系统是 ϵ-安全的。

2002 年，Hopper 等人[50] 和 Katzenbeisser 与 Petitcolas[89] 分别独立地从计算复杂性理论的角度给出了隐写安全性定义。他们假设存在一个可以严格按照信道分布 $\mathcal{D}$ 进行随机采样的黑盒预言机 $\mathcal{O}_{\mathcal{D}}$，且攻击者 Eve 可以访问 $\mathcal{O}_{\mathcal{D}}$，那么可以将攻击者描述成在进行一个尝试区分一个对象是隐写采样（嵌入了消息的采样，对应 $\mathsf{Encode}_{\mathcal{D}}$）的输出还是正常采样（未嵌入消息的采样，对应 $\mathcal{O}_{\mathcal{D}}$）的输出的区分游戏。若 Eve 认为一个对象是隐写采样的输出，她会报阳性 1，否则报阴性 0。那么，可以将攻击者优势定义为她对这两种采样方法分别报阳性的概

率之差。如果对于所有的 PPT 时间攻击者 $\mathcal{A}_D$，优势都是关于安全参数 λ 可忽略的，即

$$\left|\Pr\left[\mathcal{A}_D^{\mathsf{Encode}_D(K,\cdot,\cdot)}=1\right]-\Pr\left[\mathcal{A}_D^{\mathcal{O}_D(\cdot,\cdot)}=1\right]\right|<\mathsf{negl}(\lambda) \tag{2.3}$$

那么称该隐写系统是计算安全的。其中 $\mathsf{negl}(\lambda)$ 表示关于安全参数 λ 的可忽略函数。可忽略函数的定义为：如果一个函数 $\mu(x)$ 是可忽略函数，那么对于任意一个正多项式 P，存在一个 $N_c>0$，使得对于所有的 $x>N_c$，都有 $\mu(x)<1/P(x)$ 成立。

信息论安全和计算安全的区别在于：如果不对攻击者的计算能力进行限制，都能保证其优势为零，就是信息论安全。如果退一步，对于任意计算能力受限的攻击者而言，优势都可忽略，就是计算安全。

2.4　经典可证安全隐写构造

自从隐写的安全性定义被提出以来，研究者们相继设计了几种追求可证安全的隐写构造。这一节将逐一介绍这些方法。

2.4.1　基于拒绝采样的隐写构造

2002 年，Hopper 等人[50]在给出隐写的计算安全定义后提出了基于拒绝采样（rejection sampling）的隐写构造。在这个构造中，假设通信双方共享一个可以严格按照信道分布 D 进行随机采样的黑盒预言机 $\mathcal{O}_D$ 和一个无偏函数 f。无偏函数 f 满足：使用 $\mathcal{O}_D$ 采样出的样本对应的 f 函数值是等概的。待隐写的秘密消息是经过安全加密算法加密的：通信双方需要提前约定一个加密密钥；发送方 Alice 在执行消息嵌入算法之前需要先对消息进行加密；接收方 Bob 在执行消息提取算法之后需要对消息密文进行解密。在每一时间步 t，Alice 如果要嵌入 1 比特消息 $m_t\in\{0,1\}$，她通过拒绝采样获得一个满足 $f\left(s_t\right)=m_t$ 的符号 s_t，发送给接收方 Bob；Bob 收到 s_t 后，直接通过计算 $m_t=f\left(s_t\right)$ 获得消息比特 m_t。Bob 将所有时间步提取的消息比特连接，便可以获得完整消息。算法 2.1 展示了拒绝采样的大体流程：使用 $\mathcal{O}_D$ 采样出一个符号 s_t，若其函数值 $f\left(s_t\right)$ 不等于消息 m_t，就重新采样符号 s_t；由于可能一直无法采样出符合条件的符号，所以一般会设置一个最大采样次数 N——若采样次数已为 N，则不再重新采样，而这样会引入错误，所以还需要在加密算法与消息嵌入算法之间引入纠错编码。由于篇幅原因，本文不再对纠错编码进行详细介绍。他们将这种隐写构造的安全性归约为加密算法的安全性，即消息密文与真随机序列不可区分。2004 年，von Ahn 和 Hopper[90]在欧密会上将这种隐写构造进行扩展，将原先的对称加密算法换成公钥加密算法，并引入一个概率偏差消除方法，得到了基于拒绝采样的公钥隐写

构造。2005 年，Backes 和 Cachin[91] 又在此基础上进行改进，使其能够抵御部分主动攻击。

算法 2.1　rej_samp$(f, m_t, \mathcal{O}_D, N)$：拒绝采样

Input: 无偏函数 f，待嵌入消息 m_t，预言机 $\mathcal{O}_D$，最大采样次数 N
Output: 载密 s_t

```
i ← 1
s_t ← O_D
while i < N and f(s_t) ≠ m_t do
    s_t ← O_D
    i ← i + 1
end
return s_t
```

然而，这种基于拒绝采样的隐写构造在实际应用中存在以下问题：1) 消息嵌入算法的时间复杂度随着要嵌入的消息长度呈指数级增长；2) 当信道分布的最小熵非常低时，拒绝采样算法会失败。而且，在传统数据环境①中，还缺少一个能够严格按照信道分布进行随机采样的黑盒采样器，使得这种隐写构造仅停留在理论层面，难以在实际场景中得到应用。

2.4.2　基于算术编码的隐写构造

2003 年，Le[92] 提出了基于算术编码（arithmetic coding）的隐写构造。算术编码本身是一种信源编码（即数据压缩）技术，其编码算法能够将数据压缩为长度较短的伪随机序列，解码算法则可以压缩后的伪随机序列解压缩为原始数据。基于算术编码的隐写构造利用了隐写与信源编码之间的对偶性——隐写的消息嵌入和提取算法分别对应于信源编码的解压缩和压缩算法：发送方 Alice 使用隐写的消息嵌入算法将加密过的消息“解压缩”为载密，接收方 Bob 使用隐写的消息提取算法将载密“压缩”为加密过的消息。如果数据压缩方案是完美的②，那么这种基于算术编码的隐写构造的安全性就可以归约为加密算法的安全性。与基于拒绝采样的隐写构造对比，基于算术编码的隐写构造更加高效——速度更快、容量更高，但是需要更加苛刻的条件：需要已知数据的显式概率分布，可理解为持有一个白盒采样器，而不仅仅是基于拒绝采样的隐写构造所需要的黑盒采样器。

尽管算术编码能够非常接近 Shannon 的信源编码定理[93] 所描述的理论极限，但它并不是一个完美压缩方案。因此，基于算术编码的隐写构造几乎一定会使得载体分布与载密分布有所差异，因此无法保证安全性。具体地，这种隐写构造建立了秘密消息与载密内容之间的可逆映射，任意秘密消息都有着比特序列（即二

①本文中的“传统数据环境”指深度生成模型流行之前的数据环境。在传统数据环境中，数字数据是由传感器或者人类创建的，其生成过程复杂且难以控制。

②完美压缩方案并没有一个严格的数学定义，它通常是指一种理想的数据压缩方法，可以无损地将任何一种数据压缩到其信息内容的极限（即数据的熵），而没有任何的冗余。

进制序列）的形式，那么任意秘密消息的概率都有 2^{-l} 的形式（$l \in \mathbb{N}$），相应地，载密的概率也具有 2^{-l} 的形式。然而，载体分布几乎不可能与这种载密分布完美匹配，这使得载密分布是对载体分布进行一定调整得到的结果，这种调整引入了失真。为了减少失真，可以简单地通过填充来增加消息长度，使得载体数据的概率变得更小、更接近某个 2^{-l}。随着消息长度接近无穷大，失真能够趋于 0，但这显然是不切实际的。即便如此，引入的失真是关于消息长度（而非安全参数）的一个可忽略函数，无法达到预期的安全性。而且，在传统数据环境，更是不存在能给出数据的显式概率分布的白盒采样器，使得这种隐写构造也仅停留在理论层面，难以应用于现实。

2.5　深度生成模型及其对可证安全隐写的意义

实现可证安全隐写的一个必要条件是拥有一个能够严格按照载体（或信道）分布进行精确采样的“完美采样器”。对于基于拒绝采样的隐写构造而言，需要的是一个黑盒采样器；而对于更加高效的基于算术编码的隐写构造来说，需要的是一个白盒采样器。

近十年来，深度学习在各个领域都取得了令人瞩目的成绩，深度生成模型如生成对抗网络（generative adversarial network，GAN）[94-97]、变分自编码器（variational auto-encoder，VAE）[98-99]、自回归生成模型（auto-regressive model）[36,96,100-102]、扩散模型（diffusion model）[103-106]从出现到繁荣。近五六年来，大规模预训练模型如 GPT 系列模型[36,96,101-102]、BERT 模型[107]的出现，极大地降低了深度学习研究和应用的门槛，为人工智能开启了新的篇章，深度学习进入大工业化时代，越来越多深度生成模型的应用落地。例如，2021 年微软推出了小冰岛，在这个平台上不止有人类（human beings），还有 AI beings。这些 AI 每天陪我们聊天，为我们写诗、作画、唱歌。再如，2022 年 11 月，OpenAI 推出 AI 聊天机器人 ChatGPT，一时爆火全网，仅用两个月时间月活用户数量突破 1 亿，基于 ChatGPT 的各类应用也层出不穷。自 ChatGPT 问世以来，深度生成模型正迅速渗透到人们生活和工作的各个领域中。

这些深度生成模型是在大规模数据上训练得到，可以近似各类复杂数据的分布。它们可以合成逼真的照片、精美的画作[95,97,106,108]和几乎无法区分于人类创作的文字[36,96,101]。而且，可以预见的是，深度生成模型会越来越强大，互联网上的 AI 生成数据会越来越普遍。Science 将“人工智能变得有创造力”评选为 2022 年十大科学突破之一[109]。根据美国著名咨询公司 Gartner 发布的 Top Strategic Technology Trends for 2022[110]中的预测，到 2025 年，AI 生成数据会占到所有数据的 10% 左右（而在该报告发布时，这个比例不足 1%）。

AI 生成数据的流行为隐写提供了全新的伪装场景——我们可以将秘密消息嵌入到 AI 生成数据中。需要注意的是，这种隐写只需追求生成的载密数据与深度生成模型正常生成的数据不可区分即可。要理解这一点需要回到隐写的本质，即隐写行为需要与某个正常行为不可区分，而随着深度生成模型的日渐流行，传输 AI 生成数据已成为一种正常的行为。

举例而言，GAN 和 VAE 需要先从噪声的先验分布 $p(\mathbf{z})$ 中采样出一个噪声向量 $\mathbf{z}$，将其作为生成器网络 G_θ 的输入，可以得到对象 $\mathbf{x}$，即

$$\mathbf{x} = G_\theta(\mathbf{z}),\ \mathbf{z} \sim p(\mathbf{z}) \tag{2.4}$$

它们是隐式生成模型（即不能给出明确的概率分布的生成模型），可视为黑盒采样器，我们可以将基于拒绝采样的隐写构造部署于其上。

自回归模型则是通过条件概率分布的链式法则（chain rule）分解对象 $\mathbf{x}$ 的联合概率分布，即

$$p(\mathbf{x}) = \prod_{t=0}^{T} p\left(x_t \mid x_0, \cdots, x_{t-1}\right) = \prod_{t=0}^{T} p\left(x_t \mid x_{<t}\right) \tag{2.5}$$

其中，x_t 是 $\mathbf{x}$ 的基本单元（如图像中的像素、文本中的词元）。自回归模型是显式生成模型（即能够给出明确的概率的生成模型），可视为白盒采样器，我们可以将更加高效的基于算术编码的隐写构造部署于其上。

2.6　高效可证安全隐写构造

深度生成模型的流行，使得可证安全隐写从理论走向实用，促进了该领域研究者们的研究热情。自 2018 年以来，研究者们相继提出了几种高效可证安全隐写构造。直觉上，如果想在文本生成过程中进行消息嵌入，则需要对采样过程施加某种“控制”，使其采样出的对象可以表达消息。

在这种将秘密消息嵌入到 AI 生成数据的隐写构造中，深度生成模型给出的分布即为载体分布，也可称为原始分布。隐写对采样器施加的“控制”可能导致隐写实际采样的分布与原始分布存在差异。从分布保持的角度，我们也可以将这个差异理解为隐写对原始分布的修改程度，即破坏程度。隐写的目标是尽可能减小对原始分布的修改程度，最好是完全不修改（即分布保持）。

下文中，“原始分布”和“载体分布”表示相同含义，对应正常采样即随机采样；“隐写采样的实际分布”和“载密分布”表示相同含义，对应隐写采样。

2.6.1　基于算术编码的隐写构造

2018 年，Yang 等人[111]将基于算术编码的隐写构造扩展应用到图像生成任务上，这是第一个将深度生成模型引入到可证安全隐写领域的工作。具体地说，

他们重复使用自回归生成模型预测每个像素的概率分布，并使用算术编码的解压缩算法将秘密消息转换为像素序列。之后，Chen 等人[112]类似地将这种隐写构造扩展应用到语音合成任务上。虽然这些工作解决了高效隐写构造缺少显式生成模型的问题，但是仍然存在 2.4.2 小节中指出的安全性问题。

2021 年，Kaptchuk 等人[38]对基于算术编码的隐写构造进行改造，提出了基于区间随机化可逆采样的隐写构造，将其命名为 Meteor，从理论上解决了基于算术编码的隐写构造面临的安全性问题。该隐写构造主要在基于算术编码的隐写构造上做了两项修改：第一，与基于算术编码的隐写构造要进行一系列缩小区间操作不同，Meteor 在每一时间步都是从 $[0, 1)$ 采样；第二，每一时间步执行消息嵌入算法之前，先对消息进行重新加密，以防止随机性重用问题。因为 Meteor 与基于算术编码的隐写构造很相似，所以我们也将其归为基于算术编码的隐写构造。

然而，由于第一项修改，Meteor 在容量上比基于算术编码的隐写构造低了很多。此外，在 Meteor 作者提供的实现代码①中，原始分布 P_{c} 经历了 cutoff-rescale-round-remove-add 这一系列操作才成为隐写采样的实际分布 P_{s}。这些操作给原始分布带来了严重破坏，因此无法达到预期的安全性，图 2.3 给出了一个例子。

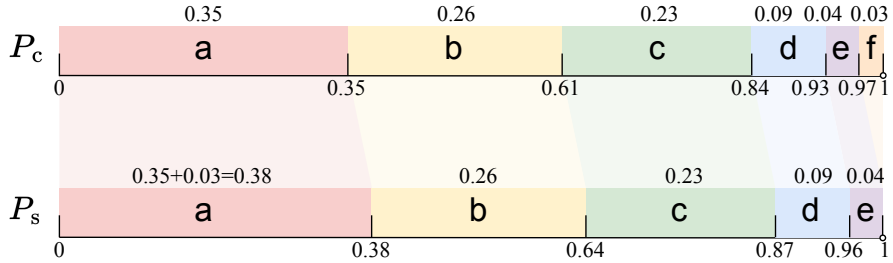


图 2.3　Meteor 对原始分布的破坏

2.6.2　基于样本空间分组的隐写构造

2021 年，Zhang 等人[39]提出了一种基于基于样本空间分组的隐写构造，取名为 ADG（adaptive dynamic grouping）。他们具体是这样做的：

1. 将所有符号按概率均匀地划分成 2^r 个分组，使得每个分组中符号的概率之和约等于 $1/2^r$，并编号为 $0, 1, \cdots, 2^r - 1$，相同分组中的所有符号都表示相同的 r 比特消息；
2. 发送方 Alice 读取 r 比特消息，将其转为十进制数 $d \in \{0, 1, \cdots, 2^r - 1\}$，并从编号为 d 的分组中随机采样出一个符号发送给接收方 Bob；
3. Bob 收到这个符号后，因为与 Alice 共享相同的模型和分组方法，可以获得完全相同的分组，从而提取出消息。

①https://gist.github.com/tusharjois/ec8603b711ff61e09167d8fef37c9b86

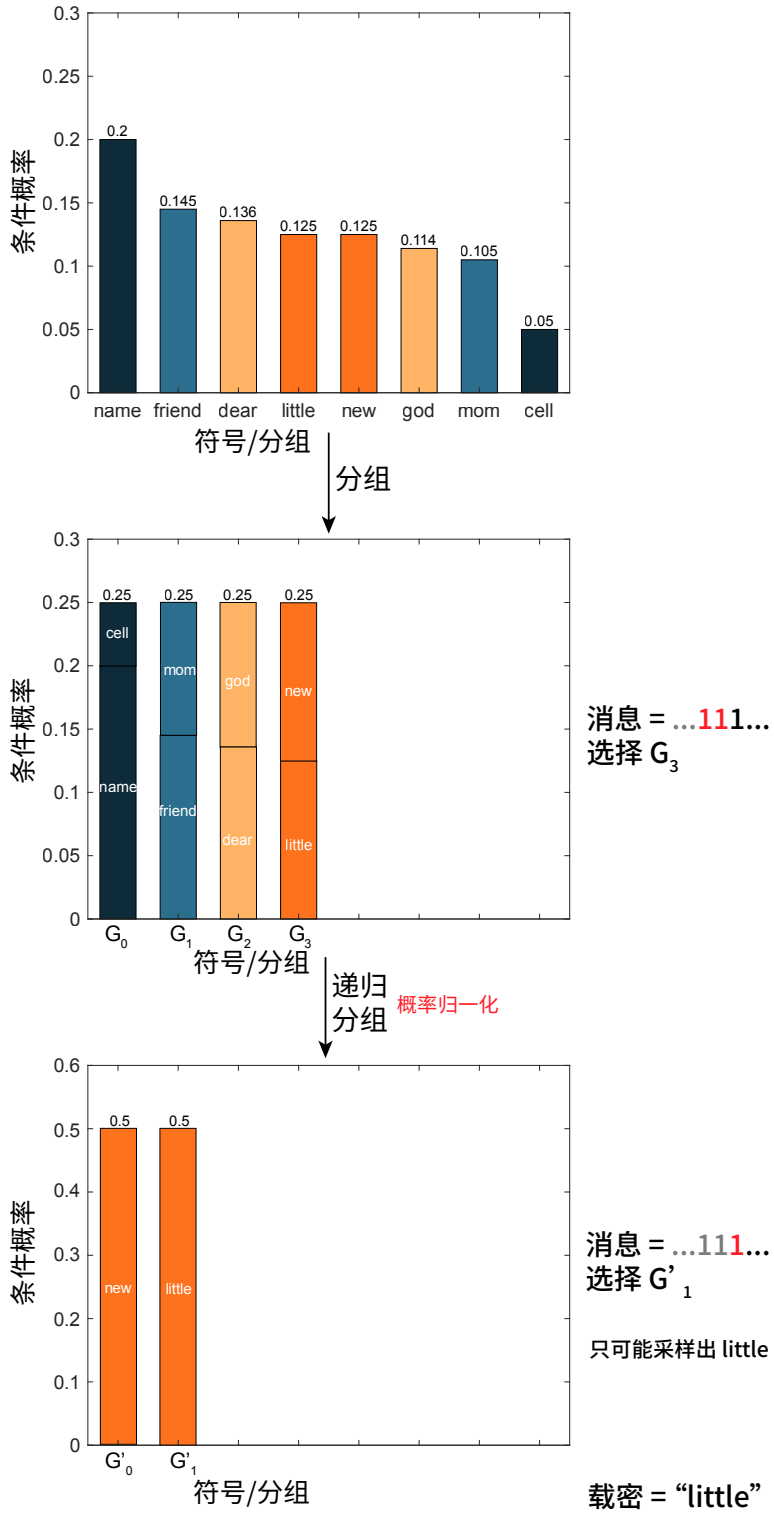


图 2.4　ADG 消息嵌入算法示意图

ADG 还通过递归分组的思想提升了嵌入容量，图 2.4 给出了一个例子。ADG 的安全性证明大致可归纳为：如果所有分组中符号的概率之和都是相等的，那么某个符号被 ADG 的消息嵌入算法采样的概率等于它被直接随机采样的概率。

