arXiv:2602.04052v1 [math.CO] 3 Feb 2026

通过对称数值半群实现的隐写信息隐藏

Jean-Christophe Pain[a,b,1]

[a]*CEA, DAM, DIF,F-91297* 阿尔帕容, 法国[b]巴黎萨克雷大学, *CEA,* 极端条件物质实验室, *91680* 布吕耶尔勒沙泰勒, 法国

摘要

我们提出一种基于弗罗贝尼乌斯硬币问题所产生的数值半群结构性质的隐写信息隐藏方案。该协议并非通过可表示整数编码数据，而是将信息嵌入精心选取的对称数值半群的间隙结构中。对称性保证了均衡的间隙密度，从而使编码值对缺乏私有生成集的观察者而言，在统计上与均匀数值噪声不可区分。该方案的安全性依赖于隐藏生成元情形下数值半群成员推断的假定平均情况困难性，从而为隐蔽通信和抗后量子信息隐藏提供一种新颖的数论原语。

1 引言

弗罗贝尼乌斯“硬币问题”（又称“货币兑换问题”）源于加性数论中一个看似简单的问题：给定一组互素的正整数 $A = \{a_1, a_2, \ldots, a_n\}$，无法将其表示为这些数的非负整数线性组合的最大整数 $F(A)$ 是多少？尽管求出这个值（即 Frobenius数）是 19 世纪数学家 Sylvester [28]和弗罗贝尼乌斯 [8], 关注的重点，但现代研究从根本上与数值半群理论交织在一起 [14]。

在这一代数背景下，数值半群 S 是一个关于加法的交换幺半群，由除有限多个遗漏值（即“间隙”）$\mathbb{N} \setminus S$之外的所有非负整数组成。Frobenius数 $F(S)$ 因而是这组间隙中的最大值，它标志着一个阈值，通常称为导子减一；超过这一阈值的所有整数都可以表示出来。二十世纪下半叶，人们对这些结构的兴趣显著复苏，这主要得益于它们在代数几何和交换代数中的重要应用 [25]。

该问题的计算复杂性反映了其结构深度；在对生成集不加限制的情况下求Frobenius数已知是NP困难的 [21, 22]。目前尚无已知的多项式时间算法能在没有额外结构信息的情况下从观测到的间隙恢复生成集，这凸显了其在信息隐藏应用中潜在的困难性。因此，闭式解备受重视，且通常仅限于特定情形。除Sylvester关于两个生成元的经典结果（$n = 2$，例如可参见文献 [13] 中的归纳证明，或附录A中的极短证明）之外，格林伯格 [9,10], 针对三个生成元的情形发展了高效算法，而罗伯茨 [23, 24] 与卢因 [16, 17] 则分别给出了等差数列与近等差数列的基础公式。值得一提的是，宾纳 [2]最近提出了Sylvester公式的一种推广。Frobenius数的一般性质见附录B。

对于由 a，d，w 定义的等差数列中的一个集合，若最大公约数$(a,\ d) = 1$，则 Frobenius数为：

$$F(a, a+d, a+2d, \ldots, a+wd) = \left(\left\lfloor \frac{a-2}{w} \right\rfloor \right) a + d(a-1). \tag{1}$$

[1]jean-christophe.pain@cea.fr

有趣的是，这个数的稳定性非常显著。例如，当 $a > w^2 - 3w + 1$ 时，从数列中移除特定的中间元素不会改变 Frobenius数。这一现象使分析重点从单纯计算 $F(A)$ 转向利用其未包含的整数的分布：间隙。

本文结构如下。第2节建立数学框架，特别着重讨论对称数值半群及其间隙结构。第3节介绍隐写协议，包括密钥生成，以及基于模间隙划分的编码策略。同一节还介绍解密算法及其通过剩余图技术实现的高效方法。第4节讨论该方案的安全性，阐明其背后的困难性假设，并概述其局限性。最后，附录C提供实现细节，并给出所提构造的原型。

2 数学预备知识

2.1 数值半群与间隙

本节回顾全文所需的数值半群理论中的主要概念 [12] ，并介绍编码与解码过程所依托的结构性工具。数值半群 $\mathcal{S}$ 是 $(\mathbb{N}, +)$ 的一个子幺半群，满足以下性质：

