科学通报 2010 年 第 55 卷 第 29 期: 2869 ~ 2873
论 文 www.scichina.com csb.scichina.com
《中国科学》杂志社
SCIENCE CHINA PRESS

# 重量固定的目标解量子搜索算法

汪翔, 鲍皖苏*, 付向群

解放军信息工程大学电子技术学院, 郑州 450004
* 联系人, E-mail: 2004bws@sina.com

2010-03-04 收稿, 2010-07-02 接受

**摘要** 针对重量固定为 $d$ 的 $n$ 维布尔向量目标解搜索问题, 给出了重量固定的向量标签表示方法与向量标签还原算法, 在此基础上提出了计算复杂性优于经典搜索算法的重量固定目标解量子搜索算法. 新算法计算复杂性是 $O\left(\sqrt{C_{n+1}^{d}}\right)$, 显著低于重量固定的目标解搜索问题经典求解算法, 并以 NTRU 公钥密码体制的私钥求解问题为例, 验证了新算法对重量固定的目标解搜索问题求解的有效性.

**关键词**
标签
量子搜索
计算复杂性
NTRU

量子计算机的概念是 Paul[1]和 Feyman[2]分别独立提出的. 1992 年, Deutsch 等人[3]设计了第一个展示量子计算机的计算能力可以超越电子计算机的量子算法, 而随后 Shor 量子计算算法[4]和 Grover 量子搜索算法[5]的相继提出, 对现代密码的安全性产生了巨大冲击, 引起了国内外学者对量子计算机、量子计算算法和量子密码研究的广泛关注[6~9], 极大地促进了该领域的迅猛发展.

本质上, Grover 量子搜索算法是量子计算机上的一个穷尽算法, 它使得现用密码算法穷尽攻击的计算复杂性由 $O(2^n)$降低到 $O(2^{n/2})$, 从而将现用密码的安全准则提上一个新的层次. 而随着 Grover 量子搜索算法研究的不断深入, Long 等人[10~13]基于相位变换提出了成功率为 1 的改进算法, 钟普查等人[14]提出了中间相遇量子搜索算法, 使得 Grover 量子搜索算法在密码学中的应用越来越广泛.

Grover 量子搜索算法是一个普适算法, 它的输入是全部 $n$ 维经典输入状态的均匀叠加态, 每一个状态都可能是目标解. 但在实际的密码体制破译中, 经常会遇到需要搜索一些有特殊条件的目标解问题(如重量是固定的目标解), 此时, 如果直接利用 Grover 量子搜索算法进行求解, 其计算复杂性并不优于经典的密码攻击方法, 甚至有时比经典搜索算法(强力攻击)的计算复杂性更大, 因此也就无法体现量子计算强大的并行能力的优势. 如何融合量子计算原理提高经典密码分析方法的效能一直是人们非常关注的一个重要问题, 本文针对重量固定的目标解搜索这一特定密码攻击问题, 给出了重量固定的目标解向量标签表示方法与向量标签还原算法, 提出了计算复杂性优于 Grover 算法的量子搜索算法, 新算法的计算复杂性显著低于重量固定的目标解搜索问题经典求解算法.

NTRU[15]公钥密码体制是 Jeffstey 等人在 1996 年提出的, 目前已被采用为 IEEE P1363 和 EESS (Consortium for Efficient Embedded Security)中公钥密码标准算法之一. 该算法以其实现速度快、安全性能高和实际应用广等优点, 受到国内外专家学者的广泛关注. 目前, 在经典密码分析中, 对 NTRU 公钥密码算法还没有有效的攻击方法. 2003 年, Ludwig[16]将 Grover 量子搜索算法应用到格规约算法中, 给出了量子计算下的 NTRU 格规约攻击算法; 然而受到格规约攻击计算复杂性的限制, NTRU 的量子格规约攻击方法的计算复杂性并不理想, 甚至比经典计算下 NTRU 的强力攻击和中间相遇攻击的计算复杂性都要大得多, 特别是当选取的参数 $N$ 较大时更为明显. 本质上, NTRU 的私钥求解问题可以抽象为一个重量

**英文版见**: Wang X, Bao W S, Fu X Q. A quantum algorithm for searching a target solution of fixed weight. Chinese Sci Bull, 2010, 55, doi: 10.1007/s11434-010-4113-4

固定的目标解搜索问题. 本文将重量固定的目标解量子搜索算法引入到 NTRU 强力攻击算法中, 大大降低了密钥穷尽的计算复杂性.

## 1 Grover 量子搜索算法

Grover 量子搜索算法用于搜索未加整理的数据库中满足条件 $f(x)=a$ 的解 $x$.

### 1.1 算法描述

该算法需要两个寄存器, 第 1 寄存器存放 $n$ 个量子比特并初始化为 $|0\rangle^{\otimes n}$, 第 2 寄存器存放 1 个量子比特并初始化为 $|1\rangle$.