然而，由于符号的概率是离散值，分组几乎不可能是完美平衡的。如果分组不是完美平衡的，ADG 隐写采样的实际分布是对原始分布进行修改后的结果，因此无法达到预期的安全性，图 2.5 给出了一个例子。

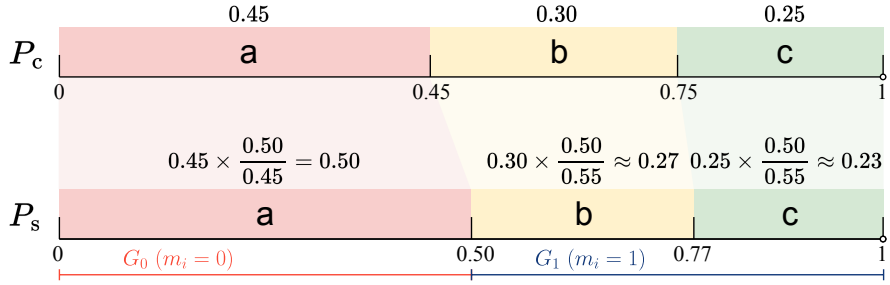


图 2.5　ADG 对原始分布的破坏

总结而言，现有的所有追求可证安全的高效隐写构造在实现过程中都存在安全性漏洞。如何设计一种在现实场景中仍能保持安全性的高效可证安全隐写构造，依然是一个亟待解决的难题。

2.7　蜜罐加密

如图 2.6 所示，一个蜜罐加密系统[3] $\mathsf{HE} = (\mathsf{HEnc}, \mathsf{HDec})$ 涉及两组算法，分别为分布转换编码器（DTE）的编码与解码算法 $(\mathsf{Encode}, \mathsf{Decode})$ 和传统加密算法及其对应的解密算法 $(\mathsf{CEnc}, \mathsf{CDec})$。

DTE 的编码算法 Encode 可以将明文转换为伪随机比特序列，解码算法 Decode 则可以将伪随机比特序列恢复为明文。文献 [3] 中将这种伪随机比特序列称为种子（seed）。为了方便后文描述，表 2.1 给出了蜜罐加密涉及的常用符号说明。如果对于任意 $M \in \mathcal{M}$，都有 $\Pr\left[\mathsf{Decode}\left(\mathsf{Encode}\left(M\right)\right) = M\right] = 1$ 成立，则称该 DTE 是正确的。

HEnc 和 HDec 的形式化分别如公式 (2.6) 和 (2.7) 所示。

$$C = \mathsf{HEnc}\left(K, M\right) = \mathsf{CEnc}\left(K, S\right) = \mathsf{CEnc}\left(K, \mathsf{Encode}\left(M\right)\right) \tag{2.6}$$

$$M = \mathsf{HDec}\left(K, C\right) = \mathsf{Decode}\left(S\right) = \mathsf{Decode}\left(\mathsf{CDec}\left(K, C\right)\right) \tag{2.7}$$

蜜罐加密系统的加密算法 HEnc 的具体步骤为：

1. 使用 DTE 的编码算法将明文 M 编码为种子 S；
2. 使用传统加密算法（如 AES）、密钥 K 对种子 S 进行加密，得到密文 C。

解密算法 HDec 则是 HEnc 的逆过程，具体步骤为：

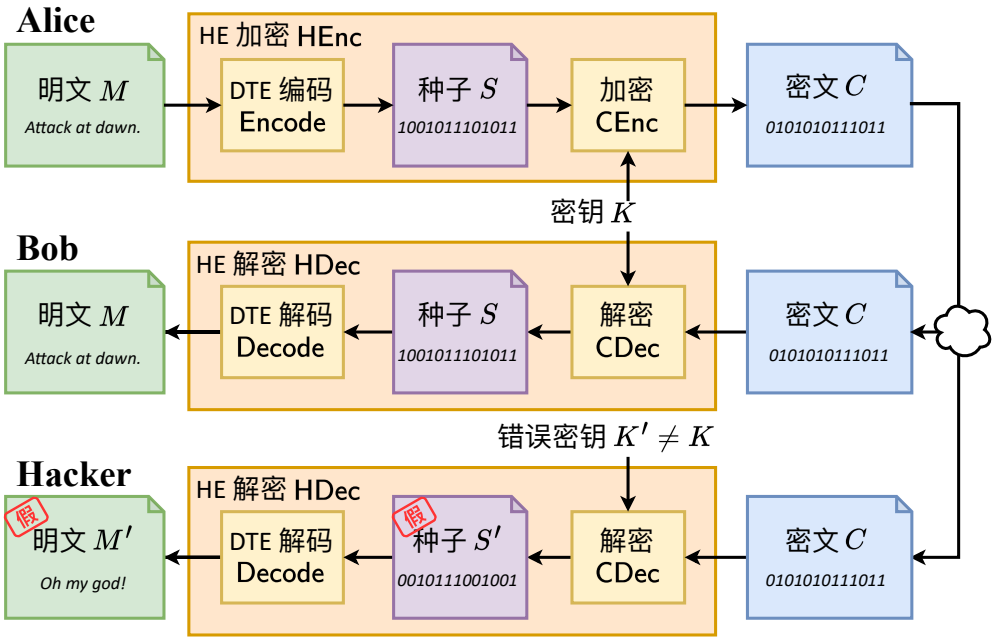


图 2.6　蜜罐加密系统示意图

表 2.1　蜜罐加密常用符号

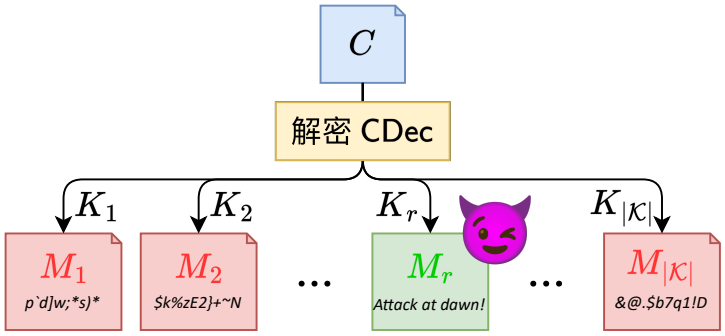
符号	符号说明
M	明文
S	种子
K	密钥
C	密文
$\mathcal{M}$	明文空间
$\mathcal{S}$	种子空间
$\mathcal{K}$	密钥空间
$\mathcal{C}$	密文空间
D	明文分布

1. 使用传统加密的解密算法、密钥 K 对密文 C 进行解密，获得种子 S；
2. 使用 DTE 的解码算法将种子 S 解码为明文 M。

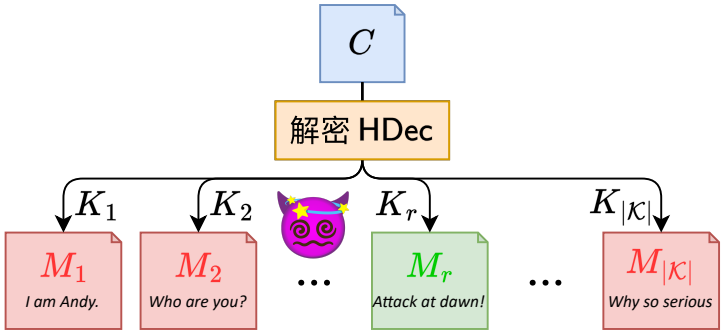
DTE 的引入，使得攻击者在尝试使用错误密钥 K' 解密密文 C 时，会得到看似合理的诱饵明文 M'，其相当于从明文数据分布 D 中做一次随机采样得到的结果。如果攻击者对密文 C 使用暴力破解攻击，通过枚举所有可能的密钥解密 C 获得大量候选明文，也难以定位出混迹于其中的唯一的真明文。图 2.7(a) 和 2.7(b) 分别展示了由传统加密得到的密文和由蜜罐加密得到的密文在面对暴力破解攻击时的不同表现。

我们专注于针对自然语言文本的蜜罐加密。在这个场景中，四个空间 $\{\mathcal{M}$、$\mathcal{S}$、$\mathcal{K}$、$\mathcal{C}\}$ 分别如下。

- 明文空间 $\mathcal{M}$：所有可能的自然语言文本组成的集合；
- 种子空间 $\mathcal{S}$：所有可能的比特序列组成的集合，这些序列的长度可能需要满足一定条件；



(a) 传统加密



(b) 蜜罐加密

图 2.7　传统加密和蜜罐加密在受到暴力破解攻击时的不同表现

- 密钥空间 $\mathcal{K}$：所有可能的固定长度（如 256 比特）的比特序列组成的集合；
- 密文空间 $\mathcal{C}$：与种子空间相同。

蜜罐加密的核心是 DTE，它负责明文与种子之间的相互转换。在 Juels 和 Ristenpart 的开创性论文[3]中，他们利用逆变换采样（inverse transform sampling）构建了一种通用的 DTE 编码方案。逆变换采样是一种从已知的概率分布中进行采样的方法。假设明文的概率分布 D 已按一定顺序 $\{M_1, M_2, \cdots, M_{|\mathcal{M}|}\}$ 排列，基于逆变换采样的编码和解码算法都需要先计算 D 对应的累积分布函数（CDF）：

$$F(M_i) = \begin{cases} 0, & \text{if } i = 0 \\ \sum_{j=1}^{i} \Pr\left[M_j\right], & \text{if } 1 \leqslant i \leqslant |\mathcal{M}| \end{cases} \tag{2.8}$$

若想要将明文 M_i 编码为种子 S，则从区间 $\left[F(M_{i-1}), F(M_i)\right)$ 中随机选取一个实数，其小数部分可作为 M_i 对应的种子 S。解码算法则是用到了 CDF 的逆函数，通过计算 $M_i = F^{-1}(S) = \min_j\left\{F(M_j) > S\right\}$ 可获得 S 对应的明文 M_i。因为这种 DTE 方案用到了 CDF，我们也可以将其称为基于 CDF 的 DTE 编码方案。

在具体实现时，$F(M_i)$ 和种子都是有限精度的。为了方便理解，图 2.8 给出了一个例子，其中明文空间规模为 $|\mathcal{M}| = 16$，精度为 $d_c = 6$。该方案根据明文分布为明文空间中的每个明文分配一个区间，每个区间中的元素数量与其对应的明文概率成正比。要编码一个明文，可以从它所对应的区间中随机选择一个元

素作为种子。

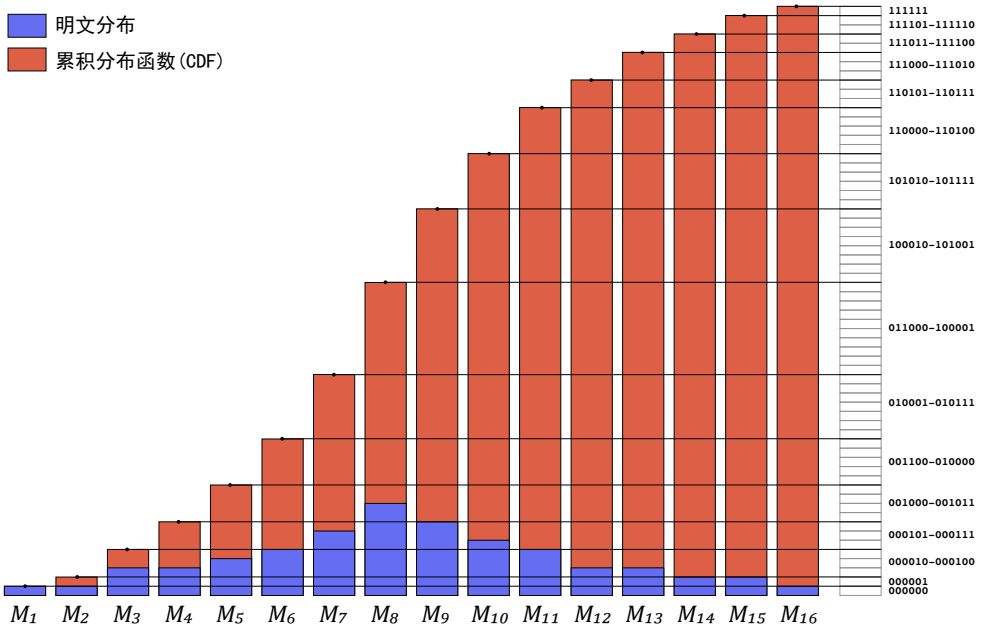


图 2.8　基于累积概率密度函数（CDF）的 DTE 编码方案

然而，这种基于 CDF 的 DTE 编码方案是一种定长编码方案，通常需要更长的编码来表示明文，这会带来更大的存储和传输开销。此外，从某种程度上来说，精度大小决定着 DTE 安全性的上限。因此，在精度的设置上需要慎重。如果精度太小，原始分布会受到比较严重的破坏；如果精度太大，不仅会导致加密和解密的时间增加，还会带来较大的存储和传输开销。因此，需要在安全性和性能之间寻求一个平衡点。

DTE 面临的另一个挑战是，如何对明文分布进行比较精准的建模。Juels 和 Ristenpart[3]仅讨论了对简单数据类型进行建模，例如信用卡号、个人识别码（PIN）以及银行卡安全码（CVV）。

2015 年，Huang 等人[113]提出了针对基因组数据的蜜罐加密方案 GenoGuard，通过构建一棵满三叉树来表示基因组数据，其中每个叶子节点代表一个完整序列，每个非叶子节点代表序列前缀。在这棵三叉树中，每个节点都对应一个长度与概率成正比的编码区间。在分布建模方面，GenoGuard 探索使用三种模型对基因组数据进行建模，包括通用的马尔可夫模型以及对基因组数据更具针对性的连续不平衡（linkage disequilibrium，LD）模型和重组（recombination）模型。Chatterjee 等人[43]提出了针对口令管理器（password vault）的蜜罐加密方案 NoCrack，尝试使用马尔可夫模型和概率上下文无关语法（probabilistic context-free grammars，PCFG）模型分别对口令分布进行建模。2016 年，Golla 等人[44]针对 Chatterjee 等人的方案[43]设计了基于 KL 散度的的攻击，并提出使用改进后的自适应马尔可夫模型对口令分布进行建模。2019 年，Cheng 等人[47]分析了 [43-44,113] 存在的

安全性漏洞，针对性地设计了分布差异攻击（distribution difference attack）和编码攻击（encoding attack），并给出了改进措施。

在对自然语言文本分布进行建模方面，2016 年，Kim 和 Yoon[45]通过字符级的 k 阶马尔可夫模型建模文本，该模型用于建模在之前 k 个字符的条件下，当前位置字符的概率分布。然而，马尔可夫模型本身存在以下缺陷[33]：

1. 马尔可夫模型并没有考虑之前所有字符，而是只考虑了其中的最后几个字符，这引入了近似误差。这种近似限制了它对语言的建模能力，尤其是在处理长期依赖关系方面。
2. 马尔可夫模型仅仅使用频率来近似概率，这可能导致其预测的概率分布非常稀疏，泛化性能差。
3. 马尔可夫模型的数据库的存储开销随着阶数 k 呈指数增长，即使 k 不是很大，这种成本也是难以接受的。

2017 年，Beunardeau 等人[46]指出传统的 PCFG 模型对文本的建模能力有限，因为其只考虑句法结构，而未考虑语义相关性，导致生成的假明文可能毫无意义。然后，他们启发式地设计了语料库引用（corpus quotation）模型：提前准备一个语料库，为这个语料库的所有子字符串（substring）分配概率，每个子字符串的概率仅与其包含的单词数量有关。然而，该模型既未考虑语义相关性，也未考虑句法正确性。

总结来说，现有的 DTE 方案通常采用传统的统计模型对明文分布进行建模，并使用基于 CDF 的编码方案将明文转换为种子。在分布建模方面，这些传统的统计模型都基于极度简化的假设，导致它们对数据的底层结构和分布特征的捕捉能力非常有限。而且，它们在应对未见过的数据时的适应能力和预测能力较弱。因此，这些模型并不适用于建模自然语言文本这种具有复杂分布特性的数据类型。在编码方案方面，基于 CDF 的编码方案往往需要更长的编码来表示明文信息。而且，在设置精度时需要谨慎：如果精度太小，原始分布会受到比较严重的破坏；如果精度太大，加解密时间将延长，同时存储和传输开销会相应增加。

2.8　本章小结

本章首先介绍了隐写的相关理论与方法，包括隐写模型、传统的修改式隐写方法、隐写安全性定义、经典可证安全隐写构造、深度生成模型及其对可证安全隐写的意义、高效可证安全隐写构造，然后介绍了蜜罐加密的相关理论与方法，为后续章节奠定了基础。

第 3 章　基于分布副本的生成式可证安全隐写构造

本章将提出一种基于分布副本的生成式可证安全隐写构造。其中第 3.1 节对现有的可证安全隐写构造在实践中面临的问题进行总结性回顾；第 3.2 节介绍文本生成的一般过程和随机采样的一种实现；第 3.3 节提出基于分布副本的可证安全隐写构造的基础版本；第 3.4 节通过递归分组的思想提升了这种隐写构造的嵌入率；第 3.5 节给出了这种隐写构造的部署场景举例；第 3.6 节从安全性、容量效率、时间效率三个方面展开实验与分析；第 3.7 节讨论这种隐写构造所需的带外开销；第 3.8 节对本章内容进行总结。

3.1　现有的可证安全隐写构造的问题

在第 2.4 节和第 2.6 节中，我们已经对现有的可证安全隐写构造进行了较为详细的介绍与分析。概括而言，现有的可证安全隐写构造大致可以分为三类：基于拒绝采样的隐写构造、基于算术编码的隐写构造和基于样本空间分组的隐写构造。第一类构造针对黑盒模型，使用样本的函数值来表达消息，虽然原理简单直接，但通常具有较低的嵌入率和较高的时间复杂度。第二和第三类构造针对白盒模型，使用样本的索引值来表达消息，尽管能达到较高的嵌入率和较快的速度，但其具体实现过程往往难以满足理论假设，因此无法达到预期的安全性。

3.2　文本生成和随机采样

在下文中，我们将主要以文本生成任务为例介绍提出的隐写算法。目前，文本生成任务一般采用自回归模型，比如 GPT 系列模型[36,96,101-102]。这类模型可以预测给定上文（context）条件下，下一个 token 的概率分布。其中，token 是自然语言处理领域的一个专业术语，一般翻译为“词元”，指的是文本中的最小语义单位，可能是单词、子词（subword）、标点符号等。所有候选词元组成的集合称为 vocabulary，一般翻译为“词表”。通过多次重复这两个过程，我们可以最终获得一段文本：

1. 使用模型预测上文条件下，下一个词元的条件概率分布；
2. 从分布中采样出一个词元作为下一个词元，附加到上文的末尾。

为了使得生成文本尽可能符合自然语言文本的统计特征，通常会采用随机采样（random sampling）策略，可以通过以下过程实现：

1. 使用伪随机数生成器（pseudo-random number generator，PRNG）产生一个伪随机数 $r^{(t)} \in [0, 1)$，PRNG 的定义及使用方法将在第 3.2.1 小节中介绍；

2. 设生成模型预测出的下一个词元的条件概率分布为 $\mathcal{P}^{(t)} = \Pr\left[x_t \mid x_{<t}\right] = \Pr\left[x_t \mid x_1, x_2, \cdots, x_{t-1}\right]$，根据 $\mathcal{P}^{(t)}$ 为词表 V 中的每个概率不为 0 的词元分配 $[0,1)$ 区间内互不相交的左闭右开区间；
3. 选择 $r^{(t)}$ 落入的区间对应的词元作为下一个词元 x_t。