- $0 \in \mathcal{S}$,
- $\mathcal{S} + \mathcal{S} \subset \mathcal{S}$,
- $\mathbb{N} \setminus \mathcal{S}$ 是有限的。

等价地，数值半群是 $\mathbb{N}$ 的有限生成加法子幺半群，其生成元的最大公约数为 1。若 $A = \{a_1, \ldots, a_n\}$ 是这样一个生成集，我们记作 $\mathcal{S} = \langle A \rangle$。补集 $\mathcal{F}(\mathcal{S})$

$$= \mathbb{N} \setminus \mathcal{S} \text{称为间隙集} \quad .\ \text{其基数}$$

$$g(\mathcal{S}) = |\mathcal{F}(\mathcal{S})|$$

是该半群的亏格，而最大的间隙称为 Frobenius数，记作 $F(\mathcal{S})$。所有大于 $F(\mathcal{S})$ 的整数都属于 $\mathcal{S}$。Frobenius数以下间隙的分布编码了该半群的大部分算术结构，也是所提出协议所利用的核心对象。

设 $m = \min(\mathcal{S} \setminus \{0\})$表示 $\mathcal{S}$的重数。关于 m 的 $\mathcal{S}$ 的Apéry集定义为 as

$$\mathrm{Ap}(\mathcal{S}, m) = \{s \in \mathcal{S} \mid s - m \notin \mathcal{S}\}.$$

Apéry集恰好包含 m 个元素，每个模 m的剩余类中各有一个，并且能够完整描述该半群：整数 x 属于 $\mathcal{S}$ 当且仅当

$$x \geq \mathrm{Ap}(\mathcal{S}, m)[x \bmod m].$$

这一刻画是后文所述解码算法的基础。实际应用中，可以通过有向图上的最短路径算法计算Apéry集；该图的顶点对应模 m 的剩余类，边的权重则由 $\mathcal{S}$ [11, 20]的生成元确定。

2.2 对称半群与密度

如果对于每个整数 z，数值半群 $\mathcal{S}$ 都满足.....，则称其为对称的。

$$z \in \mathcal{S} \iff F(\mathcal{S}) - z \notin \mathcal{S}.$$

对称性在 Frobenius数以下的可表示整数与间隙之间建立了完美的对偶关系。特别地，对称半群满足恒等式

$$g(\mathcal{S}) = \frac{F(\mathcal{S}) + 1}{2},$$

这意味着区间 $[0, F(\mathcal{S})]$ 中恰有一半的整数是间隙。这种对称性使 $\mathcal{S}$ 中的元素与区间 $[0, F(\mathcal{S})]$ 内的间隙形成完美配对。换言之，这确保了间隙的渐近密度为 $1/2$。这种均衡的间隙密度对于确保隐写不可区分性至关重要，因为它可以防止引入可检测的统计偏差。

若生成序列 $A = (a_1、\ldots\ldots、a_n)$ 对所有 $i > 1$ 都满足以下条件，则称其为望远镜式：

$$\frac{a_i}{\gcd(a_1, \ldots, a_{i-1})} \in \langle a_1, \ldots, a_{i-1} \rangle.$$

由望远镜式序列生成的数值半群已知是对称的，并且具有特别简单的 Apéry集。这些结构性质既能确保合法接收者高效地进行成员资格测试，也能提供良好可控的间隙结构，因此望远镜式半群尤其适用于密码学和隐写构造。

最后，对于对称数值半群，间隙的渐近密度为 $1/2$。将范围限制在模固定整数 M 的剩余类时，只要 M 相对于生成元较小，各类中的间隙分布仍近似均匀。这一观察为编码阶段采用的模间隙划分策略提供了依据：每个间隙都可视为承载 $\log_2 M$ 位信息，从而在嵌入率与数值范围之间实现灵活权衡。

3 协议

该方案利用秘密数值半群中间隙位置的算术结构来编码信息。合法接收方持有定义该半群的生成集，而外部观察者只能看到一串缺乏结构背景的整数。

3.1 密钥生成与望远镜式生成集

私钥由定义数值半群 $\mathcal{S} = \langle A \rangle$ 的生成集 $A = (a_1、\ldots\ldots、a_n)$ 组成。为确保合法接收方能够利用对称性并实现算法可处理性，我们仅采用望远镜式生成集。正如前文所述，望远镜式半群已知具有对称性，这直接保证了统计不可区分性所需的均衡间隙密度。此外，这种结构还可借助剩余图技术实现高效的成员资格测试。

3.2 通过模数间隙划分进行编码

我们不将单个位直接映射为间隙成员资格，而是使用剩余类对间隙集进行划分。对于固定模数 $M = 16$，通过选择一个值来编码一个 4 位半字节 $V \in \{0, \ldots, 15\}$

$$x \in \mathcal{F}(\mathcal{S}) \quad \text{such that} \quad x \equiv V \pmod{M}.$$

这种方法将编码值分散到整个数值范围内，从而避免低熵编码。只要半群是对称的，每个剩余类渐近地包含相同数量的间隙，从而保持均匀性。

3.3 解密算法

解码依赖于合法接收方高效判断某个值是否属于 $\mathcal{S}$ 的能力。为此，可构造一个模最小生成元 a_1 的剩余图。对于模 a_1 的每个剩余类，算法使用最短路径技术计算可表示的最小值。任何小于该阈值的整数必然是间隙。

4 安全性分析

本节从隐写学的角度分析所提出方案的安全性。该方案本身并非旨在提供语义加密，而是作为一个信息隐藏层，其安全性依赖于从未标记的数值数据中推断数值半群结构属性的难度。

4.1 对手模型

我们考虑一个被动对手，它观察由编码器生成的整数流（x_i）$_{i\geq 1}$ 。该对手并不掌握定义该数值半群 $\mathcal{S} = \langle A \rangle$的私有生成集 $A =$（a_1、.....、 a_n），也没有关于其结构的任何辅助信息。

攻击者的目标是区分编码信息的数值与均匀分布的数值噪声，或以不可忽略的优势恢复嵌入载荷。假设攻击者的计算能力有限，可以执行任意多项式时间预计算，但无法进行自适应查询，也无法获取预言机反馈。

4.2 困难性假设

该方案的安全性依赖于所谓“数值半群成员资格推断问题”被认为难以解决：给定一个整数 x 和从未知数值半群$\mathcal{S}$中抽取的样本，在不了解生成基的情况下，判断 x 属于 $\mathcal{S}$ 还是属于其间隙集 $\mathcal{F}$（$\mathcal{S}$）。

尽管数值半群成员资格问题已知在最坏情况下是 NP困难的，但我们并未声称存在正式的最坏情况到平均情况归约。相反，安全性依赖于这样一种平均情况困难性：生成集经选择后不具有超递增性、没有冗余元素，且各元素的数量级相近。目前尚无已知的多项式时间算法，能够在平均情况下根据未标记的间隙观测推断出隐藏的生成集。

4.3 通过对称性实现统计不可区分性

该协议的核心特征之一是使用对称数值半群。对于这类半群，对称关系

$$z \in \mathcal{S} \iff F(\mathcal{S}) - z \notin \mathcal{S}$$

在区间 $[0, F(\mathcal{S})]$ 内建立了可表示整数与间隙之间的完美配对。因此，该区间内恰有一半的整数是间隙。

这种均衡密度意味着，在有界数值窗口内，间隙成员资格的表现近似于无偏的伯努利随机变量。因此，只要 $\mathcal{S}$ 的结构仍然隐藏，在标准的基于频率的统计检验下，编码值在统计上就与均匀分布的数值噪声不可区分。

4.4 对基于格的攻击的抵抗能力

经典的格约化攻击，例如 LLL（Lenstra–Lenstra–Lovász） [15] 或 BKZ（Block Korkine-Zolotarev） [3, 26], ，当底层生成集呈现出明显的结构性偏差时，例如超递增序列或低密度表示，对背包型构造非常有效。