总体上，文本生成任务的一般过程如图 3.1 所示。其中，种子是用于初始化 PRNG 的整数或向量。

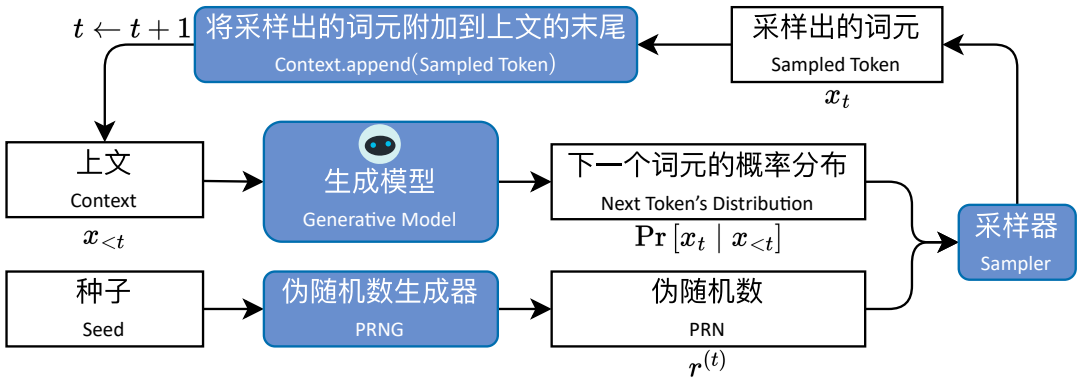


图 3.1　文本生成任务的一般过程

3.2.1　伪随机数生成器

定义 3.1　(计算安全的伪随机数生成器[114-115]) 设 G 是一个确定性多项式时间算法，若对于任意 λ 比特输入 $s \in \{0,1\}^{\lambda}$，算法 G 输出一个长度为 $\ell(\lambda)$ 的比特序列，其中 ℓ 是一个多项式。如果 G 满足以下两个条件，则称 G 是一个计算安全的伪随机数生成器：

1. **扩展性（Expansion）**：对于所有 λ，都有 $\ell(\lambda) > \lambda$。
2. **伪随机性（Pseudo-randomness）**：对于所有概率多项式时间的区分器 $\mathcal{A}$，存在一个可忽略函数 negl 使得以下不等式成立

$$\left|\Pr\left[\mathcal{A}\left(G(s)\right) = 1\right] - \Pr\left[\mathcal{A}\left(s'\right) = 1\right]\right| \leqslant \mathrm{negl}(\lambda) \tag{3.1}$$

其中，种子 s 是从 $\{0,1\}^{\lambda}$ 中均匀随机选择的，s' 是从 $\{0,1\}^{\ell(\lambda)}$ 中均匀随机选择的，两者均为真随机比特序列。

如何使用 PRNG 产生 $[0,1)$ 区间内的小数？假设 PRNG 输出的比特序列 $G(s) = b_0, b_1, \cdots, b_i, \cdots, b_{\ell(\lambda)-1}$，其中 $b_i \in \{0,1\}$。在每一时间步 t，从 $G(s)$ 中依次取出连续的 h 比特 $b_{ht}, b_{ht+1}, \cdots, b_{ht+h-1}$，通过计算

$$r^{(t)} = \frac{\sum_{i=0}^{h-1} b_{ht+i} \times 2^i}{2^h} \tag{3.2}$$

可以获得 $[0,1)$ 区间内的伪随机数 $r^{(t)} \in [0,1)$，h 可称为该伪随机数的精度。

3.3　基于分布副本的可证安全隐写基础构造

我们观察到，对于一个可选词元数大于 1 的概率分布，其在 $[0,1)$ 上的区间分配方案并不是唯一的。举例而言，假设有 a 和 b 两个可选词元，概率分别为 0.4 和 0.6，可以有如表 3.1 所示的区间分配方案。我们将一个概率分布的多种区间分配方案称为这个分布的副本，即**分布副本**（distribution copies）。

表 3.1　区间分配方案举例

方案索引	a	b
0	[0.0, 0.4)	[0.4, 1.0)
1	[0.6, 1.0)	[0.0, 0.6)

基于这个观察，在使用显式生成模型进行生成任务的过程中，我们可以为生成模型预测的概率分布创建多个分布副本，并使用分布副本的索引值来表达消息。为方便描述，我们将索引值为 i 的分布副本简记为分布副本 i（i 从 0 开始编号）。接下来介绍这种基于分布副本的隐写算法的消息嵌入和提取过程。

消息嵌入：在每一时间步 t，发送方 Alice 在生成过程中嵌入消息的过程为：

1. 生成模型 $\mathcal{G}$ 在上文 $x_{<t}$ 的条件下预测下一个词元的概率分布 $\mathcal{P}^{(t)}$；
2. 若要嵌入 n_t 比特消息，则为 $\mathcal{P}^{(t)}$ 创建 2^{n_t} 个分布副本，赋予索引值 $\{0,1,\cdots,2^{n_t}-1\}$；
3. 读取 n_t 比特消息，将该消息片段转为十进制数 i；
4. 消耗一个由 PRNG 产生的伪随机数 $r^{(t)}$，选择 $r^{(t)}$ 在分布副本 i 中落入的区间对应的词元作为下一个词元 x_t；
5. 将 x_t 附加在 $x_{<t}$ 尾部。

消息提取：相应地，在每一时间步，接收方 Bob 可以通过判断当前词元 x_t 是从哪个分布副本中采样获得的来提取消息。具体地，Bob 在每一时间步 t 提取消息的过程为：

1. 生成模型 $\mathcal{G}$ 在上文 $x_{<t}$ 的条件下预测下一个词元的概率分布 $\mathcal{P}^{(t)}$；
2. 若要提取 n_t 比特消息，则为 $\mathcal{P}^{(t)}$ 创建 2^{n_t} 个分布副本，赋予索引值 $\{0,1,\cdots,2^{n_t}-1\}$；
3. 消耗一个由 PRNG 产生的伪随机数 $r^{(t)}$，假设只有在分布副本 i 中，$r^{(t)}$ 落入词元 x_t 对应的区间，那么可以唯一地确定这个词元是从分布副本 i 中采样获得；
4. 将 i 转为 n_t 比特序列，这就是消息的一个片段。

只要通信双方共享相同的设置，接收方就能与发送方同步每一时间步的状态，并正确提取消息。具体来说，共享上文和生成模型意味着能获得相同的条件概率分布，从而创建相同的分布副本；共享 PRNG 和种子意味着能获得相同的伪随机数；共享分布副本和伪随机数就可以正确提取消息。

然而，这种隐写构造需要解决以下问题：

1. 如何创建多个分布副本？
2. 每一时间步嵌入消息长度 n_t 应该如何确定？
3. 若 $r^{(t)}$ 在不止一个分布副本中都落入某个词元对应的区间，应该如何嵌入和提取消息？

针对第 1 个问题，我们采用一种简单的方法：循环移位（rotate）。首先，约定一种初始区间分配方案作为分布副本 0（如将词表 V 中所有概率不为 0 的词元按字典序排列在 $[0, 1)$ 上）；然后，将分布副本 0 中的所有词元对应的区间向左循环移位 $i \times 2^{-n_t}$ 得到分布副本 i，其中，n_t 为这一轮采样要嵌入的消息长度，以比特为单位。如图 3.2 所示，所谓向左循环移位，就是所有区间的左端点和右端点都减去移位距离，并将小于 0 的部分置于最右侧。在此过程中，所有区间的开闭性保持不变。

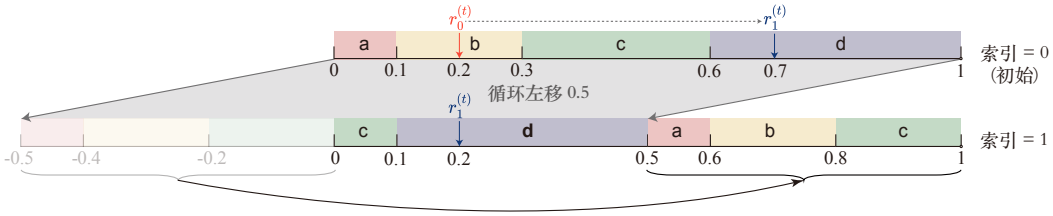


图 3.2　基于分布副本的隐写方法嵌入 1 比特消息实例

为了方便理解，这里给出一个实例。假设有 a、b、c、d 四种词元，概率分别为 0.1、0.2、0.3、0.4。不妨以 a-b-c-d 顺序排列的分配方案作为初始区间分配方案，即分布副本 0，如图 3.2 中的第一行所示。假设 Alice 想要嵌入 1 比特消息，她需要创建 $2^1 = 2$ 个分布副本，对应的移位步长为 $1/2 = 0.5$。她将分布副本 0 向左循环移位 0.5，得到分布副本 1，如图 3.2 中的第二行所示。假设这一时间步 Alice 要嵌入 $m^{(t)} = 1$，消耗的伪随机数 $r^{(t)} = 0.2$，因为 $r^{(t)}$ 在分布副本 1 中落入 d 对应的区间，所以她采样出 d 发送给 Bob。如果 Bob 与 Alice 同步这一时间步的状态，就可以创建相同的分布副本。根据相同的伪随机数 $r^{(t)} = 0.2$ 和收到的 d，Bob 能够唯一地确定这个 d 是从分布副本 1 中采样获得的，因此他提取出消息为 1。

针对第 2 和第 3 个问题，我们将在下文中给出唯一提取条件，引入争议区间的概念，并计算嵌入容量的期望值。

3.3.1　唯一提取条件与争议区间

不难推出，当消耗的伪随机数在所有分布副本中落入的区间对应的词元互不相同时，接收方可以唯一地提取消息。沿用上文实例中的概率分布，图 3.2 展示了嵌入 1 比特消息时分布副本的构造情况，此时，无论消耗的伪随机数是多

少，都可以唯一地提取消息。然而，当发送方试图嵌入 2 比特消息时，需要创建 $2^2 = 4$ 个分布副本，对应的移位步长为 1/4 = 0.25。创建的 4 个分布副本如图 3.3 所示，会出现消耗的伪随机数 $r^{(t)}$ 在多个分布副本中落入的区间对应的词元相同的情况，即不同分布副本中分配给相同词元的区间具有交集，且消耗的伪随机数正好落入交集中，此时接收方无法唯一提取消息，我们将这些交集称为争议区间（disputed ranges），如图 3.3 中灰色遮罩覆盖部分所示[①]。试图嵌入 2 比特消息时（图 3.3），如果消耗的伪随机数不幸落入争议区间中，则无法嵌入，只能回退到嵌入 1 比特消息的情况（图 3.2）。注意，争议区间的存在并不会影响消息提取的正确性，因为接收方与发送方同步相同的状态，可以计算出争议区间并判断消耗的伪随机数是否落入争议区间。因此，他知道发送方嵌入了多少比特消息，可以创建与发送方相同的分布副本来提取消息。

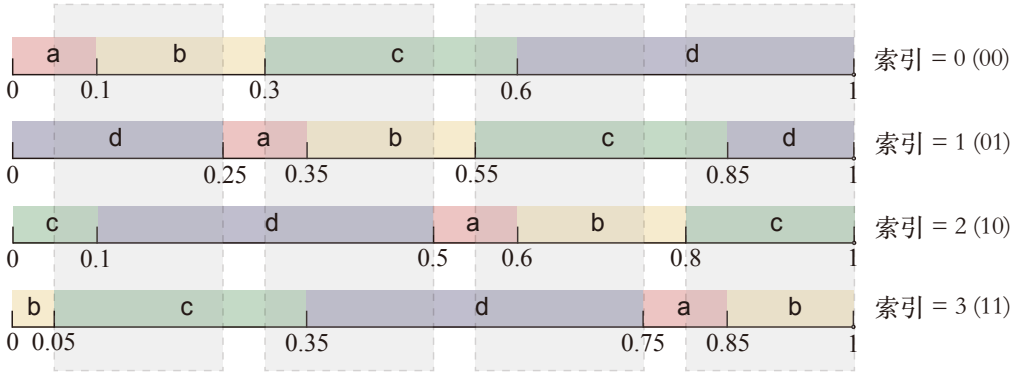


图 3.3　基于分布副本的隐写方法嵌入 2 比特消息实例

3.3.2　嵌入率

嵌入率是指平均每生成一个词元可嵌入的消息长度，本文中我们使用符号 β 来表示。类似地，我们可以将某一时间步 t 生成词元 x_t 能嵌入的消息长度定义为在这一时间步的瞬时嵌入率 $\beta^{(t)} \in \mathbb{N}$，对应的移位步长为 $2^{-\beta^{(t)}}$。设这一时间步模型预测的分布中，概率的最大值为 $p_{\max}^{(t)}$。令 $\mathrm{B}^{(t)} \in \mathbb{R}_{\geqslant 0}$ 为 $\beta^{(t)}$ 的上界，要使 $\beta^{(t)}$ 尽可能大，那么 $\beta^{(t)} = \lfloor \mathrm{B}^{(t)} \rfloor$。由分布副本创建方法和唯一提取条件，可得

$$\frac{p_{\max}^{(t)}}{2} < 2^{-\mathrm{B}^{(t)}} \leqslant p_{\max}^{(t)} \tag{3.3}$$

又 $\beta^{(t)} = \lfloor \mathrm{B}^{(t)} \rfloor$，所以有

$$\left\lfloor \log_2 \frac{1}{p_{\max}^{(t)}} \right\rfloor \leqslant \beta^{(t)} \leqslant \left\lceil \log_2 \frac{1}{p_{\max}^{(t)}} \right\rceil \tag{3.4}$$

令 $\alpha = \log_2 \left(1/p_{\max}^{(t)} \right)$，若 $\alpha \in \mathbb{N}$，则 $\beta^{(t)} = \alpha$；否则 $\beta^{(t)}$ 为 $\lfloor \alpha \rfloor$ 或 $\lceil \alpha \rceil$，具体是哪个值取决于消耗的伪随机数是否落入争议区间中。因此，可以推断出嵌入率

[①]例如，$[0.1, 0.25)$ 之所以是争议区间，是因为这个区间在分布副本 2 和分布副本 3 上对应的词元都为 d。

的期望值是所有时间步概率分布的最小熵[①]的平均值，即有：

$$\beta \approx \frac{1}{T}\sum_{t=1}^{T}\log_2\frac{1}{p_{\max}^{(t)}} \tag{3.5}$$

其中 T 表示生成词元数量。

3.3.3　一种等价的实现

如图 3.2 所示，将所有区间向左循环移位 d 等价于将消耗的伪随机数 $r^{(t)}$ 向右循环移位 d。在下文中，我们统一通过将消耗的伪随机数向右循环移位来创建分布副本（分布副本 0 可视为通过将消耗的伪随机数向右循环移位 0 获得）。为了叙述方便，我们在算法 3.1 中定义算法 $\mathsf{rotate}(a,d,e)$，用于计算在区间 $[0,e)$ 中将 a 循环移位 d 得到的值。

算法 3.1　$\mathsf{rotate}(a,d,e)$: 在 $[0,e)$ 区间中将 a 循环移位 d（$|d|<e$）

Input: 原始值 a，移位距离 d，区间右端点（不包含）e
Output: 移位后的值 b
/* $0<d<e$ 时表示向右循环移位，$-e<d<0$ 时表示向左循环移位 */
$b \leftarrow a+d$
if $b \geqslant e$ **then**
　$b \leftarrow b-e$
end
if $b<0$ **then**
　$b \leftarrow b+e$
end
return b

3.3.4　安全性证明

要证明这种基于分布副本的隐写构造产生的载密与载体的分布在计算上不可区分，鉴于载密和载体的差别仅在于采样过程中使用的随机变量，那么只需证明这两种采样过程使用的随机变量的分布在计算上不可区分。

在任意时间步 t，假设要嵌入长度为 n_t 比特的消息 $m^{(t)}$。依据概率论的基本公理，消息 $m^{(t)}$ 的所有可能取值的概率之和等于 1，即

$$\sum_{m\in\{0,1\}^{n_t}}\Pr\left[m^{(t)}=m\right]=1 \tag{3.6}$$

隐写者使用计算安全的 PRNG 生成伪随机比特序列 $b_{ht},b_{ht+1},\cdots,b_{ht+h-1}$，通过公式 (3.2) 将其转化为 $[0,1)$ 区间内精度为 h（$h\geqslant n_t$）的伪随机小数 $r^{(t)}$。那么对于所有 $a\in\left\{i\times 2^{-h}\right\}_{i=0,\cdots,2^h-1}$，存在一个关于安全参数 λ 可忽略的函数 ε 使得以下不等式成立：

$$\left|\Pr\left[r^{(t)}=a\right]-2^{-h}\right|\leqslant\varepsilon(\lambda) \tag{3.7}$$

[①]一个概率分布的最小熵等于 $\log\left(1/p_{\max}\right)$，其中 $p_{\max}$ 表示该分布中概率的最大值。

所以

$$2^{-h}-\varepsilon(\lambda)\leqslant \Pr\left[r^{(t)}=a\right]\leqslant 2^{-h}+\varepsilon(\lambda) \tag{3.8}$$

实际上，可以将 $r^{(t)}$ 视为正常采样使用的随机变量。设 $r'^{(t)}$ 为隐写采样使用的随机变量，$\mathsf{dcm}(m)$ 为消息 m 对应的十进制数，$z(a,m)=\mathsf{rotate}\left(a,-\mathsf{dcm}(m)\times 2^{-n_t},1\right)$ 表示 a 在 $[0,1)$ 区间内向左循环移位 $\mathsf{dcm}(m)\times 2^{-n_t}$ 得到的值，那么对于 $a\in\left\{i\times 2^{-h}\right\}_{i=0,\cdots,2^h-1}$，$m\in\{0,1\}^{n_t}$，有

$$\begin{aligned}
&\sum_a\left|\Pr\left[r'^{(t)}=a\right]-\Pr\left[r^{(t)}=a\right]\right| \\
=&\sum_a\left|\sum_m\Pr\left[m^{(t)}=m\right]\times\Pr\left[r^{(t)}=z(a,m)\right]-\Pr\left[r^{(t)}=a\right]\right| \\
=&\sum_a\left|\sum_m\Pr\left[m^{(t)}=m\right]\times\left[\Pr\left[r^{(t)}=z(a,m)\right]-\Pr\left[r^{(t)}=a\right]\right]\right| \\
\leqslant&\sum_m\Pr\left[m^{(t)}=m\right]\sum_a\left|\Pr\left[r^{(t)}=z(a,m)\right]-\Pr\left[r^{(t)}=a\right]\right| \\
\leqslant&\sum_m\Pr\left[m^{(t)}=m\right]\sum_a\underbrace{\left|\left[2^{-h}+\varepsilon(\lambda)\right]-\left[2^{-h}-\varepsilon(\lambda)\right]\right|}_{\text{由式 (3.8)}} \\
=&\sum_m\Pr\left[m^{(t)}=m\right]\cdot 2^h\cdot 2\varepsilon(\lambda) \\
=&\,2^{h+1}\cdot\varepsilon(\lambda)
\end{aligned} \tag{3.9}$$

因为 h 是与 λ 无关的常数，那么存在关于 λ 的可忽略函数 $\varepsilon'(\lambda)=2^{h+1}\cdot\varepsilon(\lambda)$ 使得以下不等式成立：

$$\sum_{a\in\left\{i\times 2^{-h}\right\}_{i=0,\cdots,2^h-1}}\left|\Pr\left[r'^{(t)}=a\right]-\Pr\left[r^{(t)}=a\right]\right|\leqslant\varepsilon'(\lambda) \tag{3.10}$$

鉴于载密和载体的差别仅在于采样过程中使用的随机变量，而隐写采样使用的随机变量 $r'^{(t)}$ 和正常采样使用的随机变量 $r^{(t)}$ 的分布是计算上不可区分的，因此，可以推导出载密和载体分布在计算上不可区分。

3.4　通过递归分组思想提升嵌入率

虽然第 3.3 节中介绍的隐写构造可以保持分布，从而实现可证安全，但其嵌入率仅仅约为所有时间步的概率分布的最小熵的平均值，与其理论极限（所有时间步的概率分布的熵的平均值）仍有一定距离。因此，我们将进一步研究如何提高嵌入率。

考虑一个简单的例子，假设有 a、b、c 三种词元，概率分别为 0.5、0.25、0.25。如果直接使用第 3.3 节给出的隐写构造（下文中简记为**基础构造**），那么只能嵌入 1 比特消息（因为如果试图嵌入 2 比特消息，则整个 $[0,1)$ 都是争议区间）。我们的想法是，通过递归分组的思想来提高嵌入率：首先将 b 和 c 放入一个分组 G 中，这样，一开始 a 和 G 的概率分别为 0.5 和 0.5，使用基础构造一定可以嵌入 1 比特消息，如果第一次选择 G，因为其中的 b 和 c 在归一化后的概率分别为 0.5 和 0.5，那么可以再次使用基础构造额外嵌入 1 比特消息。这样，嵌入消息长度的期望是 1.5 比特。提升后的隐写构造的消息嵌入算法在每一时间步的过程可归纳为两个步骤：通过递归分组构建一棵二叉树，然后在若干次子节点选择的过程（即从根节点走到某个叶子节点的过程）中嵌入消息。实际上，这相当于将原先的一轮多元分布采样分解为多轮二元分布采样，每次采样都有一定概率能嵌入 1 比特消息。

如何创建二叉树，可以使得隐写嵌入率尽可能大？我们采取的贪心策略是尽可能减小争议区间的总长度：在每个非叶子节点上，争议区间总长度越小，嵌入 1 比特消息的可能性就越大。我们将词表 V 中的第 j 个词元记作 V_j，将 $\mathcal{G}$ 在时间步 t 预测的 V_j 的条件概率记作 $p_j^{(t)}$。假设一个非叶子节点的两个子节点分别为 G_{left} 和 G_{right}，易推出此次子节点选择的争议区间大小为

$$\left| \sum_{V_j \in G_{\text{left}}} p_j^{(t)} - \sum_{V_j \in G_{\text{right}}} p_j^{(t)} \right| \tag{3.11}$$

由于 Huffman 树能够有效地构建一系列尽可能平衡的二元分布，我们因此决定采用 Huffman 树。

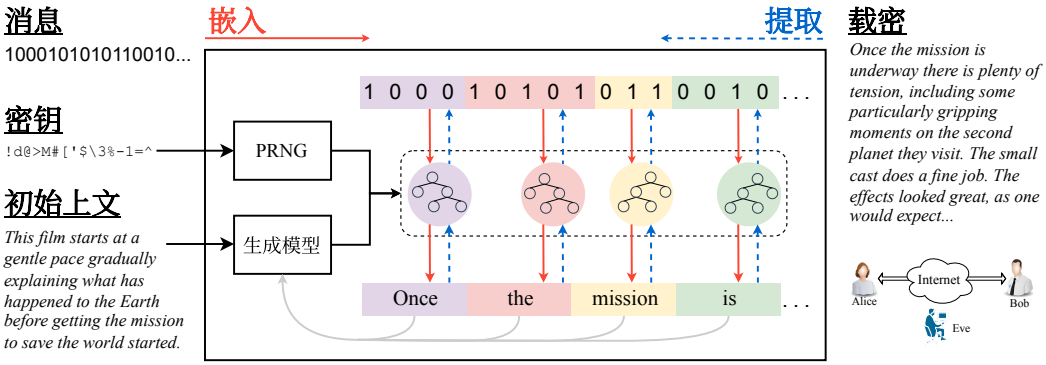


图 3.4　Discop 示意图

我们取分布副本英文（**dis**tribution **cop**ies）中的部分字母，将提升后的隐写构造称为 Discop。其示意图如图 3.4 所示。在下文中，我们将首先介绍 Discop 的消息嵌入算法与消息提取算法都涉及的两个子过程——创建 Huffman 树和子节点选择，然后介绍 Discop 的消息嵌入算法和消息提取算法的总体过程。

创建 Huffman 树：在每一时间步 t，根据 $\mathcal{G}$ 预测的条件概率分布 $\mathcal{P}^{(t)}$ 创建一

棵 Huffman 树，它的每个叶子节点对应一个词元，每个非叶子节点对应一个包含多个词元的分组。每一时间步开始时，我们在树的根节点处。创建 Huffman 树的线性复杂度算法如算法 3.2 所示。

算法 3.2　create_tree$\left(\mathcal{P}^{(t)}\right)$：创建 Huffman 树的线性复杂度算法

```
Input: 概率分布 P^(t)
Output: 创建的 Huffman 树的根结点 root
indices, probs ← P^(t)
nodes ← [ ]
n ← probs.size()
for i ← 0 to n − 1 do
    node ← Node(indices[i], probs[i], ∅, ∅)
    nodes.append(node)
end
q1 ← queue(nodes)
q2 ← queue()
subroutine GetMin()
    nonlocal q1, q2
    if q1.size() > 0 and q2.size() > 0 and
       q1.front().prob < q2.front().prob then
        item ← q1.front()
        q1.pop_front()
    else if q1.size() = 0 then
        item ← q2.front()
        q2.pop_front()
    else if q2.size() = 0 then
        item ← q1.front()
        q1.pop_front()
    else
        item ← q2.front()
        q2.pop_front()
    return item
while q1.size() + q2.size() > 1 do
    left ← GetMin()
    right ← GetMin()
    prob ← left.prob + right.prob
    q2.push_back(Node(∅, prob, left, right)
end
if q2.size() > 0 then
    root ← q2.front()
else
    root ← q1.front()
end
return root
```

子节点选择：树的每个非叶子节点都有两个子节点供选择，这使得我们有一定概率能嵌入 1 比特消息。假设我们希望嵌入 $m_i \in \{0,1\}$，先使用 PRNG 产生一个伪随机数 $r \in \left[0, \text{node.prob}\right)$，其中 node.prob 是当前节点包含的所有词元的概率之和。然后根据 $\text{rotate}\left(r, m_i \times 0.5 \times \text{node.prob}, \text{node.prob}\right)$ 的值落入哪个子节点对应的区间来决定选择哪个子节点。具体地，如果 $m_i = 0$，那么走到 r 落入的区

间对应的子节点；如果 $m_i = 1$，那么走到 $\mathrm{rotate}\big(r, 0.5 \times \mathrm{node.prob}, \mathrm{node.prob}\big)$ 落入的区间对应的子节点。但是，如果 r 与其循环移位后的值落入相同子节点对应的区间（即 r 落入争议区间），则我们只能走到这个子节点，且此次子节点选择无法嵌入消息，m_i 留到之后的时间步嵌入。

算法 3.3　Discop 的消息嵌入算法的主循环

Input: 初始上文 C，伪随机数生成器 PRNG，生成模型 $\mathcal{G}$，种子 K，待嵌入消息 **m**
Output: 载密 S

$S \leftarrow$ " "
PRNG.set_seed(K)
while not the end of **m do**
    $\mathcal{P}^{(t)} \leftarrow \mathcal{G}(C)$　　// 在上文条件下下一个词元的概率分布
    $n_m, w \leftarrow \mathrm{sample}\big(\mathcal{P}^{(t)}, \mathbf{m}, \mathrm{PRNG}\big)$　　// 本次嵌入的消息长度、采样出的词元
    $\mathbf{m} \leftarrow \mathbf{m}[n_m :]$
    $C \leftarrow C \| w$
    $S \leftarrow S \| w$
end
return S

消息嵌入：Discop 的消息嵌入算法的主循环如算法 3.3 所示，单个时间步的采样过程如算法 3.4 所示。假设我们想要嵌入消息 **m**，在每一时间步 t，先根据 $\mathcal{P}^{(t)}$ 创建一棵 Huffman 树。我们从它的根节点出发，重复子节点选择过程，直到当前节点已经是叶子节点，将其对应的词元作为下一个词元附加到上文 C 的末尾。重复上述过程，直到消息被完全嵌入，得到载密 S。

消息提取：提取消息的顺序与嵌入消息的顺序相同，即从第一个到最后一个遍历所有词元。接收方只要与发送方共享相同的设置，就可以在每一时间步同步所有状态，创建相同的 Huffman 树，然后根据载密 S 确定从树的根节点到叶节点的路径，通过递归地判断当前词元属于哪个子节点来提取消息。最后，他将提取出的所有消息片段连接在一起，获得完整消息。

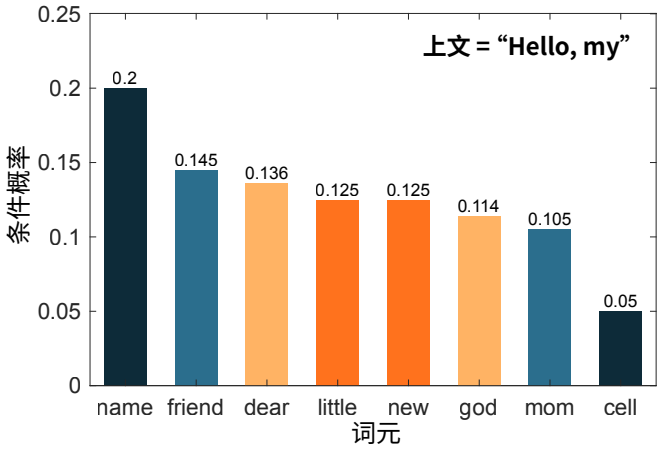


图 3.5　下一个词元的条件概率分布例子

为了方便理解，这里给出消息嵌入算法的一个实例。假设我们想要嵌入的消息为 $[\cdots, 1, 0, 1, \cdots]$，上文 C 为 "Hello, my"，生成模型 $\mathcal{G}$ 预测的上文 C 条件下

算法 3.4　sample$(\mathcal{P}^{(t)}, \mathbf{m}, \mathrm{PRNG})$：Discop 的消息嵌入算法在时间步 t 采样

Input: 概率分布 $\mathcal{P}^{(t)}$，待嵌入消息 $\mathbf{m}$，伪随机数生成器 PRNG
Output: 本次嵌入的消息长度 n_m，采样出的词元 w

struct {
    token, prob, left, right
} Node
node ← create_tree $(\mathcal{P}^{(t)})$　　// node 初始值为 Huffman 树的根结点
$n_m \leftarrow 0$
while not node.is_leaf() **do**
    $r \leftarrow$ PRNG.random $(0,$ node.prob$)$　　// $r \in [0,$ node.prob$)$
    $r_0 \leftarrow r$
    $r_1 \leftarrow$ rotate $(r, 0.5 \times$ node.prob, node.prob$)$
    separator ← node.left.prob
    next ← [ ]
    if $r_0 <$ separator **then**
        next[0] ← node.left
    else
        next[0] ← node.right
    end
    if $r_1 <$ separator **then**
        next[1] ← node.left
    else
        next[1] ← node.right
    end
    if next[0] ≠ next[1] **then**　　// 如果 r 在移位前后落入不同节点对应的区间
        $n_m \leftarrow n_m + 1$　　// 则可以嵌入 1 比特消息
    end
    node ← next $[\mathbf{m}[n_m]]$　　// 走到选中的子节点
end
$w \leftarrow$ node.token
return n_m, w

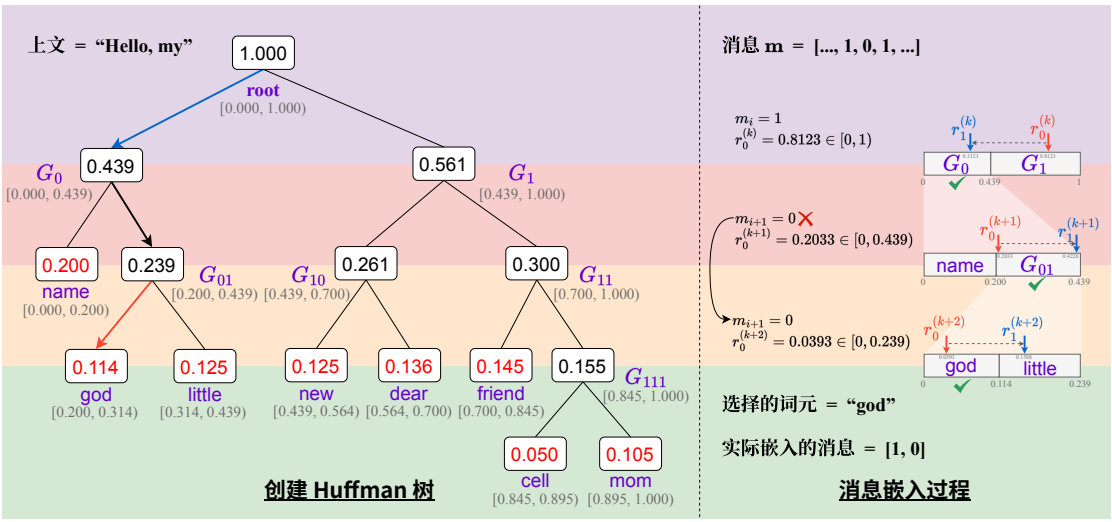


图 3.6　Discop 嵌入算法示例

下一个词元的概率分布如图 3.5 所示。图 3.6 则展示了这个分布对应的 Huffman 树和消息嵌入过程，具体如下：

1. 最初，我们在 Huffman 树的根节点处，这是一个非叶子节点，要做子节点选择。使用 PRNG 产生一个伪随机数 $r^{(k)} \in [0,1)$，假设为 $r^{(k)} = 0.8123$，将其在 $[0,1)$ 区间内向右循环移位 0 和 0.5 分别得到 $r_0^{(k)} = 0.8123$ 和 $r_1^{(k)} = 0.3123$，它们分别落入子节点 G_1 和 G_0 所对应的区间中。因为要嵌入 $m_i = 1$，所以我们走到 $r_1^{(k)}$ 对应的 G_0。
2. 现在，我们在 G_0，还是一个非叶子节点，要继续做子节点选择。使用 PRNG 产生一个伪随机数 $r^{(k+1)} \in [0, 0.439)$，假设为 $r^{(k+1)} = 0.2033$，将其向右循环移位 0 和 $0.5 \times 0.439 = 0.2195$ 分别得到 $r_0^{(k+1)} = 0.2033$ 和 $r_1^{(k+1)} = 0.4228$，不幸的是，它们都落入子节点 G_{01} 对应的区间，我们只能走到 G_{01}，且无法嵌入 $m_{i+1} = 0$，只能将其留到之后的时间步。
3. 现在，我们在 G_{01}，还是一个非叶子节点，要继续做子节点选择。使用 PRNG 产生一个伪随机数 $r^{(k+2)} \in [0, 0.239)$，假设为 $r^{(k+2)} = 0.0393$，将其向右循环移位 0 和 $0.5 \times 0.239 = 0.1195$ 分别得到 $r_0^{(k+2)} = 0.0393$ 和 $r_1^{(k+2)} = 0.1588$，它们落入不同的子节点 god 和 little 所对应的区间。因为要嵌入 $m_{i+1} = 0$，所以我们走到 $r_0^{(k+2)}$ 对应的 god。
4. 现在，我们在 god，它是一个叶子节点，所以将 god 作为下一个词元，附加到上文 C 末尾。通过上述步骤，我们实际嵌入了 2 比特消息 $[1,0]$，随后的比特由后面类似的过程嵌入。

3.4.1　复杂度分析

对于 Discop，无论是嵌入还是提取算法，每一时间步主要包括两个过程：创建一棵 Huffman 树和若干次子节点选择（即从根节点走到叶节点）。对于按概率降序排列的分布，创建一棵 Huffman 树可达线性复杂度 $O(|V|)$，从树的根节点走到一个叶子节点的平均复杂度为 $O(\log|V|)$。算法 3.2 展示了创建 Huffman 树的线性复杂度算法。

3.5　部署场景

理论上，Discop 可以被部署于任何显式生成模型上。在这里，我们除了将 Discop 部署于文本生成模型上，还部署于自回归语音合成模型上，如表 3.2 所示。

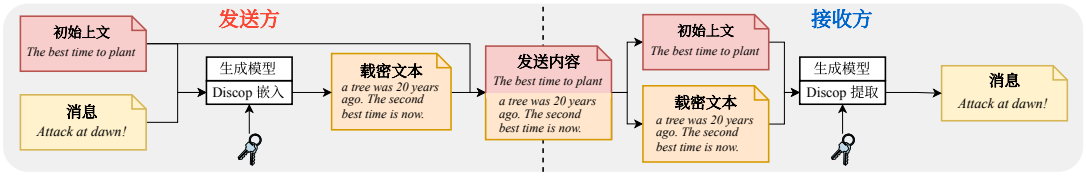
在开始隐写通信之前，通信双方必须提前就通信协议达成一致，即需要共享相同的设置。基本的设置包括使用的生成模型、PRNG、种子。针对特定任务，可能还有其他设置，将在下面介绍。

表 3.2　已实现 Discop 部署的生成任务/模型

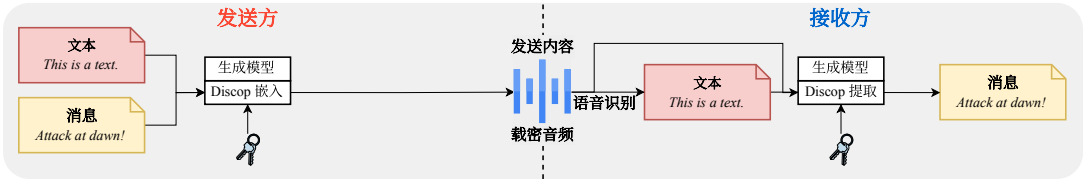
任务	描述	模型
文本生成 Text Generation	输入上文，生成其续写。	GPT-2 DistilGPT-2 Transformer-XL
语音合成 Text-to-Speech	输入一段文本，合成其对应的语音。	Tacotron + WaveRNN

文本生成：通信双方需要共享初始上文，该上文可以为空（对应无条件生成）或是一定数量的词元或句子（对应有条件生成）。如图 3.7(a) 所示，假设通信双方约定好使用 c 个词元作为初始上文，那么发送方 Alice 任意地选择 c 个词元作为初始上文，并使用生成模型和 Discop 嵌入算法在生成其续写的过程中嵌入消息，获得载密文本。她将初始上文和载密文本连接并发送给接收方 Bob。Bob 收到文本后，可以根据协议将其拆分为初始上文和载密文本两部分，然后使用生成模型和 Discop 提取算法提取消息。

语音合成：如图 3.7(b) 所示，发送方 Alice 使用生成模型和 Discop 嵌入算法在合成语音的过程中嵌入消息，获得载密音频，发送给接收方 Bob。通信双方需要共享语音对应的文本，这可以通过发送方直接将文本作为附件与语音一起发送，或通过接收方进行语音识别来实现。Bob 持有音频和文本，使用生成模型和 Discop 提取算法提取消息。



(a) 文本生成



(b) 语音合成

图 3.7　在两种生成任务上部署 Discop 的示意图

3.6　实验与评估

3.6.1　实验设置

核采样（nucleus sampling）也称为 top-p 采样[116]，现已在各种生成任务中广泛应用，它是这样做的：将所有词元按概率降序排列，取累积概率超过 p

的最小词元集合，归一化它们的概率（同乘一个比例使得概率之和等于 1），从归一化后的概率分布中随机采样。我们评估了截断参数 p 取不同值（$p = 0.80, 0.92, 0.95, 0.98, 1.00$）时隐写方法的性能。

在已实现 Discop 部署的生成任务中，设置分别如下。

文本生成：从影评数据集 IMDb[117] 中选取 100 条文本，取每一条文本的前 3 个句子作为初始上文，使用生成模型和消息嵌入算法为每个初始上文续写 100 个词元。

语音合成：从影评数据集 IMDb[117] 中选取 100 条文本，使用生成模型和消息嵌入算法合成每条文本的第 1 句对应的语音。

我们将 Discop 与之前试图实现可证安全的隐写方法 ADG、Meteor 进行对比。Meteor 的初始版本的嵌入率较低，其作者设计了一种启发式排序算法提高其嵌入率，我们对这两种版本都进行了测试。由于 Meteor 本质上是基于算术编码的基础隐写构造的改进版，所以这里不再对基于算术编码的基础隐写构造进行测试。

所有实验都在一台机器上进行，其硬件配置为 CPU 3.00GHz、128GB RAM、NVIDIA RTX 3090。涉及的数据集都通过 Hugging Face 的 Datasets 库[118] 加载。