在当前构造中，攻击者接触不到生成元的线性组合，只能接触到经间隙选择掩蔽的孤立成员资格实例。此外，生成元经过选择，既避免了超递增行为，又使其大小之间保持有界比率。这些约束阻止了直接构造适用于约化技术的格基，并显著限制了经典格攻击的适用性。

4.5 加盐与范围扩展

为进一步减少结构泄露，我们对编码值应用加盐机制：

$$x' = x + k \cdot \mathrm{lcm}(a_i, a_j),$$

其中，$(a_i,\ a_j)$ 是一对固定的生成元，k 是一个随机整数。

由于 $\mathrm{lcm}(a_i,\ a_j)$ 属于数值半群 $\mathcal{S}$，加上该数的整数倍不会改变合法接收方对 x 是否为间隙的判断。对于不了解生成元的外部观察者而言，这种变换会扩大可观测数值范围，并破坏余数与间隙位置之间的相关性。

我们强调，本文分析并不构成正式的密码学证明。特别是，我们并未声称能够将其归约到标准的密码学困难性假设，也未考虑自适应或主动对手模型。因此，所提出的方案应被视为一种隐写原语或隐蔽编码层，适合与传统密码加密机制结合使用。

尽管存在这些局限，该构造为信息隐藏引入了加性数论的新颖应用。通过利用对称数值半群及其间隙分布的结构性质，该协议实现了均衡的统计行为，并能抵抗直接的结构推断，这为数值半群理论在安全隐蔽通信中的应用开辟了新方向。

5 结论

本研究表明，数值半群理论为隐写信息隐藏提供了一个前景广阔且相对鲜有探索的框架。通过在对称数值半群的间隙结构中编码数据，所提出的协议实现了统计平衡、结构混淆，并使合法接收方能够高效解码。

尽管本方案并未声称具备经典意义上的形式化密码学安全性，但它为隐蔽通信引入了新颖的数论视角，并揭示了加性组合学与信息安全之间的新联系。未来工作将研究参数选择、经验不可区分性测试，以及抵御先进的基于格的攻击和组合攻击的能力。

A 西尔维斯特公式

方程 $ax + by = N$ 的解具有如下形式：$(x_0 + bt,\ y_0 - at)$，其中 $t \in \mathbb{Z}$，且 $(x_0,\ y_0)$ 是该方程的一个特解。设 $N \geq ab - a - b$，并令 t 为一个整数，使得$\leq y_0 - at \leq a - 1$，则有

$$(x_0 + bt)a = N - (y_0 - at)b > ab - a - b - (a - 1)b = -a,$$

从而得到 $x_0+ bt > -1$，即 $x_0 + bt \geq 0$。因此，方程 $ax + by = N$ 有正整数解 ,
and

$$F(a, b) \geq ab - a - b.$$

如果方程 $ax + by = ab - a - b$ 没有正整数解，那么 $ab = a(x + 1) + b(y + 1)$。由于最大公约数$(a, b) = 1$，我们有 $a|(y + 1)$ 和 $b|(x + 1)$，这意味着 $y + 1 \geq a$ 和 $x + 1 \geq b$。随后，

$$ab = a(x + 1) + b(y + 1) \geq 2ab$$

这是矛盾的，因此

$$F(a, b) \geq ab - a - b,$$

最终得到西尔维斯特公式 $F(a,\ b) = ab - a - b$。

B Frobenius数的特殊情形与渐近行为

B.1 等比数列与 Frobenius数

设正整数 a, b, k，满足最大公约数$(a,\ b) = 1$，并定义等比数列

$$A_k(a,b) = \{a^k, a^{k-1}b, a^{k-2}b^2, \ldots, b^k\}.$$

令 $\mathcal{S} = \langle A_k(a,\ b)\rangle$ 为由该集合生成的数值半群。我们将 $F(\mathcal{S})$ 记为 Frobenius数，即不能表示为 $A_k(a,\ b)$ 中元素的非负线性组合的最大整数；将 $g(\mathcal{S}) = |\mathcal{F}(\mathcal{S})|$ 记为亏格，即 $\mathcal{S}$ 中间隙的个数。为方便起见，我们引入求和记号

$$\sigma_r(a,b) = a^r + a^{r-1}b + a^{r-2}b^2 + \cdots + b^r = \sum_{i=0}^{r} a^{r-i}b^i, \tag{2}$$