3.6.2　评估指标

隐写的两大追求是安全性和效率。其中，效率又可以分为容量效率和时间效率。我们基于这些追求选择评估指标。

1. 安全性

隐写的首要追求是“行为安全”，即尽可能使得隐写行为和正常行为不可区分。我们使用 Cachin[88] 从信息论角度给出的安全性指标——载体分布与载密分布之间的 KL 散度 $D_{\mathrm{KL}}\left(P_{\mathrm{c}} \| P_{\mathrm{s}}\right)$ 作为安全性度量。在每一时间步，生成模型给出原始分布（即载体分布 P_{c}）。为了嵌入消息，隐写方法可能会对原始分布进行一定修改，我们将修改后的分布称为隐写采样的实际分布（即载密分布 P_{s}）。那么，可以直接根据公式 (2.2) 计算这两个分布之间的 KL 散度。我们主要关注 KL 散度的平均情况和最坏情况。

- **平均 KLD：**通过将所有时间步的 KL 散度求和除以总时间步数获得，可以表征隐写方法对原始分布的平均破坏程度，越低越好。
- **最大 KLD：**单个时间步的 KL 散度的最大值，可以表征隐写方法对原始分布的最严重破坏程度，越低越好。

2. 容量效率

嵌入率的定义是平均每生成一个词元可嵌入的消息长度。因为生成过程涉及随机采样，所以嵌入率的理论极限并不是固定不变的，现有的评估指标无法很

好地反映隐写方法的嵌入率离其理论极限有多远。为了更好地刻画嵌入率与理论极限（即所有时间步的概率分布的熵的平均值）之间的距离，我们提出了一种全新的度量指标——**对熵的利用率**（Utilization，缩写为 **Util**），将其定义为嵌入率与其理论极限的比值，越大越好。

3. 时间效率

为了实现实时隐写通信，我们通常希望隐写方法可以在较短时间内完成消息嵌入和提取。这些隐写方法的消息嵌入和提取过程非常相似，因此它们的消息嵌入时间与消息提取时间是相当的，这里我们只关注消息嵌入时间。我们定义了一个名为**平均时间**的度量，通过嵌入消息的总时间除以嵌入消息总长度计算，越小越好。

此外，我们还希望与正常生成过程相比，隐写的消息嵌入和提取引入的额外时间消耗尽可能少。

3.6.3　实验结果与分析

1. 与之前方法对比

因为 ADG 和 Meteor 只在文本生成任务上实现了部署，所以我们也只在这个生成任务上与它们进行对比。这里，我们只展示将几种隐写方法部署于 GPT-2 模型得到的实验结果（因为部署于其他文本生成模型得到的结论是相似的），如表 3.3 所示。在下文和表 3.3 中，Meteor w/o sort 和 Meteor 分别对应 Meteor 的基础版本和增加了启发式排序算法的版本，Discop w/o recursion 对应我们在 3.3 中提出的基础构造，Random Sampling 对应随机采样，即不嵌入消息的正常生成，作为参考。

根据前面的定义与分析，**平均时间**、**平均 KLD**、**最大 KLD**、**Util** 是衡量隐写方法性能的关键指标，表中也列出了**总时间**、**容量**、**熵**供参考。

总体来看，Discop 在几乎所有指标上都好于之前的方法。

安全性：ADG、Meteor w/o sort、Meteor 对原始数据分布的修改程度可能导致攻击者获得区分载体与载密的不可忽略的优势。而 Discop 能够很好地保持原始分布。

容量效率：ADG 和 Meteor 的 Util 相当，大概在 0.77 到 0.87 之间。Meteor w/o sort 的 Util 只有 0.67 到 0.76 之间。Discop 可以比较充分地利用熵，嵌入率可以达到理论极限的 0.92 到 0.95 之间。递归策略大大提高了这种基于分布副本的隐写方法的嵌入率，生成一个词元平均能嵌入 3.48 到 5.76 比特消息。

时间效率：在 p 取大多数值时，除 Meteor 外的所有隐写方法的总时间不会比 Random Sampling 的长太多。与随机采样相比，它们不会引入过多额外时间。相反，Meteor 设计了过于复杂的启发式排序算法，大大增加了消息嵌入时间。这

导致了如果要嵌入相同长度的消息，Meteor 所需时间是 Discop 所需时间的 3.45 到 144.88 倍，这在需要实时隐写通信时可能是无法接受的。

表 3.3　将 Discop 部署于 GPT-2 模型上的实验结果

方法	p	总时间	平均时间 ↓	平均 KLD ↓	最大 KLD ↓	容量	熵	Util ↑
ADG	0.80	96.71	3.16E-03	7.93E-03	6.76E-02	3.07	3.95	0.78
	0.92	104.48	2.57E-03	1.02E-02	4.75E-02	4.06	4.93	0.82
	0.95	114.72	2.62E-03	1.09E-02	4.73E-02	4.38	5.34	0.82
	0.98	150.68	3.08E-03	1.20E-02	4.54E-02	4.89	5.83	0.84
	1.00	846.27	1.57E-02	1.31E-02	4.99E-02	5.39	6.26	0.86
Meteor w/o sort	0.80	95.58	3.73E-03	5.13E-02	8.28E+00	2.56	3.83	0.67
	0.92	96.16	2.79E-03	8.17E-03	5.62E+00	3.44	4.82	0.71
	0.95	98.57	2.61E-03	3.40E-03	1.30E+00	3.78	5.15	0.73
	0.98	105.37	2.51E-03	6.59E-04	1.74E+00	4.20	5.61	0.75
	1.00	251.48	**5.56E-03**	1.05E-06	1.68E-05	4.52	5.96	0.76
Meteor	0.80	282.18	9.71E-03	5.57E-02	9.01E+00	2.91	3.76	0.77
	0.92	1359.87	3.33E-02	9.34E-03	4.63E+00	4.09	4.87	0.84
	0.95	2334.54	5.21E-02	2.77E-03	6.98E-01	4.48	5.23	0.86
	0.98	5559.88	1.16E-01	5.57E-04	8.23E-01	4.79	5.60	0.86
	1.00	47301.20	9.11E-01	1.06E-06	1.68E-05	5.19	5.98	0.87
Discop w/o recursion	0.80	101.33	5.52E-03	**0**	**0**	1.84	3.84	0.48
	0.92	102.78	5.00E-03	**0**	**0**	2.06	4.83	0.43
	0.95	103.11	4.74E-03	**0**	**0**	2.17	5.29	0.41
	0.98	105.81	4.70E-03	**0**	**0**	2.25	5.68	0.40
	1.00	145.81	6.38E-03	**0**	**0**	2.29	6.03	0.38
Discop	0.80	104.30	**2.99E-03**	**0**	**0**	3.48	3.79	**0.92**
	0.92	104.36	**2.29E-03**	**0**	**0**	4.55	4.86	**0.94**
	0.95	107.07	**2.21E-03**	**0**	**0**	4.84	5.18	**0.94**
	0.98	115.13	**2.17E-03**	**0**	**0**	5.29	5.59	**0.95**
	1.00	362.63	6.29E-03	**0**	**0**	5.76	6.08	**0.95**
Random Sampling	0.80	91.21	N/A	0	0	N/A	3.80	N/A
	0.92	90.89	N/A	0	0	N/A	4.80	N/A
	0.95	92.39	N/A	0	0	N/A	5.15	N/A
	0.98	95.20	N/A	0	0	N/A	5.59	N/A
	1.00	174.09	N/A	0	0	N/A	5.87	N/A

注：容量即嵌入率。Util 即对熵的利用率。总时间的单位是秒，平均时间的单位是秒/词元，平均 KLD、最大 KLD、嵌入率、熵的单位都为比特/词元。

2. Discop 引入的额外时间

以正常生成（随机采样）消耗时间为基准时间，我们测试了 Discop 的消息嵌入算法引入的额外的时间比例，实验结果如表 3.4 所示。由我们在第 3.4.1 小节中给出的复杂性分析可知，与正常生成相比，Discop 的大部分额外时间消耗在创建 Huffman 树上，所以额外时间大致与候选元素数 $|V|$ 成正比。对于候选元素数为 50,257 的 GPT-2 而言，top-p 截断（$p \neq 1.00$）去除了大量候选元素，使得额外时间相对较小。而对于候选元素数为 1024 的 WaveRNN，不管 p 的取值如何，Discop 引入的额外时间都相对较小。

表 3.4　Discop 的消息嵌入算法引入的额外时间比例

任务/模型	p	随机采样时间（秒）	Discop 采样时间（秒）	比例
文本生成 GPT-2	0.80	91.21	104.30	1.14
	0.92	90.89	104.36	1.15
	0.95	92.39	107.07	1.16
	0.98	95.20	115.13	1.21
	1.00	174.09	362.63	2.08
语音合成 WaveRNN	0.80	2500.56	2679.60	1.07
	0.92	2522.93	2599.81	1.03
	0.95	2520.88	2649.25	1.05
	0.98	2537.15	2700.39	1.06
	1.00	2744.50	3582.87	1.31

3. 隐写分析实验

虽然第 3.3.4 节已经给出了这种基于分布副本的隐写构造的安全性证明，但为了工作的完整性，我们仍试图使用现有的隐写分析方法区分载体（由随机采样生成）和载密（由隐写采样生成）。现有的隐写分析方法的一般流程为：1) 在与通信双方相同的设置下生成大量载体数据与载密数据，然后训练一个区分两种数据的二元分类器；2) 使用训练好的分类器来预测单个样本是载体还是载密。值得注意的是，生成的载体数据和载密数据是成对的，如在文本生成任务中，我们对每一条初始上文都分别生成一个载体和一个载密。

我们仅在 $p = 0.95$ 的设置下进行隐写分析实验，为了更好地训练隐写分析器，我们生成了更多的样本。除了生成样本数量和 p 的取值之外，其他设置与上文中一致。

文本生成：生成了 10,000 个载体和 10,000 个载密。使用 3 种文本隐写分析器，分别为 FCN[119]、R-BiLSTM-C[120]、BiLSTM-Dense[121]。

语音合成：生成了 1000 个载体和 1000 个载密。使用音频隐写分析器 Combined Time and Frequency（CTaF）[122]。

我们使用在测试集上的检测错误率 P_{E} 评估隐写方法的安全性，这个指标越接近 50% 表示隐写方法越安全。在所有隐写分析实验中，将生成的数据集被分成训练集、验证集、测试集，它们的比例为 3:1:1。实验结果如表 3.5 所示。这些隐写分析器的 P_{E} 都接近 50%，这表明，这些隐写分类器难以区分载体数据和载密数据，这也验证了 Discop 的安全性。

4. 生成质量评估实验

Discop 使载密分布与载体分布不可区分，这意味着理论上隐写生成的质量与正常生成的质量相当。为了验证这一点，我们对载体数据和载密数据进行了生成质量评估实验。

文本生成：将第 3 小节中生成的载体文本和载密文本数据集分别表示为 T_{c}

表 3.5　对 Discop 的隐写分析实验结果

任务	隐写分析器	P_{E}
文本生成	FCN[119]	50.10%
	R-BiLSTM-C[120]	50.45%
	BiLSTM-Dense[121]	49.95%
语音合成	CTaF[122]	49.50%

和 $\boldsymbol{T}_{\mathrm{s}}$。我们可以获得每个数据集的困惑度（perplexity，ppl）的均值 mean(ppl) 与标准差 std(ppl)，如表 3.6 所示。可以看出它们很接近，表明 Discop 生成的文本质量与随机采样生成的文本质量相当。表 3.7 展示了生成文本载体与载密的实例，也可以直观看出载体与载密文本质量相当。

表 3.6　文本生成质量评估

数据集	mean(ppl)	std(ppl)
$\boldsymbol{T}_{\mathrm{c}}$	42.858	21.512
$\boldsymbol{T}_{\mathrm{s}}$	42.229	22.770

表 3.7　生成文本载体与载密实例

上文： I remember this film, it was the first film I had watched at the cinema.
载体（随机采样）： There was a last minute switch because I had watched it a decade ago in the City School balcony of a big theatre where the writers were also prominent. So I just watched it, cut it, then cut it. When I was about 15 years old, I went to the Millie Murphy director's ticket and watched The Girl who Would Get Away With Murder and then I watched an Ode to love. It's just magic. Once we got into that period with Baby Driver and Gravity and Rushmore, the characters played what I'd always dreamed. You know, they'd come up with creative, futuristic, ridiculously stupid songs from here and then the originals would come out, the normal birds would just sit there and wait for a substitution in the end credits to get on the schedule. And I just loved that whole process of watching this wonderful, there is so much greater potential here and it's been so exciting to continue to put in so much work and still succeed with these great movies in the works
载密（Discop）： I was very nervous, at the time. I was 22, and the director was actually at a birthday party in Paris and just saying the doors were closed. But I put on a large black motorcycle, got my album saved and drove on, doing videos about history and the films. So I was really excited. And I found that in this movie, we were at a concert in London. And that concert was where every song and every act was at the movie playing the music that could be heard and the message was, "By God, there's people. But time has passed, and we're here, maybe from another world. Let's see how things can go." And at the end of the movie I said, "I hope the thing happened a bit, because there's no way you can change what you know. You know? Something has happened. Just let it happen. Let it happen." And at the end, I say to myself, "Hey, maybe we can change

语音合成：将第 3 小节中生成的载体语音和载密语音数据集分别表示为 $\boldsymbol{A}_{\mathrm{c0}}$ 和 $\boldsymbol{A}_{\mathrm{s}}$，另外再额外生成包含 1000 个载体语音的数据集 $\boldsymbol{A}_{\mathrm{c1}}$。我们使用 Perceptual Evaluation of Speech Quality[123]（PESQ）来评估语音生成质量，它可以衡量两个

音频片段之间的相似度。对于来自两个不同数据集中的一对音频（对应相同的文本），我们可以计算它们之间的 PESQ。表 3.8 展示了 A_{c0} 和 A_s 之间和 A_{c0} 和 A_{c_1} 之间的 PESQ 的均值和标准差。可以看出它们数值很接近，表明 Discop 合成的语音质量与随机采样合成的语音质量相似。

表 3.8　语音合成质量评估

数据集		mean(PESQ)	std(PESQ)
A_{c0}	A_s	3.354	0.122
A_{c0}	A_{c1}	3.356	0.128

3.7　讨论

带外开销：通常情况下，在执行隐写嵌入算法之前，发送方会先对消息进行加密。相应地，在执行隐写提取算法之后，接收方需要对消息密文进行解密。此外，隐写嵌入算法和隐写提取算法还涉及隐写密钥。因此，整个隐写系统一般涉及两套密钥：加密密钥和隐写密钥，这两者都需要提前通过安全的带外信道传输。对于 Discop 而言，隐写密钥是用于初始化 PRNG 的种子，其他设置（如选用的生成模型、PRNG）可视为协议的一部分公开。为了避免带外开销，一种可能的解决方案是引入现有的公钥隐写方法，如 [90,124]，通过构建一个类似于混合加密系统的混合隐写系统，先利用公钥隐写方法协商这两套密钥，后续使用 Discop 进行隐写通信。

3.8　本章小结

本章提出了一种适用于显式生成模型的全新的可证安全隐写构造：通过循环移位的方式为模型给出的概率分布创建多个分布副本，并利用分布副本的索引值来表达消息。为了防止提取歧义，我们给出了这种隐写构造的唯一提取条件，并引入了争议区间的概念。通过数学推导，我们给出了这种隐写构造的嵌入率期望值。随后，我们介绍了创建分布副本的一种简单的等价实现，给出了这种隐写构造的安全性证明。接着，我们通过递归分组的思想提升了这种隐写构造的嵌入率。具体地，我们通过 Huffman 树将原先的一轮多元分布采样分解为多轮二元分布采样，这样每轮二元分布采样都有一定概率能嵌入 1 比特消息。我们将提升后的隐写算法称为 Discop，并实现了其在文本生成和语音合成这两个生成任务上的部署。在文本生成任务上，将 Discop 与现有的试图实现可证安全的隐写构造 ADG、Meteor 进行对比实验。实验结果表明，在安全性、容量效率、时间效率方面，Discop 在几乎所有指标上都好于之前的隐写构造。值得一提的是，我们提出了一种全新的容量效率度量指标——对熵的利用率（Util），可以更好地

刻画嵌入率与其理论极限之间的距离。Discop 的嵌入率能达到理论极限的 92% 到 95% 左右。对比正常生成，Discop 引入的额外时间相对较小。隐写分析实验表明，现有隐写分析器无法有效区分载体和由 Discop 生成的载密。生成质量评估实验表明，载体和由 Discop 生成的载密质量相当。

第 4 章　基于生成式可证安全隐写的蜜罐加密方案

我们观察到，生成式隐写与蜜罐加密共享相似的内核。并且从某种意义上说，它们有着相同的追求——力求在编码时尽可能地保持原始数据分布。本章将尝试着把生成式可证安全隐写领域的一些思想应用于文本蜜罐加密系统的设计中，以改进其性能。其中第 4.1 节对现有的分布编码转换器（DTE）存在的问题进行总结性回顾；第 4.2 节对生成式隐写与蜜罐加密之间的关系进行分析；第 4.3 节提出基于深度生成模型和算术编码的 DTE；第 4.4 节讨论在传统加密方法的选择上的注意事项；第 4.5 节从时间效率、压缩率、安全性三个方面展开实验与分析；第 4.6 节对本章内容进行总结。

4.1　现有的 DTE 的问题

蜜罐加密的加密算法包含两个步骤：先使用 DTE 将明文编码为种子（伪随机比特序列），再使用传统加密算法将种子加密为密文。蜜罐加密的解密算法则是逆过程：先使用传统加密算法的解密算法将密文解密为种子，再使用 DTE 将种子解码为明文。蜜罐加密的作用是，当攻击者尝试使用错误密钥解密密文时，会得到看似合理的假明文（也称为诱饵明文）。这样，即使她通过暴力破解获得了所有候选明文，也难以从中定位出混迹于其中的唯一真明文。

蜜罐加密的核心是 DTE，它应具备两个核心功能：

1. **分布建模**：对明文分布进行尽可能准确的建模。
2. **可逆映射**：根据模型给出的分布构建明文与种子（伪随机比特序列）之间的可逆映射，以实现编码与解码。

这样，我们可以将 DTE 的失真分解为两部分：

1. **建模失真**：模型潜在分布 $\boldsymbol{D}'$ 与明文实际分布 $\boldsymbol{D}$ 之间的距离。
2. **编码失真**：候选明文分布 $\boldsymbol{D}''$ 与模型潜在分布 $\boldsymbol{D}'$ 之间的距离。

我们在第 2.7 节中提到，现有的 DTE 通常采用传统的统计模型（如马尔可夫模型、PCFG 模型）来建模明文分布，并通过基于 CDF 的定长编码方案将明文转换为种子。传统的统计模型对自然语言文本这种复杂数据的建模能力非常有限，也基本不具备泛化能力，随着近年来深度学习特别是深度生成模型的高速发展，传统的统计模型已不再是最优选项。基于 CDF 的编码方案往往需要更长的编码来表示明文信息。而且，在设置精度时需要谨慎：如果太小，原始分布会受到严重破坏；如果太大，不仅会导致加密和解密的时间变长，还会造成较大的存储和传输开销。如何减小建模失真和编码失真，是一个亟待解决的问题。

4.2　生成式隐写与蜜罐加密之间的关系

为了方便描述，这里以生成式文本隐写与文本蜜罐加密之间的关系为例进行说明。将文本换成其他信息载体，情况也是类似的。

生成式隐写中的消息和蜜罐加密中的种子都有着比特序列的形式。

如图 4.1 所示，生成式文本隐写与文本蜜罐加密的 DTE 之间是某种“对偶关系”。具体地说，生成式文本隐写的消息嵌入算法和文本蜜罐加密的 DTE 解码算法都是将比特序列转换为自然语言文本的过程，生成式文本隐写的消息提取算法和文本蜜罐加密的 DTE 编码算法都是将自然语言文本转换为比特序列的过程。

更为巧合的是，生成式隐写和蜜罐加密的 DTE 都追求尽可能地保持分布：生成式隐写希望载密分布与载体分布尽可能接近，蜜罐加密希望候选明文分布与模型潜在分布尽可能接近。实际上，载体分布正是模型潜在分布，载密分布也与候选明文分布是等价的。理想情况下的生成式隐写是可证安全的，在这种情况下，载密分布与载体分布相等或者不可区分。

这个观察启发我们尝试将生成式可证安全隐写领域的一些思想应用于蜜罐加密系统的设计中。

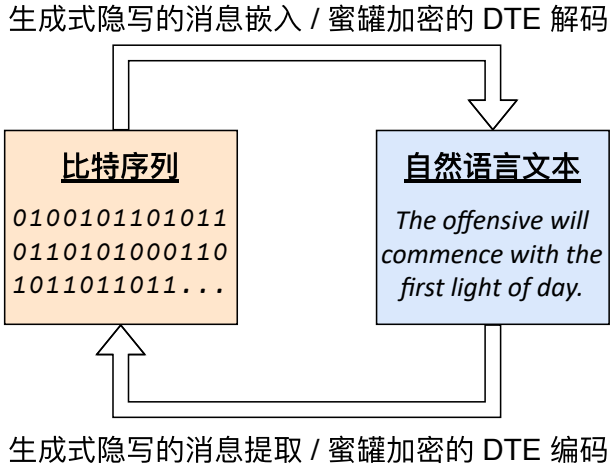


图 4.1　生成式文本隐写与蜜罐加密之间的关系

4.3　基于深度生成模型和算术编码的 DTE

我们考虑从以下两个方面入手，尝试将生成式隐写领域的思想应用于文本蜜罐加密系统的设计中，从而改进文本蜜罐加密系统：

1. **减小建模损失：**当前生成式隐写通常采用较为流行的深度生成模型，鉴于它们在对自然语言文本分布的建模方面展现出卓越的性能，我们将其引入到文本蜜罐加密系统的设计中。与传统的统计模型相比，深度生成模型能够处理更加复杂的数据模式，具备更加强大的泛化能力和自适应性。

2. **减小编码损失：**基于算术编码的隐写构造是生成式隐写中的一类经典方法，考虑到算术编码可以较好地保持分布，我们将其引入到文本蜜罐加密系统的设计中。使用算术编码来编码消息，可以使得候选明文分布 D'' 很接近于模型潜在分布 D'。此外，算术编码本身具备出色的数据压缩能力，使得编码得到的种子更短，进而使得密文更短，从而大幅降低存储和传输开销。

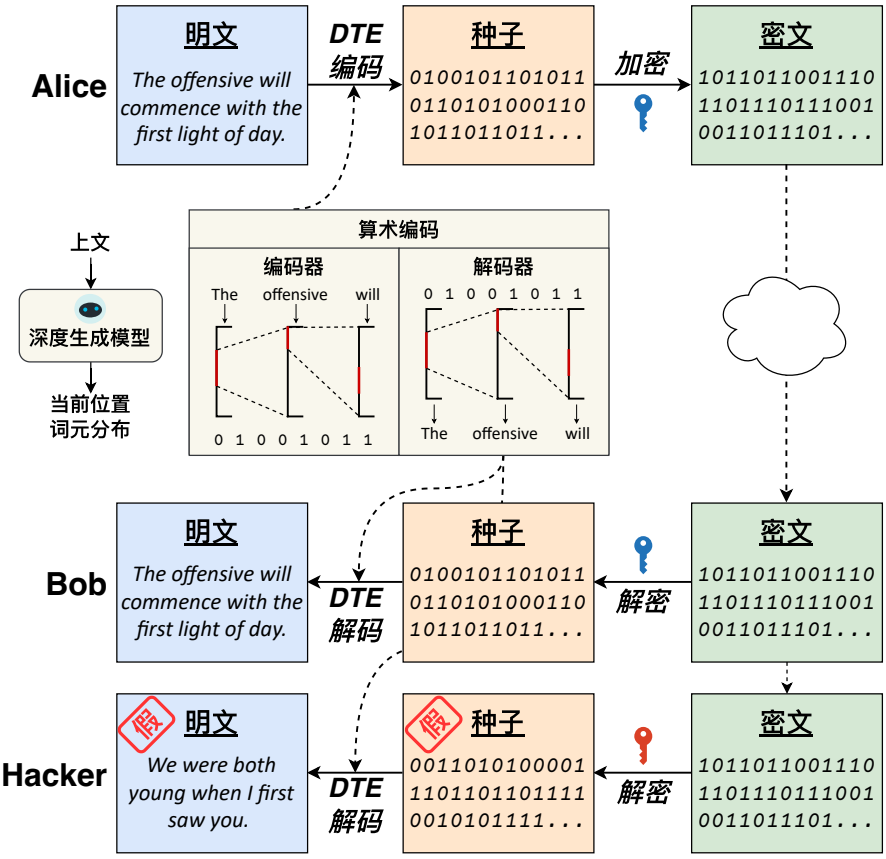


图 4.2　基于深度生成模型和算术编码的文本蜜罐加密方案

具体地，如图 4.2 所示，我们可以将这种基于深度生成模型和算术编码的 DTE 编码算法的基本框架描述如下：

1. **词元化：**使用深度生成模型的词元分析器（tokenizer）对待编码明文进行分词，得到一系列词元；
2. **逐词元处理（循环）：**在每一时间步 t，深度生成模型在上文 $x_{<t}$ 的条件下预测当前位置的词元概率分布，算术编码根据明文当前位置的词元及模型给出的概率分布更新内部状态，并适时输出种子片段，重复这个过程，直至明文所有词元均处理完成；
3. 将所有种子片段连接起来，即可获得种子。

蜜罐加密追求真明文与假明文难以区分，然而，上述基本框架如果不加处理，其在明文层面和种子层面都可能会引入可区分特征。

1. **明文（文本）层面：**自然语言文本（特别是正式文书）中的每个句子通常

是完整句子，其末尾一般都有句末标点（如句号、问号、感叹号等）。然而，使用错误密钥解密得到的假明文的末尾很可能是一个不完整的句子，不包含句末标点。

2. **种子（伪随机比特序列）层面：**由真明文编码得到的种子在 DTE 解码时会没有冗余地“走完”解码过程。然而，由于算术编码是变长编码方案，使用错误密钥解密得到的种子很可能包含冗余比特。

此外，在种子层面，1 字节（等于 8 比特）是最常用的数字信息基本单位，通常也是加密和解密的最小处理单位，然而，算术编码是变长编码方案，使用其对明文进行编码产生的种子 S 的长度不一定恰好为 8 的整数倍。

我们希望通过填充（padding）的方式解决上述问题、消除可区分特征。直觉上，填充方案的设计应满足下列原则：

1. 填充方案能够消除上述可区分特征，且不会引入新的可区分特征；
2. 在 DTE 解码时，填充部分能够被容易且准确地识别并忽略。

根据这些原则，我们设计了明文和种子两个层面上的伪随机填充方案。

4.3.1　明文层面上的伪随机填充

如算法 4.1 所示，在将明文编码为种子之前，我们先利用生成模型 $\mathcal{G}$ 为原始明文生成一个续写句子，然后随机地移除其末端的部分词元，以确保这段续写不构成一个完整句子。相应地，在 DTE 解码过程中，我们可以通过去除末端的不完整句子来轻松去除这种填充。为了使这种填充符合自然语言文本的统计特性，生成时我们使用的是随机采样的策略。

算法 4.1　message_padding$(M,\mathcal{G})$：明文层面上的伪随机填充

Input: 原始明文 M，生成模型 $\mathcal{G}$
Output: 填充后的明文 M'

padding_tokens ← [ ]
while not contain_ending_punctuation(padding_tokens) **do**
    $\mathcal{P} \leftarrow \mathcal{G}(M \| \text{to\_string}(\text{padding\_tokens}))$　　// 下一个词元的条件概率分布
    $w \leftarrow \text{random\_sample}(\mathcal{P})$
    padding_tokens ← padding_tokens $\| w$
end
$r \leftarrow \text{random\_int}(1, \text{size}(\text{padding\_tokens}) - 1)$
padding_tokens ← padding_tokens[0:r]　　// 移除末端部分词元
$M' \leftarrow M \| \text{to\_string}(\text{padding\_tokens})$
return M'

需特别指出的是，有时待加密的原始明文并非以标准的句末标点结束。为确保 DTE 解码过程中不会误删明文末尾的内容，在进行编码前，我们会在这类不是以标准的句末标点结束的明文末端添加一个句号。这一措施虽然会造成解密得到的明文与原始明文有一个标点的差别，但不会改变其原始语义。

4.3.2　种子层面上的伪随机填充

在对种子进行加密之前，我们先将其填充至长度为 8 的整数倍。如果直接使用密码学中现有的填充方案，如全零填充、PKCS #7 填充，则会使得使用错误密钥解密得到的种子中几乎不可能有格式正确的填充内容，因此我们需要设计一种新的填充方案。

考虑到种子是伪随机的，我们也使用伪随机比特序列来对种子进行填充。设种子原本的长度为 s，填充长度为 z，那么有 $s+z\equiv 0 \pmod 8$ 成立。为了简便，取 $z=(8-s \bmod 8) \bmod 8$。需要注意的是，种子层面的填充可能会在 DTE 解码时引入新的文本。因此，我们应避免种子层面的填充解码出句末标点符号，以免无法正确地移除明文层面的填充。如算法 4.2 所示，我们利用试错法的思想，为原始种子生成一个长度为 z 的伪随机比特序列填充并尝试将其解码为文本，检查其是否会引入句末标点符号。如果其会引入句末标点符号，则需要重新生成伪随机比特序列填充，直至其不会引入句末标点符号。

算法 4.2　seed_padding(S)：种子层面上的伪随机填充

Input: 原始种子 S
Output: 填充后的种子 S'
$s \leftarrow \mathsf{size}(S)$
$z=(8-s \bmod 8) \bmod 8$
if $z=0$ **then**
    $S' \leftarrow S$
else
    padding_bits ← [ ]
    while True **do**
        padding_bits ← get_random_bits(z)
        if not contain_ending_punctuation(decode_step(padding_bits)) **then**
            break　　// 种子层面的填充不应引入句末标点符号
        end
    end
    $S' \leftarrow S \|$ padding_bits
end
return S'

在结合了两个层面的伪随机填充算法之后，我们设计的基于深度生成模型和算术编码的 DTE 编码的过程如算法 4.3 所示，与之相对应的 DTE 解码过程如算法 4.4 所示。

4.3.3　算术编码

Shannon 的信源编码定理[93]给出了数据压缩的理论极限。根据这个定理，一个词元的最佳码长是 $-\log_2 p$ 比特，其中 p 为该词元出现的概率。那么，所有词元的平均码长为 $-\sum_j p_j \log_2 p_j$ 比特，即词元的概率分布的信息熵。

为了逼近数据压缩的理论极限，算术编码不会对消息序列进行分割处理，而

算法 4.3　Encode$(M, \mathcal{G})$：DTE 编码算法

Input: 明文 M，生成模型 $\mathcal{G}$
Output: 种子 S

$M \leftarrow \text{message\_padding}(M, \mathcal{G})$　// 明文层面上的伪随机填充
$\text{tokens} \leftarrow \text{tokenize}(M)$　// 词元化
$l \leftarrow \text{tokens.size}()$
$S \leftarrow [\,]$　// 种子
for $t \leftarrow 0$ **to** $l - 1$ **do**
    $\mathcal{P} \leftarrow \mathcal{G}(\text{tokens}[:t])$
    $S_t \leftarrow \text{encode\_step}(\mathcal{P}, \text{message\_tokens}[t])$
    $S \leftarrow S \| S_t$
end
$S \leftarrow \text{seed\_padding}(S)$　// 种子层面上的伪随机填充
return S

算法 4.4　Decode$(S, \mathcal{G})$：DTE 解码算法

Input: 种子 S，生成模型 $\mathcal{G}$
Output: 明文 M

$l \leftarrow S.\text{size}()$
$\text{decoded\_tokens} \leftarrow [\,]$
$i \leftarrow 0$
while $i < l$ **do**
    $\mathcal{P}^{(t)} \leftarrow \mathcal{G}(\text{decoded\_tokens}[:t])$
    $\text{token}, n \leftarrow \text{decode\_step}(\mathcal{P}^{(t)}, S[i:])$　// 解码出的词元及其对应的编码长度
    $\text{decoded\_tokens} \leftarrow \text{decoded\_tokens} \| \text{token}$
    $i \leftarrow i + n$
end
$M \leftarrow \text{to\_string}(\text{decoded\_tokens})$
$M \leftarrow \text{remove\_message\_padding}(M)$　// 去除末端的非完整句子
return M

是直接将整个消息序列编码为 $[0, 1)$ 区间内的一个二进制小数。具体地，算术编码的编码过程是递归地缩小区间的过程：

1. 起初，区间的初始值为 $[0, 1)$。
2. 根据生成模型预测的词元概率分布，将当前区间划分为 $|V|$ 个互不重叠的子区间分配给词表中的每个词元，其中 $|V|$ 表示词表规模，每个子区间的长度与其对应的词元的概率成正比；读入一个待编码词元，选择其对应的子区间作为新的当前区间。重复这个过程，直到消息的所有词元都被处理。
3. 待所有词元都被处理完成后，最终区间中的任意一个数值（二进制）都可以作为整个序列的编码。

编码过程中的区间大小反映了所有已处理词元的联合概率，使得总编码长度逼近数据压缩的理论极限。

上述描述是建立在计算机拥有无限精度这个假设上的一种简化，并不符合实际情况。在具体实现时，算术编码通常使用两个有限精度的二进制移位寄存器来分别存储当前区间的左右端点，也是类似地通过递归地缩小区间的方式来实

现编码。由于精度有限，为每个词元分配的子区间的左右端点可能会经过一定的舍入。需要注意的是，每处理一个词元，如果两个移位寄存器有公共前缀，移位寄存器就会进行重新归一化（renormalization）：两个寄存器同时左移将公共前缀移出，并在右侧补充新的比特，左端点对应的寄存器补充全 0，右端点对应的寄存器补充全 1。待所有词元都被处理完成后，将左端点对应的寄存器的所有输出拼接起来就是整个明文序列对应的编码。为了方便理解，表 4.1 给出了这种实现方法处理一个词元的例子。假设有 a、b、c 三个可选词元，概率都为 1/3，将寄存器的精度设为 8。若要编码 a，由于其对应的子区间的公共前缀为 0，通过左移将这一比特移出，然后左端点补充 0，右端点补充 1。编码 c 也是类似的。若要编码 b，由于其对应的子区间没有公共前缀，那么不需要进行重新归一化。

表 4.1　基于移位寄存器实现的算术编码示例

词元	概率	精度为 8 的区间		输出前缀	重新归一化后的区间
		分数	二进制		
a	1/3	[0/256, 85/256)	**0**0000000 – **0**1010100	**0**	00000000 – 10101001
b	1/3	[85/256, 171/256)	01010101 – 10101010	/	01010101 – 10101010
c	1/3	[171/256, 256/256)	**1**0101011 – **1**1111111	**1**	01010110 – 11111111

4.4　传统加密方法的选择

蜜罐加密系统的加密算法的第二步是使用传统加密方法对 DTE 编码得到种子进行加密。需要注意的是，一些密码体制（如分组密码的 CBC 模式）会先将待加密内容填充至特定长度再进行加密。然而，与在 4.3.2 中提到的类似，如果一个蜜罐加密系统采用了这些需要填充的密码体制，使用错误密钥解密得到的种子几乎不可能有格式正确的填充内容，这样就引入了可区分特征，暴露了该密钥是错误密钥。因此，蜜罐加密系统应避免使用这种需要填充的密码体制。

4.5　实验与评估

4.5.1　实验设置

考虑到明文的多样性，我们选择 OpenWebText2 数据集[125]作为实验数据集。该数据集收集了 Reddit 上得分较高的帖子中提及的网页内容。我们从该数据集中随机选取 100 条文本，取每一条文本的前 4 个句子作为明文，使用本章设计的 DTE 对其进行编码得到种子，然后使用传统加密算法对种子进行加密得到密文。我们选用的传统加密算法为 Salsa20，这是一种流密码算法，密钥长度为 256 位。在深度生成模型的选择方面，我们分别选用了 Llama 2[126]、GPT-2[36]、DistilGPT-2[127]分别进行实验。其中，Llama 2 模型我们选用其 7B 版本，而 GPT-2

模型我们则选用其 124M、355M、774M 和 1.5B 这四个版本。

所有实验都在一台机器上进行，其硬件配置为 CPU 3.00GHz、128GB RAM、NVIDIA RTX A6000。数据集通过 Hugging Face 的 Datasets 库[118]加载。

4.5.2　评估指标

1. 时间效率

我们通常希望 DTE 能够快速地完成编码和解码，为此，我们统计了 DTE 的平均编码时间和平均解码时间，单位为秒/词元。鉴于传统密码体制的加解密速度非常快，其耗时相较于 DTE 编解码来说可以忽略不计，所以我们不对这部分时间进行统计。

2. 压缩率

压缩率作为评价信源编码压缩效果的一个关键指标，其定义为原始数据大小与编码后数据大小的比值。压缩率的数值越大，表示数据压缩的效果越佳。

3. 安全性

我们使用模型潜在分布 $\boldsymbol{D}'$ 与候选明文分布 $\boldsymbol{D}''$ 之间的 KL 散度作为评估安全性的一项指标，其表征了编码失真。每一时间步，生成模型给出模型分布 $\boldsymbol{D}'$。算术编码为每个词元分配的子区间的左右端点可能会经过一定的舍入，候选明文分布 $\boldsymbol{D}''$ 是模型分布 $\boldsymbol{D}'$ 经过舍入得到的结果。

此外，我们从攻击者可能采取的攻击策略着手考虑。攻击者在截获密文后，可以对密文进行暴力破解攻击：通过枚举所有可能的密钥来解密密文获得所有可能的种子，通过对所有可能的种子执行 DTE 解码来获得所有候选明文。为了从中快速定位出真明文，她应选择何种指标对候选明文进行排序？假设存在一个最优排序指标，多次使用这个指标进行攻击实验，真明文的平均排序越接近 50%，则表明这个蜜罐加密系统越安全——这是因为 50% 相当于随机猜测得到的结果。然而，目前并无针对文本蜜罐加密系统的攻击排序指标的相关研究。我们启发式地尝试使用由 AI 生成文本检测器给出的“不是 AI 生成文本的概率”即“是人类写的文本的概率”作为排序指标。因为真明文一般为人类写的文本，而假明文可视为 AI 生成文本。我们选用的 AI 生成文本检测器为 Fast-DetectGPT[128]。

4.5.3　实验结果与分析

实验结果如表 4.2 所示。实验结果表明，随着使用的深度生成模型的规模的增大，平均编/解码时间呈现出相应的增长趋势。举例来说，在纯 CPU 环境下，使用 Llama2 7B、GPT-2 124M，DistilGPT-2 82M 对 100 个词元进行编码所需的平均时间分别为 92.8 秒、14.5 秒、13.8 秒。当引入 GPU 加速后，编码 100 个词元所需的平均时间则分别降至 14.6 秒、2.86 秒、2.67 秒。平均解码时间约为相

表 4.2　本章提出的蜜罐加密方案的各项指标

模型	编码时间		解码时间		压缩率	KL 散度	p_{human} 排名
	CPU	GPU	CPU	GPU			
Llama 2 7B	9.28E-01	1.46E-01	5.68E-01	5.20E-02	9.62	2.57E-12	50.55%
GPT-2 1.5B	3.27E-01	9.31E-02	1.94E-01	4.29E-02	7.97	2.59E-09	56.58%
GPT-2 774M	2.69E-01	5.09E-02	1.52E-01	2.84E-02	7.82	4.39E-08	34.08%
GPT-2 355M	1.88E-01	3.92E-02	1.07E-01	2.02E-02	7.51	1.71E-08	20.18%
GPT-2 124M	1.45E-01	2.86E-02	8.31E-02	1.22E-02	6.80	4.00E-09	20.00%
DistilGPT-2 82M	1.38E-01	2.67E-02	7.71E-02	8.15E-03	5.89	1.34E-08	22.86%

注：表中各项指标均为平均值。p_{true} 表示由 Fast-DetectGPT 给出的“不是生成文本的概率”。编/解码时间的单位为秒/词元，KL 散度的单位为比特/词元。