即指数为 r [19, 29] 的等比级数中所有项的和。使用这一记号， $A_k(a,\ b)$ 的 Frobenius数由 Tripathi公式 [29] 给出:

$$F(A_k(a,b)) = \sigma_{k+1}(a,b) - \sigma_k(a,b) - \left(a^{k+1} + b^{k+1}\right). \tag{3}$$

例如，当 $k = 2$ 时，我们有

$$\sigma_2(a,b) = a^2 + ab + b^2, \quad \sigma_3(a,b) = a^3 + a^2b + ab^2 + b^3,$$

因此

$$F(A_2(a,b)) = \sigma_3(a,b) - \sigma_2(a,b) - (a^3 + b^3) = a^2b + ab^2 - ab = ab(a + b - 1).$$

如果该半群是对称的，那么亏格为

$$g(\mathcal{S}) = \frac{F(\mathcal{S}) + 1}{2}.$$

对于较大的 $k > 2$, σ_k, $(a,\ b)$ 可用于表示中间和、生成函数，或简化组合推导。需要注意的是， $\sigma_k(a,\ b)$ 本身既不是弗罗贝尼乌斯数，也不是亏格，而是一种便于简洁书写公式的辅助记号。值得一提的是，针对三个数，已有相应公式和快速算法。 [30]

B.2 渐近界与平均行为

当生成元的数量较少时，“硬币问题”的复杂度仍然可控。20 世纪 70 年代末，人们建立了用于处理 n 个生成元的重要算法框架，其中以 Selmer [27] 的工作尤为突出。然而，随着 n 增大，手工计算往往十分繁琐。在此背景下，研究方向转向了紧确的渐近界。Davison 为修正后的弗罗贝尼乌斯数 $f(a_1,\ a_2,\ a_3) = g(a_1,\ a_2,\ a_3) + a_1 + a_2 + a_3$ 建立了一个下界，证明:

$$f(a_1,a_2,a_3) \geq \sqrt{3a_1a_2a_3}. \tag{4}$$

正如 Beck 和 Zacks [1] 所述，这一界限相当精确。关于平均行为，Ustinov [31] 证明，对于三个变量， f 的期望值按以下规律增长:

$$f(a_1,a_2,a_3) \sim \frac{8}{\pi}\sqrt{a_1a_2a_3}. \tag{5}$$

加性组合学中较新的研究表明，对于固定的 d，Frobenius数通常按以下方式缩放

$$g(a_1,\ldots,a_d) \sim \sqrt[d-1]{(d-1)!\prod a_i}.$$

B.3 Wilf猜想

一个由来已久的一个尚未解决的开放问题是 Wilf 提出的猜想 [32], , 该猜想将 Frobenius 数联系起来 r F, 数值半群的 遗漏值（亏格 g）的数量，以及生成元 d 的数量：

$$d \geq \frac{F+1}{F+1-g}. \tag{6}$$

虽然一般情形仍未解决，但研究已取得重要进展。Moscariello 和 Sam- martano [18]验证了这一不等式的渐近版本。近期，人们已证实该猜想适用于亏格较小的数值半群（$g \leq 60$），以及 Apéry集具有特定模式的数值半群 [5]。Eliahou [7] 进一步改进了这些界限，为该猜想适用于嵌入维数较高的半群提供了更有力的证据，并表明可表示整数与不可表示整数之比，从根本上受到生成集密度的制约。

C FrobCrypt，一种 Python 实现

下面这个名为“FrobCrypt”的程序可以用来测试这种方法。FrobCrypt 是一种安全工具，可将你的消息隐藏在一串数字中。对其他人来说，这看起来只是随机数据；而对你来说，这是一条秘密消息。

C.1 程序中的不同步骤

如果密钥是一组“魔法硬币”（例如 5 和 7），那么只用这些硬币，有些金额就“无法”凑出（例如 1、2、3、4、6、8 和 9）。这些金额称为“间隙”。第一步是加密：程序会接收消息，并将其隐藏在这些“无法凑出”的数字中。安全性由这样一个事实来保证：没有密钥（即这些魔法硬币），攻击者就无法分辨普通数字和间隙。最后一步是解密：由于我们掌握密钥，因此可以轻松找出间隙并读取隐藏的消息。