同设置下的平均编码时间的 1/3～2/3 左右。随着使用的深度生成模型的规模的增大，平均压缩率也呈现出相应的增长趋势，这是因为规模越大的模型对自然语言文本分布的建模越准确。通过使用算术编码，原本平均需要 5.89～9.62KB 存储的数据现在只需要 1KB 即可，大幅节省了存储和传输开销。平均 KL 散度的值在 10^{-12}～10^{-8} 这个量级，表明使用算术编码引入的编码损失较小。如果使用规模较小的模型，我们启发式定义的攻击排序指标能够将真明文排在比较靠前（20%）的位置，这虽然说明了这种排序指标的有效性，但也表明这些模型的建模损失还比较大。如果使用规模较大的模型，这种攻击排序指标将真明文排在接近 50% 的位置。

在 GPU 环境下，我们选用 Llama 2 7B 模型进行算术编码与 CDF 编码的对比实验。对于 CDF 编码，精度取 16、24、32、40、48。实验结果如表 4.3 所示。结果表明，算术编码的压缩率是 CDF 编码的 5～15 倍，这表明算术编码能够显著降低存储和传输开销。在编码失真方面，CDF 编码在低精度时 KL 散度较大，而算术编码与精度为 48 的 CDF 编码的 KL 散度相近。不过，算术编码在时间消耗方面略高于 CDF 编码。具体而言，算术编码所需的编码时间比 CDF 编码高出约 25%～32%，而解码时间则比 CDF 编码高出约 1%～5%。

表 4.3　算术编码与 CDF 编码对比

编码算法	编码时间	解码时间	压缩率	KL 散度
CDF 16	1.16E-01	5.08E-02	1.84	9.04E-01
CDF 24	1.15E-01	5.07E-02	1.23	1.77E-03
CDF 32	1.11E-01	4.97E-02	0.94	1.63E-06
CDF 40	1.16E-01	5.14E-02	0.73	6.63E-10
CDF 48	1.16E-01	5.09E-02	0.61	3.13E-13
算术编码	1.46E-01	5.20E-02	9.62	2.57E-12

注：CDF 后的数字表示精度。表中各项指标均为平均值。编/解码时间的单位为秒/词元，KL 散度的单位为比特/词元。

表 4.4 展示了本章提出方案在选择 Llama 2 模型时的真假明文实例，其中灰色字体部分表示文本层面的填充。可以看出，这种基于深度生成模型和算术编码

的文本蜜罐加密方案可以产生质量较高的诱饵明文。

4.6　本章小结

本章尝试将生成式可证安全隐写领域的一些思想应用于文本蜜罐加密中。从减小 DTE 的建模损失和编码损失着手，设计了一种基于深度生成模型和算术编码的 DTE：首先对明文进行分词，然后使用算术编码根据深度生成模型预测的概率分布将明文编码为种子。为了消除可区分特征，我们在明文和种子两个层面上设计了伪随机填充方案。从时间效率、压缩率、安全性三个方面展开实验，并从攻击者视角启发式地设计了针对文本蜜罐加密的攻击排序指标。实验结果表明，该 DTE 具有良好的编解码速度。算术编码使得该 DTE 具有较高的压缩率，较小的编码损失。当使用规模较大的深度生成模型时，启发式设计的攻击排序指标将真明文排在接近 50% 的位置，接近随机猜测得到的结果，这说明提出的方案在这种设置下的建模损失较小、安全性较高。

表 4.4　真假明文举例

类型	文本
真明文	In the past few years, there has been a surge of deep generative models, which can approximate complex non-trivial data distributions and synthesize realistic data. Moreover, their popularity is gradually increasing, providing great scenarios and techniques for efficient, provably secure steganography in practice. The increasing popularity of pretrained text generation models, however
假明文	The sounds of nature lingered throughout, painting a tranquil atmosphere where the mind could dwell over the mysteries of the universe and the enigma of human existence. Strange ripples of a magic unexplained seemed to linger on the surface of silence, as strange whispers and hums and echoes could be heard calling out slowly from the distance⋯ever pulling together something not
假明文	Others, however, found the time to be difficult, and kept themselves busy with unending tasks, trying to put behind them their inability to go to sleep. In the thick of their storm, they would spin themselves into crisps with endless talk of shoulds, coulds, and woulds. Some would claim that effort was futile, but the truth is, many of the most innov
假明文	Her mind drifted happily until the shivers of the sea choir shook her body in alert. Smothered by a sea of insight, her eyes travelled across the others with wisdom eyeing them with manic excitement, their words underlining the court of several truths
假明文	Here is the start of a poem written on such a night. God Giving and Glorious, is a one-term art class designed to help you build your poetic prowess. With the guidance and direction of Sydney Williams, you will learn how artists use their work to position themselves as exceptional leaders. Through study and practical application you will develop a deeper understanding
假明文	It is time to celebrate the unbelievable beauty and the architectural splendor of Gujarat, as it opens its exquisite door for the cultural event of a lifetime. It is a time to welcome the month of Janmashthami, by celebrating Shree Krishna Janma Divas, and by visiting the City of Lord Krishna, DWARKA. So, pack your bags and boldly place your foot on the arrivals
假明文	The memory of this, the first time I held our son, Justin, awakened sleeping emotions of love, bliss and wonder. The now, when I held my new baby daughter Kiana, gripped me with similar feelings of wonder as I thought back to the birth of my older children: Delilah, who is 6 years old, Kelton, who is 4 and Kiana, only a
假明文	Looking up at the stars and listening to the night whispered softly into my ears, I would, quite often, fall into a spell of enchantment and despair, all at once. However, my unearthly tresses were frazzled into rustic flax, when it had come to a startling turn around time. My world had changed; no longer
假明文	This is how our writer spends her evenings with her muse, her magic wand and an empty spiral notebook. This is her second work of art. One article in each of two consecutive Sunday newspapers of the New York Times. These were the only two articles that remain, for this was home to our writer and her telescopic lens. As one no
假明文	Yet before stories of dreams and fantasies could be written would come the night to dawn, evoking a time for silence. For some time, that time has when shame like the devious whispers of a village gossip, became the words of a whispering romantic. He closed upon the breach
假明文	Though now this serenity too shall fall, for it is upon this lonely trail winding through the lush forest that an impending terror only known for its horrifying tales shall soon befall on the unwary. What terrors could lie in wait in this land filled with mystic enchantments⋯? Come my dear reader, the time has come to explore the mystical realm
假明文	I awoke to a familiar view... and yet not so much. My words whispered, “I love this sight!” Though the darken depths are becoming all too familiar, the twilight time still holds a variety of beauty that sways me from the last jump of shadows. Yes, the scented absence
假明文	It was during such moments that Noliviere Bailee found herself embracing a form of poetry. It was a fancier version than the one that had been infused since her early elementary school days. After all, this poetry was expressed through the pages of paper and ink; it was a means to write to her heart’s desire. The words had become a cheap hobby
假明文	As I found the delightful space in my journal, I felt truly blessed and grateful. As I navigated brushes, bristles, and the calligraphy pen that moved across the pages, the night became ever so sweet. There was a sense of tranquility that came over me, and as these simple moments drifted into the stillness of my being, I knew I was content. This space became the muses

注：灰色字体部分表示文本层面的填充。

第 5 章　总结与展望

5.1　工作总结

本文在互联网迅猛发展与广泛普及这个时代背景下，从数据隐私保护的两大核心需求——隐蔽性和保密性出发，围绕生成式可证安全隐写和文本蜜罐加密展开研究。分析了现有的追求可证安全的隐写构造在实践中存在的问题。针对显式生成模型，从分布保持的角度，设计了全新的基于分布副本的高效可证安全隐写构造。通过将生成式可证安全隐写领域的思想应用于蜜罐加密，设计了基于深度生成模型和算术编码的文本蜜罐加密方案。本文的主要创新点总结如下。

1. 提出了一种基于分布副本的高效可证安全隐写构造

通过循环移位的方式为模型给出的概率分布创建多个分布副本，并利用分布副本的索引值来表达消息。通过数学推导，给出了这种隐写构造的嵌入率期望值。从分布保持的角度，给出了这种隐写构造的安全性证明。通过递归分组的思想显著提升了这种隐写构造的嵌入率，并实现了在文本生成和语音合成这两个任务上的部署。提出了一种全新的容量效率度量指标——对熵的利用率，可以更好地刻画嵌入率与其理论极限之间的距离。实验表明，该隐写构造的嵌入率能接近其理论极限。对比正常生成，其引入的额外时间相对较小，从而确保了该方案在实际应用中的高效性。

2. 提出了一种基于深度生成模型和算术编码的文本蜜罐加密方案

探讨了生成式隐写与蜜罐加密之间的关系，并尝试将生成式可证安全隐写领域的一些思想应用于文本蜜罐加密系统的设计中。从减小 DTE 的建模损失和编码损失着手，设计了一种基于深度生成模型和算术编码的 DTE：首先对明文进行分词，然后使用算术编码根据深度生成模型给出的概率分布将明文编码为种子。为了消除可区分特征，在明文和种子两个层面上设计了伪随机填充方案。实验表明，所提出的方案压缩率较高、编码损失较小，在使用 GPU 加速时速度较快，在使用规模较大的模型时，建模损失较小、安全性较高。

5.2　未来工作展望

这里将简要描述本文的改进空间及潜在的发展方向。

1. 基于分布副本的隐写构造需要通信双方共享驱动伪随机数生成器（PRNG）的种子，因此目前仅适用于对称密钥隐写场景，尚无法简单地直接扩展到公钥隐写场景。如何将这种隐写构造应用于公钥隐写场景，将是一个有趣的方向。

2. 基于分布副本的隐写构造目前的部署场景局限于显式生成模型，如自回归模型。尽管这种模型在文本生成任务中广泛使用，但在图像生成、语音合成等领域上并不是主导性模型。如何使这种隐写构造兼容隐式生成模型，也是一个值得探索的方向。
3. 目前所有追求可证安全的隐写构造都不具有鲁棒性，这意味着它们要求载密在无损信道中进行传输。尽管文本在传输过程中通常不会遭受有损处理，但是图像、语音等数据可能受到压缩、重编码等有损操作。研究鲁棒的可证安全隐写，有助于拓宽可证安全隐写的应用范围，也是一个重要的方向。
4. 由于深度生成模型具有较高的不可解释性，我们无法以比较严谨的方式证明深度生成模型的潜在分布与明文实际分布足够接近。在文本蜜罐加密方案上，结合一些密码学理论分析以及 AI 的可解释性研究，或许是一个可行方向。

参 考 文 献

[1] FRIDRICH J. Steganography in digital media: Principles, algorithms, and applications[M]. Cambridge: Cambridge University Press, 2009.

[2] FRIDRICH J. 数字媒体中的隐写术：原理，算法和应用[M]. 许漫坤, 译. 国防工业出版社, 2014.

[3] JUELS A, RISTENPART T. Honey encryption: Security beyond the brute-force bound [C/OL]//NGUYEN P Q, OSWALD E. Lecture Notes in Computer Science: Advances in Cryptology – EUROCRYPT 2014. Berlin, Heidelberg: Springer, 2014: 293-310. DOI: 10.1007/978-3-642-55220-5_17.

[4] LOW S H, MAXEMCHUK N F, BRASSIL J T, et al. Document marking and identification using both line and word shifting[C/OL]//Proceedings of INFOCOM'95: Vol. 2. 1995: 853-860 vol.2. DOI: 10.1109/INFCOM.1995.515956.

[5] LOW S H, MAXEMCHUK N F, LAPONE A M. Document identification for copyright protection using centroid detection[J/OL]. IEEE Transactions on Communications, 1998, 46 (3): 372-383. DOI: 10.1109/26.662643.

[6] LOW S H, MAXEMCHUK N F. Performance comparison of two text marking methods [J/OL]. IEEE Journal on Selected Areas in Communications, 1998, 16(4): 561-572. DOI: 10.1109/49.668978.

[7] BRASSIL J T, LOW S, MAXEMCHUK N F, et al. Electronic marking and identification techniques to discourage document copying[J/OL]. IEEE Journal on Selected Areas in Communications, 1995, 13(8): 1495-1504. DOI: 10.1109/49.464718.

[8] BRASSIL J T, LOW S, MAXEMCHUK N F. Copyright protection for the electronic distribution of text documents[J/OL]. Proceedings of the IEEE, 1999, 87(7): 1181-1196. DOI: 10.1109/5.771071.

[9] TAN L, SUN X, SUN G. Print-scan resilient text image watermarking based on stroke direction modulation for chinese document authentication: Vol. 21[Z]. 2012: 12.

[10] RIZZO S G, BERTINI F, MONTESI D. Content-preserving text watermarking through unicode homoglyph substitution[C/OL]//IDEAS '16: Proceedings of the 20th International Database Engineering & Applications Symposium. New York, NY, USA: Association for Computing Machinery, 2016: 97-104. DOI: 10.1145/2938503.2938510.

[11] 张卫明, 王宏霞, 李斌, 等. 多媒体隐写研究进展[J/OL]. 中国图象图形学报, 2022. http://www.cjig.cn/jig/article/html/20220610.

[12] TOPKARA M, RICCARDI G, Hakkani-Tür D, et al. Natural language watermarking: Chal-

lenges in building a practical system[C/OL]//Security, Steganography, and Watermarking of Multimedia Contents VIII: Vol. 6072. International Society for Optics and Photonics, 2006: 60720A. DOI: 10.1117/12.643560.

[13] DAI Z X, HONG F, CUI G H, et al. Watermarking text document based on statistic property of part of speech string[J]. Tongxin Xuebao/Journal on Communications, 2007, 28: 108-113.

[14] MURPHY B, VOGEL C. Statistically constrained shallow text marking: Techniques, evaluation paradigm, and results[C/OL]//Security, Steganography, and Watermarking of Multimedia Contents IX: Vol. 6505. International Society for Optics and Photonics, 2007: 65050Z. DOI: 10.1117/12.713355.

[15] MURPHY B, VOGEL C. The syntax of concealment: Reliable methods for plain text information hiding[C/OL]//Security, Steganography, and Watermarking of Multimedia Contents IX: Vol. 6505. International Society for Optics and Photonics, 2007: 65050Y. DOI: 10.1117/12.713357.

[16] WANG H, XINGMING S, LIU Y, et al. Natural language watermarking using chinese syntactic transformations[J/OL]. Information Technology Journal, 2008, 7. DOI: 10.3923/itj.2008.904.910.

[17] ATALLAH M J, RASKIN V, CROGAN M, et al. Natural language watermarking: Design, analysis, and a proof-of-concept implementation[C/OL]//MOSKOWITZ I S. Lecture Notes in Computer Science: Information Hiding. Berlin, Heidelberg: Springer, 2001: 185-200. DOI: 10.1007/3-540-45496-9_14.

[18] Shirali-Shahreza M H, Shirali-Shahreza M. A new synonym text steganography[C/OL]//2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing. 2008: 1524-1526. DOI: 10.1109/IIH-MSP.2008.6.

[19] MUHAMMAD H Z, RAHMAN S M S A A, SHAKIL A. Synonym based malay linguistic text steganography[C/OL]//2009 Innovative Technologies in Intelligent Systems and Industrial Applications. 2009: 423-427. DOI: 10.1109/CITISIA.2009.5224169.

[20] CHANG C Y, CLARK S. Practical linguistic steganography using contextual synonym substitution and vertex colour coding[C]//Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Cambridge, MA: Association for Computational Linguistics, 2010: 1194-1203.

[21] UEOKA H, MURAWAKI Y, KUROHASHI S. Frustratingly easy edit-based linguistic steganography with a masked language model[C/OL]//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Online: Association for Computational Linguistics, 2021: 5486-5492. DOI: 10.18653/v1/2021.naacl-main.433.

[22] YANG X, ZHANG J, CHEN K, et al. Tracing text provenance via context-aware lexical substitution[C/OL]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 36. 2022: 11613-11621. DOI: 10.1609/aaai.v36i10.21415.

[23] ZHANG J, SHEN J, WANG L, et al. Coverless text information hiding method based on the word rank map[C/OL]//Cloud Computing and Security. Springer, Cham, 2016: 145-155. DOI: 10.1007/978-3-319-48671-0_14.

[24] ZHOU Z, MU Y, ZHAO N, et al. Coverless information hiding method based on multi-keywords[C/OL]//SUN X, LIU A, CHAO H C, et al. Lecture Notes in Computer Science: Cloud Computing and Security. Cham: Springer International Publishing, 2016: 39-47. DOI: 10.1007/978-3-319-48671-0_4.

[25] ZHOU Z, MU Y, WU Q M J. Coverless image steganography using partial-duplicate image retrieval[J/OL]. Soft Computing, 2019, 23(13): 4927-4938. DOI: 10.1007/s00500-018-3151-8.

[26] WAYNER P. Mimic functions[J/OL]. Cryptologia, 1992, 16(3): 193-214. DOI: 10.1080/0161-119291866883.

[27] CHAPMAN M, DAVIDA G. Hiding the hidden: A software system for concealing ciphertext as innocuous text[C/OL]//HAN Y, OKAMOTO T, QING S. Lecture Notes in Computer Science: Information and Communications Security. Berlin, Heidelberg: Springer, 1997: 335-345. DOI: 10.1007/BFb0028489.

[28] DAI W, YU Y, DAI Y, et al. Text steganography system using markov chain source model and des algorithm[J/OL]. JSW, 2010, 5: 785-792. DOI: 10.4304/jsw.5.7.785-792.

[29] MORALDO H. An approach for text steganography based on markov chains[Z]. 2014.

[30] LUO Y, HUANG Y, LI F, et al. Text steganography based on ci-poetry generation using markov chain model[J/OL]. KSII Transactions on Internet and Information Systems, 2016, 10: 4568-4584. DOI: 10.3837/tiis.2016.09.029.

[31] SHNIPEROV A N, NIKITINA K A. A text steganography method based on markov chains [J/OL]. Automatic Control and Computer Sciences, 2016, 50(8): 802-808. DOI: 10.3103/S0146411616080174.

[32] FANG T, JAGGI M, ARGYRAKI K. Generating steganographic text with lstms[C]// Proceedings of ACL 2017, Student Research Workshop. Vancouver, Canada: Association for Computational Linguistics, 2017: 100-106.

[33] YANG Z, GUO X, CHEN Z, et al. Rnn-stega: Linguistic steganography based on recurrent neural networks[J/OL]. IEEE Transactions on Information Forensics and Security, 2019, 14 (5): 1280-1295. https://doi.org/10.1109/TIFS.2018.2871746.

[34] DAI F, CAI Z. Towards near-imperceptible steganographic text[C/OL]//Proceedings of the

57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 4303-4308. DOI: 10.18653/v1/P19-1422.

[35] ZIEGLER Z, DENG Y, RUSH A. Neural linguistic steganography[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 1210-1215. DOI: 10.18653/v1/D19-1115.

[36] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners [Z]. 2019: 24.

[37] SHEN J, JI H, HAN J. Near-imperceptible neural linguistic steganography via self-adjusting arithmetic coding[C/OL]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, 2020: 303-313. DOI: 10.18653/v1/2020.emnlp-main.22.

[38] KAPTCHUK G, JOIS T M, GREEN M, et al. Meteor: Cryptographically secure steganography for realistic distributions[C/OL]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual Event Republic of Korea: ACM, 2021: 1529-1548. DOI: 10.1145/3460120.3484550.

[39] ZHANG S, YANG Z, YANG J, et al. Provably secure generative linguistic steganography [C/OL]//Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Online: Association for Computational Linguistics, 2021: 3046-3055. DOI: 10.18653/v1/2021.findings-acl.268.

[40] YANG Z L, ZHANG S Y, HU Y T, et al. Vae-stega: Linguistic steganography based on variational auto-encoder[J/OL]. IEEE Transactions on Information Forensics and Security, 2020, 16: 880-895. DOI: 10.1109/TIFS.2020.3023279.

[41] YANG Z, GONG B, LI Y, et al. Graph-stega: Semantic controllable steganographic text generation guided by knowledge graph[EB/OL]. 2020. DOI: 10.13140/RG.2.2.30301.44001.

[42] YANG Z, XIANG L, ZHANG S, et al. Linguistic generative steganography with enhanced cognitive-imperceptibility[J/OL]. IEEE Signal Processing Letters, 2021, PP: 1-1. DOI: 10.1109/LSP.2021.3058889.