C.2 优势与程序的使用

输出看起来像原始传感器数据或随机噪声。该程序速度非常快，因为它使用了经过优化的数学算法（Dijkstra算法 [4, 6]）来快速加密和解密。它也很安全，因为它基于这样的数学问题：如果没有密钥，计算上被认为难以解决。代码完全使用 Python 编写，而且非常容易使用。步骤如下：

- 创建密钥并加密消息：from FrobCrypt import FrobCipher, generate_secure_k ey.
- 生成新的密钥：my_key = generate_secure_key(5) cipher = FrobCipher(my_key)
- 加密单词：单词 ’BONJOUR’ 将变成一个由秘密数字组成的列表。

包含以上所有步骤的 Python 程序如下：

导入 heapq 导入
随机数 导入
math

class FrobCipher: def __init__(self, key_set): """ 使用密钥（数字集合）初始化系统 """

```
        self.A   =sorted(list(key_set)) self.a1 = self.A[0]# 间
        隙结构的计算 self.n_struct   =self._build_residue_
        graph() self.F   =max(self.n_struct.values()) -
        self.a1

    def _build_residue_graph(self): """ 计算最短路径（迪杰斯特拉算法）以找出
    间隙 """ n = {r:float('inf') for r in range(self.a1)} n[0] = 0queue = [(0,
    0)]while queue: d, r = heapq.heappop(queue) if d > n[r]: continue for
    x inself.A: nv, nr =d + x, (d + x) % self.a1 if nv <n[nr]: n[nr] = nv
    heapq.heappush(queue, (nv, nr)) return n



    def encrypt_byte(self, byte_val): """ 将1个八位字节转换为2个弗罗贝尼乌斯数（各4位） """ output = []#
    拆分为半字节（四位组） for nibblein [(byte_val >> 4) & 0x0F, byte_val & 0x0F]: found =False while
    not found: candidate =random.randint(1, self.F) # 检查：它是否为间隙，并且是否对应一个半字节？ if
    candidate < self.n_struct[candidate % self.a1] and (candidate % 16 == nibble):
    output.append(candidate) found = True return output



    def decrypt_byte(self, n1, n2): """ 从这两个数字中恢复原始八位字节
    """ high = n1% 16 low = n2% 16 return (high << 4) | low



def generate_secure_密钥(n_elements=5): """ 生成健壮随
机密钥 (PGCD = 1) """ while True: a = 随机数.
randint(500, 1000) b = a +随机数.randint(1, 200) if
math.最大公约数(a, b) != 1: continue 密钥 = [a, b]for _
in range(n_elements - 2):
```

```
                next_val = (随机数.randint(1, 3) * 密钥[0]) + (随机数.randint(1, 3)* 密钥[-1]) if next_
            val not in 密钥: 密钥.append(next_val) if math.最大公约数(*密钥) == 1: return set(密钥)


# --- 测试区 --- if __name__ == "__main__
": # 1. 密钥生成 my_密钥 =生成_安全_密钥
(5) print(f"密钥已生成：{my_密钥}")
cipher =FrobCipher(my_密钥) # 2. 测试
消息 msg = "BONJOUR" print(f"原始消
息：{msg}")



    # 3. 加密 encrypted_stream = []for char
    inmsg: nums =cipher.encrypt_
    byte(ord(char)) encrypted_
    stream.extend(nums)

    print(f"加密流：{encrypted_stream}")

    # 4. 解密 decrypted_msg =""" for i inrange(0, len(encrypted_stream), 2): val =
    cipher.decrypt_byte(encrypted_stream[i] ,encrypted_stream[i+1]) decrypted_
    msg += chr(val)


    print(f"解密消息：{decrypted_msg}")
```

参考文献

[1] M. Beck 与 S. Zacks，弗罗贝尼乌斯线性丢番图问题的改进上界，《Advances in Applied Mathematics》**32**，454–467（2004年）。

[2] D. S. Binner，*Sylvester* 一项与弗罗贝尼乌斯硬币问题相关的结果的推广，《Journal of Integer Sequences》**24**，文章编号 21.8.4（2021年）。

[3] Y. C hen 与 P. Q. Nguyen，*BKZ 2.0*：带剪枝的更优格约化，ASIACRYPT，2011年 .

[4] T. H. Cormen、C. E. Leiserson、R. L. Rivest 与 C. Stein，《算法导论》（第4版，麻省理工学院出版社，2022年）。

[5] M. Delgado，关于 *Eliahou* 的一个问题和 *Wilf* 猜想，《数学杂志》**288**，595–627（2018年）。

[6] E. W. D ijkstra，关于图的两个问题的注记，《数值数学》**1**，269–271（1959年） .

[7] S. Eliahou，Wilf 猜想及其变体，预印本 arXiv:2006.01254（2020年）。[8] G. Frobenius，关于硬币问题，柏林讲座，1894年。[9] H. Greenberg，确定 Frobenius数的算法，马里兰大学计算机科学系（技术报告），1988年。[10] H. Greenberg，非负整数线性丢番图方程的解，《算法杂志》9，343–353（1988年）。[11] J. Herzog，阿贝尔半群及半群环的生成元与关系，《数学手稿》3，175–193（1970年）。[12] J. M. Howie，《半群理论基础》（牛津大学出版社，1985年）。[13] G. Kapetanakis 和 I. Rizos，关于两个面额的弗罗贝尼乌斯硬币问题的归纳证明，https://arxiv.org/abs/2308.03050 [14] N. Kaplan，数值半群计数，《美国数学月刊》124，862–875（2017年）。[15] A. Lenstra、H. Lenstra Jr. 和 László Lovász，有理数系数多项式的因式分解，《数学年刊》261，515–534（1982年）。[16] M. Lewin，关于近似等差数列的弗罗贝尼乌斯问题，《纯粹与应用数学杂志》273，5–18（1975年）。[17] M. Lewin，求解弗罗贝尼乌斯问题的算法，《纯粹与应用数学杂志》276，68–82（1975年）。[18] A. Moscariello 和 A. Sammartano，关于 Wilf 对 Frobenius数提出的猜想，《数学杂志》280，47–53（2015年）。[19] D. C. Ong 和 V. Ponomarenko，几何数列的 Frobenius数，《整数：组合数论电子期刊》8，文章 A33（2008年）。[20] P. Paillier，基于合数次剩余类的公钥密码系统，Eurocrypt，第1592卷，1999年。[21] J. L. Ramìrez Alfonsìn，弗罗贝尼乌斯问题的复杂性，《组合数学》16，143–147（1996年）。[22] J. L. Ramìrez Alfonsìn，《丢番图 Frobenius 问题》（牛津大学出版社，牛津，2005年）。[23] J. B. Roberts，关于一个丢番图问题，《加拿大数学杂志》8，444–448（1956年）。[24] J. B. Roberts，关于线性形式的注记，《美国数学会会刊》7，465–469（1956年）。[25] J. C. Rosales 和 P. A. García-Sánchez，《数值半群》（施普林格科学与商业媒体，第20卷，2009年）。[26] C. P. Schnorr，多项式时间格基约化算法的层级，《理论计算机科学》53，201–224（1987年）。[27] E. S. Selmer，关于弗罗贝尼乌斯线性丢番图问题，《纯粹与应用数学杂志》293/294，1–17（1977年）。[28] J. J. Sylvester，数学问题及其解答。第7382题，《教育时报》41，171–182（1884年）。[29] A. Tripathi，关于几何数列的弗罗贝尼乌斯问题，《整数：组合数论电子期刊》8，文章 A43（2008年）。

[30] A. Tripathi，三变量中的弗罗贝尼乌斯数公式，《数论杂志》**170**，368–389（2017）。

[31] A. Ustinov，关于含三个参数的弗罗贝尼乌斯数弱渐近性的 *Arnold* 问题的解，《数学文集》**200**，131–160（2009）。

[32] H. S. Wilf，用于货币兑换问题的灯环算法，《美国数学月刊》**85**，562–565（1978）。