[43] CHATTERJEE R, BONNEAU J, JUELS A, et al. Cracking-resistant password vaults using natural language encoders[C/OL]//2015 IEEE Symposium on Security and Privacy. 2015: 481-498. DOI: 10.1109/SP.2015.36.

[44] GOLLA M, BEUSCHER B, DÜRMUTH M. On the security of cracking-resistant password vaults[C/OL]//CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: Association for Computing Machinery,

2016: 1230-1241. DOI: 10.1145/2976749.2978416.

[45] KIM J I, YOON J W. Honey chatting: A novel instant messaging system robust to eavesdropping over communication[C/OL]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2016: 2184-2188. DOI: 10.1109/ICASSP.2016.7472064.

[46] BEUNARDEAU M, FERRADI H, GÉRAUD R, et al. Honey encryption for language [C/OL]//PHAN R C W, YUNG M. Lecture Notes in Computer Science: Paradigms in Cryptology – Mycrypt 2016. Malicious and Exploratory Cryptology. Cham: Springer International Publishing, 2017: 127-144. DOI: 10.1007/978-3-319-61273-7_7.

[47] CHENG H, ZHENG Z, LI W, et al. Probability model transforming encoders against encoding attacks[C]//28th USENIX Security Symposium (USENIX Security 19). 2019: 1573-1590.

[48] SIMMONS G J. The prisoners' problem and the subliminal channel[C]//Advances in Cryptology. Springer, 1984: 51-67.

[49] KERCKHOFFS A. La cryptographic militaire[J]. Journal des sciences militaires, 1883: 5-38.

[50] HOPPER N J, LANGFORD J, von Ahn L. Provably secure steganography[C/OL]//YUNG M. Advances in Cryptology — CRYPTO 2002. Berlin, Heidelberg: Springer, 2002: 77-92. DOI: 10.1007/3-540-45708-9_6.

[51] 钱振兴, 张卫明, 卢伟, 等. 多媒体安全基础导论[M]. 复旦大学出版社, 2022.

[52] WESTFELD A, PFITZMANN A. Attacks on steganographic systems[C/OL]//PFITZMANN A. Lecture Notes in Computer Science: Information Hiding. Berlin, Heidelberg: Springer, 2000: 61-76. DOI: 10.1007/10719724_5.

[53] FRIDRICH J J, GOLJAN M, DU R. Detecting lsb steganography in color and gray-scale images[J/OL]. IEEE MultiMedia, 2001, 8(4): 22-28. DOI: 10.1109/93.959097.

[54] SHARP T. An implementation of key-based digital signal steganography[C/OL]//MOSKOWITZ I S. Lecture Notes in Computer Science: Information Hiding. Berlin, Heidelberg: Springer, 2001: 13-26. DOI: 10.1007/3-540-45496-9_2.

[55] MIELIKAINEN J. Lsb matching revisited[J/OL]. IEEE Signal Processing Letters, 2006, 13 (5): 285-287. DOI: 10.1109/LSP.2006.870357.

[56] ZHANG X, WANG S. Efficient steganographic embedding by exploiting modification direction[J/OL]. IEEE Communications Letters, 2006, 10(11): 781-783. DOI: 10.1109/LCOMM.2006.060863.

[57] FRIDRICH J, LISONEK P. Grid colorings in steganography[J/OL]. IEEE Transactions on Information Theory, 2007, 53(4): 1547-1549. DOI: 10.1109/TIT.2007.892768.

[58] CRANDALL R. Some notes on steganography[EB/OL]. 1998. https://dde.binghamton.edu/download/Crandall_matrix.pdf.

[59] WESTFELD A. F5—a steganographic algorithm[C/OL]//MOSKOWITZ I S. Information

Hiding. Berlin, Heidelberg: Springer, 2001: 289-302. DOI: 10.1007/3-540-45496-9_21.

[60] FRIDRICH J, SOUKAL D. Matrix embedding for large payloads[J/OL]. IEEE Transactions on Information Forensics and Security, 2006, 1(3): 390-395. DOI: 10.1109/TIFS.2006.879281.

[61] FRIDRICH J, FILLER T. Practical methods for minimizing embedding impact in steganography[C/OL]//Security, Steganography, and Watermarking of Multimedia Contents IX: Vol. 6505. SPIE, 2007: 13-27. DOI: 10.1117/12.697471.

[62] ZHANG W, ZHANG X, WANG S. Maximizing steganographic embedding efficiency by combining hamming codes and wet paper codes[C/OL]//SOLANKI K, SULLIVAN K, MADHOW U. Information Hiding. Berlin, Heidelberg: Springer, 2008: 60-71. DOI: 10.1007/978-3-540-88961-8_5.

[63] FILLER T, JUDAS J, FRIDRICH J. Minimizing embedding impact in steganography using trellis-coded quantization[C/OL]//Media Forensics and Security II: Vol. 7541. SPIE, 2010: 38-51. DOI: 10.1117/12.838002.

[64] FILLER T, JUDAS J, FRIDRICH J. Minimizing additive distortion in steganography using syndrome-trellis codes[J/OL]. IEEE Transactions on Information Forensics and Security, 2011, 6(3): 920-935. DOI: 10.1109/TIFS.2011.2134094.

[65] LI W, ZHANG W, LI L, et al. Designing near-optimal steganographic codes in practice based on polar codes[J/OL]. IEEE Transactions on Communications, 2020, 68(7): 3948-3962. DOI: 10.1109/TCOMM.2020.2982624.

[66] YAO Q, ZHANG W, CHEN K, et al. Ldgm codes based near-optimal coding for adaptive steganography[J/OL]. IEEE Transactions on Communications, 2023: 1-1. DOI: 10.1109/TCOMM.2023.3342247.

[67] PEVNÝ T, FILLER T, BAS P. Using high-dimensional image models to perform highly undetectable steganography[C/OL]//BÖHME R, FONG P W L, Safavi-Naini R. Lecture Notes in Computer Science: Information Hiding. Berlin, Heidelberg: Springer, 2010: 161-177. DOI: 10.1007/978-3-642-16435-4_13.

[68] PEVNÝ T, BAS P, FRIDRICH J. Steganalysis by subtractive pixel adjacency matrix[J/OL]. IEEE Transactions on Information Forensics and Security, 2010, 5(2): 215-224. DOI: 10.1109/TIFS.2010.2045842.

[69] HOLUB V, FRIDRICH J. Designing steganographic distortion using directional filters [C/OL]//2012 IEEE International Workshop on Information Forensics and Security (WIFS). 2012: 234-239. DOI: 10.1109/WIFS.2012.6412655.

[70] HOLUB V, FRIDRICH J. Digital image steganography using universal distortion[C/OL]//IH&MMSec '13: Proceedings of the First ACM Workshop on Information Hiding and

Multimedia Security. New York, NY, USA: Association for Computing Machinery, 2013: 59-68. DOI: 10.1145/2482513.2482514.

[71] FRIDRICH J, KODOVSKY J. Rich models for steganalysis of digital images[J/OL]. IEEE Transactions on Information Forensics and Security, 2012, 7(3): 868-882. DOI: 10.1109/TIFS.2012.2190402.

[72] LI B, TAN S, WANG M, et al. Investigation on cost assignment in spatial image steganography[J/OL]. IEEE Transactions on Information Forensics and Security, 2014, 9(8): 1264-1277. DOI: 10.1109/TIFS.2014.2326954.

[73] LI B, WANG M, HUANG J, et al. A new cost function for spatial image steganography [C/OL]//2014 IEEE International Conference on Image Processing (ICIP). 2014: 4206-4210. DOI: 10.1109/ICIP.2014.7025854.

[74] FRIDRICH J, KODOVSKÝ J. Multivariate gaussian model for designing additive distortion for steganography[C/OL]//2013 IEEE International Conference on Acoustics, Speech and Signal Processing. 2013: 2949-2953. DOI: 10.1109/ICASSP.2013.6638198.

[75] SEDIGHI V, COGRANNE R, FRIDRICH J. Content-adaptive steganography by minimizing statistical detectability[J/OL]. IEEE Transactions on Information Forensics and Security, 2016, 11(2): 221-234. DOI: 10.1109/TIFS.2015.2486744.

[76] LI B, WANG M, LI X, et al. A strategy of clustering modification directions in spatial image steganography[J/OL]. IEEE Transactions on Information Forensics and Security, 2015, 10 (9): 1905-1917. DOI: 10.1109/TIFS.2015.2434600.

[77] DENEMARK T, FRIDRICH J. Improving steganographic security by synchronizing the selection channel[C/OL]//IH&MMSec '15: Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security. New York, NY, USA: Association for Computing Machinery, 2015: 5-14. DOI: 10.1145/2756601.2756620.

[78] ZHANG W, ZHANG Z, ZHANG L, et al. Decomposing joint distortion for adaptive steganography[J/OL]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 27(10): 2274-2280. DOI: 10.1109/TCSVT.2016.2587388.

[79] UPHAM D. JSteg[EB/OL]. [2024-03-10]. https://zooid.org/~paul/crypto/jsteg/.

[80] PROVOS N. Defending against statistical steganalysis[C]//10th USENIX Security Symposium (USENIX Security 01). 2001.

[81] SALLEE P. Model-based steganography[C/OL]//KALKER T, COX I, RO Y M. Digital Watermarking. Berlin, Heidelberg: Springer, 2004: 154-167. DOI: 10.1007/978-3-540-24624-4_12.

[82] HETZL S, MUTZEL P. A graph–theoretic approach to steganography[C/OL]//DITTMANN J, KATZENBEISSER S, UHL A. Lecture Notes in Computer Science: Communications and

Multimedia Security. Berlin, Heidelberg: Springer, 2005: 119-128. DOI: 10.1007/11552055_12.

[83] GUO L, NI J, SHI Y Q. An efficient jpeg steganographic scheme using uniform embedding [C/OL]//2012 IEEE International Workshop on Information Forensics and Security (WIFS). 2012: 169-174. DOI: 10.1109/WIFS.2012.6412644.

[84] GUO L, NI J, SU W, et al. Using statistical image model for jpeg steganography: Uniform embedding revisited[J/OL]. IEEE Transactions on Information Forensics and Security, 2015, 10(12): 2669-2680. DOI: 10.1109/TIFS.2015.2473815.

[85] LI W, ZHANG W, CHEN K, et al. Defining joint distortion for jpeg steganography[C/OL]//IH&MMSec '18: Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security. New York, NY, USA: Association for Computing Machinery, 2018: 5-16. DOI: 10.1145/3206004.3206008.

[86] 张卫明, 陈可江, 俞能海. 可证安全隐写：理论、应用与展望[J/OL]. 网络空间安全科学学报, 2023, 1(1): 38-46. http://www.journalofcybersec.com/CN/Y2023/V1/I1/38.

[87] SHANNON C E. Communication theory of secrecy systems[J/OL]. The Bell System Technical Journal, 1949, 28(4): 656-715. DOI: 10.1002/j.1538-7305.1949.tb00928.x.

[88] CACHIN C. An information-theoretic model for steganography[C/OL]//AUCSMITH D. Information Hiding. Berlin, Heidelberg: Springer, 1998: 306-318. DOI: 10.1007/3-540-49380-8_21.

[89] KATZENBEISSER S, PETITCOLAS F A P. Defining security in steganographic systems [C/OL]//DELP III E J, WONG P W. Security and Watermarking of Multimedia Contents IV. San Jose, CA, 2002: 50-56. http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=877726. DOI: 10.1117/12.465313.

[90] AHN L V, HOPPER N J. Public-key steganography[C]//International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2004: 323-341.

[91] BACKES M, CACHIN C. Public-key steganography with active attacks[C/OL]//KILIAN J. Lecture Notes in Computer Science: Theory of Cryptography. Berlin, Heidelberg: Springer, 2005: 210-226. DOI: 10.1007/978-3-540-30576-7_12.

[92] LE T V. Efficient provably secure public key steganography: 2003/156[Z]. 2003.

[93] SHANNON C E. A mathematical theory of communication[J/OL]. The Bell System Technical Journal, 1948, 27(3): 379-423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[94] GOODFELLOW I, Pouget-Abadie J, MIRZA M, et al. Generative adversarial nets[C]//Advances in Neural Information Processing Systems: Vol. 27. Cambridge, MA, USA: Curran Associates, Inc., 2014: 2672-2680.

[95] ZHU J Y, PARK T, ISOLA P, et al. Unpaired image-to-image translation using cycle-

consistent adversarial networks[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 2223-2232.

[96] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[C]// Advances in Neural Information Processing Systems: Vol. 33. Online: Curran Associates, Inc., 2020: 1877-1901.

[97] PATASHNIK O, WU Z, SHECHTMAN E, et al. Styleclip: Text-driven manipulation of stylegan imagery[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, QC, Canada: IEEE, 2021: 2085-2094.

[98] KINGMA D P, WELLING M. Auto-encoding variational bayes[C]//Proceedings of the 2th International Conference on Learning Representations. Banff, AB, Canada, 2014.

[99] KINGMA D P, WELLING M, et al. An introduction to variational autoencoders[J]. Foundations and Trends® in Machine Learning, 2019, 12(4): 307-392.

[100] van den Oord A, KALCHBRENNER N, ESPEHOLT L, et al. Conditional image generation with pixelcnn decoders[C]//Advances in Neural Information Processing Systems: Vol. 29. Curran Associates, Inc., 2016.

[101] RADFORD A, NARASIMHAN K. Improving language understanding by generative pre-training[Z]. 2018.

[102] OUYANG L, WU J, JIANG X, et al. Training language models to follow instructions with human feedback[C]//KOYEJO S, MOHAMED S, AGARWAL A, et al. Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022. 2022.

[103] Sohl-Dickstein J, WEISS E, MAHESWARANATHAN N, et al. Deep unsupervised learning using nonequilibrium thermodynamics[C]//Proceedings of the 32nd International Conference on Machine Learning. PMLR, 2015: 2256-2265.

[104] SONG Y, ERMON S. Generative modeling by estimating gradients of the data distribution [C]//Advances in Neural Information Processing Systems: Vol. 32. Curran Associates, Inc., 2019.

[105] HO J, JAIN A, ABBEEL P. Denoising diffusion probabilistic models[C]//Advances in Neural Information Processing Systems: Vol. 33. Curran Associates, Inc., 2020: 6840-6851.

[106] RAMESH A, DHARIWAL P, NICHOL A, et al. Hierarchical text-conditional image generation with clip latents[A]. 2022. arxiv: 2204.06125.

[107] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[C/OL]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technolo-

gies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 4171-4186. DOI: 10.18653/v1/N19-1423.

[108] RAMESH A, PAVLOV M, GOH G, et al. Zero-shot text-to-image generation[C]// Proceedings of the 38th International Conference on Machine Learning. PMLR, 2021: 8821-8831.

[109] PENNISI E, STOKSTAD E, GIBBONS A, et al. Breakthrough of the year: Runners-up [J/OL]. Science, 2022, 378(6625): 1162-1167. DOI: 10.1126/science.adg2799.

[110] GARTNER. Top strategic technology trends for 2022[J]. Gartner, 2022.

[111] YANG K, CHEN K, ZHANG W, et al. Provably secure generative steganography based on autoregressive model[C/OL]//YOO C D, SHI Y Q, KIM H, et al. Lecture Notes in Computer Science: Vol. 11378 Digital Forensics and Watermarking - 17th International Workshop, IWDW 2018, Jeju Island, Korea, October 22-24, 2018, Proceedings. Springer, 2018: 55-68. https://doi.org/10.1007/978-3-030-11389-6_5.

[112] CHEN K, ZHOU H, ZHAO H, et al. Distribution-preserving steganography based on text-to-speech generative models[J/OL]. IEEE Transactions on Dependable and Secure Computing, 2021: 3343-3356. DOI: 10.1109/TDSC.2021.3095072.

[113] HUANG Z, AYDAY E, FELLAY J, et al. Genoguard: Protecting genomic data against brute-force attacks[C/OL]//2015 IEEE Symposium on Security and Privacy. 2015: 447-462. DOI: 10.1109/SP.2015.34.

[114] KATZ J, LINDELL Y. 现代密码学：原理与协议[M]. 任伟, 译. 国防工业出版社, 2011.

[115] KATZ J, LINDELL Y. Introduction to modern cryptography, second edition[M]. CRC Press, 2014.

[116] HOLTZMAN A, BUYS J, DU L, et al. The curious case of neural text degeneration[C]//8th International Conference on Learning Representations. Addis Ababa, Ethiopia,: OpenReview.net, 2020.

[117] MAAS A L, DALY R E, PHAM P T, et al. Learning word vectors for sentiment analysis[C/OL]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics, 2011: 142-150. http://www.aclweb.org/anthology/P11-1015.

[118] LHOEST Q, VILLANOVA DEL MORAL A, JERNITE Y, et al. Datasets: A community library for natural language processing[C/OL]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021: 175-184. https://aclanthology.org/2021.emnlp-demo.21.

[119] YANG Z, HUANG Y, ZHANG Y J. A fast and efficient text steganalysis method[J]. IEEE

Signal Processing Letters, 2019, 26(4): 627-631.

[120] NIU Y, WEN J, ZHONG P, et al. A hybrid r-bilstm-c neural network based text steganalysis [J/OL]. IEEE Signal Processing Letters, 2019, 26(12): 1907-1911. DOI: 10.1109/LSP.2019.2953953.

[121] YANG H, BAO Y, YANG Z, et al. Linguistic steganalysis via densely connected lstm with feature pyramid[C/OL]//Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security. Denver CO USA: ACM, 2020: 5-10. DOI: 10.1145/3369412.3395067.

[122] LUO W, LI H, YAN Q, et al. Improved audio steganalytic feature and its applications in audio forensics[J]. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2018, 14(2): 1-14.

[123] RIX A W, BEERENDS J G, HOLLIER M P, et al. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs[C/OL]//IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2001, 7-11 May, 2001, Salt Palace Convention Center, Salt Lake City, Utah, USA, Proceedings. 2001: 749-752. https://doi.org/10.1109/ICASSP.2001.941023.

[124] ZHANG X, CHEN K, DING J, et al. Provably secure public-key steganography based on elliptic curve cryptography[J/OL]. IEEE Transactions on Information Forensics and Security, 2024, 19: 3148-3163. DOI: 10.1109/TIFS.2024.3361219.

[125] GAO L, BIDERMAN S, BLACK S, et al. The pile: An 800gb dataset of diverse text for language modeling: arXiv:2101.00027[M/OL]. arXiv, 2020. DOI: 10.48550/arXiv.2101.00027.

[126] TOUVRON H, MARTIN L, STONE K, et al. Llama 2: Open foundation and fine-tuned chat models: arXiv:2307.09288[M/OL]. arXiv, 2023. DOI: 10.48550/arXiv.2307.09288.

[127] SANH V, DEBUT L, CHAUMOND J, et al. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter[C]//NeurIPS EMC^2 Workshop. 2019.

[128] BAO G, ZHAO Y, TENG Z, et al. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature[C]//The Twelfth International Conference on Learning Representations. 2023.

在读期间发表的学术论文与取得的研究成果

已发表或已录用论文

1. **Jinyang Ding**, Kejiang Chen, Yaofei Wang, Na Zhao, Weiming Zhang, and Nenghai Yu, "Discop: Provably Secure Steganography in Practice Based on 'Distribution Copies'," in *2023 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, May 2023, pp. 2238-2255.
2. Xin Zhang, Kejiang Chen, **Jinyang Ding**, Yuqi Yang, Weiming Zhang, and Nenghai Yu, "Provably Secure Public-Key Steganography Based on Elliptic Curve Cryptography," in *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 19, pp. 3148-3163, 2024.
3. Xilong Wang, Yaofei Wang, Kejiang Chen, **Jinyang Ding**, Weiming Zhang, and Nenghai Yu, "ICStega: Image Captioning-based Semantically Controllable Linguistic Steganography," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, June 2023.
4. 张晏铭, 陈可江, **丁锦扬**, 张卫明, 俞能海. 利用跨模态信息检索的鲁棒隐蔽通信. 中国图象图形学报, 29(02): 0369-0381.