(1) 制造 $n$ 量子比特的均匀叠加态. 对第 1 寄存器中的每一量子位置 $|0\rangle$ 都进行 Hadmard 变换, 有以下结果:

$$H^{\otimes n}|0\rangle^{\otimes n}=\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle=|s\rangle,$$

通过上述操作, 实现了 $n$ 个量子比特的均匀叠加, 得到状态 $|s\rangle$.

(2) oracle 描述. 未加整理的数据库搜索问题也可以变换为一个判定问题, 给出一个 oracle, 它可以迅速计算函数值 $f(x)$与$\alpha$比较, 并给出结果

$$\begin{cases} f_a(x)=0, if\ f(x)\neq a, \\ f_a(x)=1, if\ f(x)=a. \end{cases}$$

oracle 的作用可以写成

$$|x\rangle\xrightarrow{\text{oracle}}(-1)^{f_a(x)}|x\rangle.$$

我们说 oracle 通过改变解的相位来标记搜索问题的解.

(3) 执行 Grover 迭代 $\pi\sqrt{2^n}/4$ 次. Grover 迭代包括下面两个步骤:

① 执行 oracle;

② 对均匀叠加态 $|s\rangle$ 执行酉变换 $I_\varphi=2|\varphi\rangle\langle\varphi|-I$, 其中, $|\varphi\rangle=\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle$.

执行 oracle 的作用在于从叠加态中标记搜索问题的解, 酉变换 $I_\varphi$的作用在于对问题的解的几率幅进行放大, 进而降低非解的几率幅. 通过多次 Grover 迭代, 就可以使得解的几率幅达到最大, 从而, 在测量时以较高的概率输出问题的解.

(4) 测量第 1 寄存器得到一个搜索问题的解.

### 1.2 算法分析

Grover 量子搜索算法的 oracle 是可以简单实现的, 因此整个算法的计算复杂性为 $O(2^{n/2})$, 只取决于迭代次数的复杂性, 且算法不需要存储, 算法成功率为 1.

## 2 重量固定的目标解搜索问题

重量固定的目标解搜索问题就是对所有重量为某个固定值 $d$ 的输入进行搜索. 如果直接利用经典搜索算法求解, 实际上就是对 $n$ 维布尔向量中所有重量为 $d$ 的向量进行搜索, 然后找出使得 $f(x)=a$ 成立的目标解. 因此, 由组合数学的知识, 可以很容易的估计出重量固定的目标解经典搜索算法的计算复杂性为 $O\left(C_n^d\right)$. 如果直接利用 Grover 量子搜索算法求解, 根据前一节的分析可知, 求解算法的计算复杂性则是 $O(2^{n/2})$. 当需要搜索的目标解重量 $d$ 较小时, 有 $C_n^d<2^{n/2}$, 也就是说利用 Grover 量子搜索算法的效果反而没有经典搜索算法的效果好.

实际上, 由于 Grover 量子搜索算法是一个普适算法, 它的输入是全部 $n$ 维经典输入状态的均匀叠加态, 需要搜索的对象是所有的 $n$ 维布尔向量, 并且在求解过程无法区分需要搜索的向量重量, 每一个状态都可能是目标解. 但是, 在目标解重量固定为 $d$ 的条件下, 重量不为 $d$ 的状态显然不是它的目标解, 此时如果再以全部的 $n$ 维经典输入状态作为目标解的搜索空间, 则增大了算法的搜索空间, 也就没有将目标解重量固定这一条件充分利用起来. 而利用经典搜索算法时, 由于可以做到只对重量为 $d$ 的向量进行搜索, 因此经典算法的搜索空间只有 $C_n^d$. 搜索空间的不同也就决定了它们计算复杂性的差异, 所以对于重量固定的目标解搜索问题, 直接利用 Grover 量子搜索算法求解时效果并不理想.

## 3 重量固定的向量标签以及标签还原算法

对于重量固定的目标解搜索问题, 要提高其量子搜索算法的有效性, 就不能直接对所有 $n$ 维布尔向量都进行搜索, 而只对所有重量固定为 $d$ 的输入状态进行搜索, 这就需要将重量固定为 $d$ 的经典输入状态

以低于 $n$ 维的向量表示出来. 为此, 我们给出重量固定为 $d$ 的 $n$ 维向量的 $t$ ($t < n$)维向量标签表示方法和向量标签还原算法.

**定义 1** 设 $n$ 维布尔向量 $v$ 的重量为 $d$, 其中 1 出现的位置分别是 $i_1,i_2,\cdots,i_d$, 其余位置上的信息都为 0, 并且满足 $1\leqslant i_1<\cdots<i_d\leqslant n$, 则向量 $v$ 的"标签"定义为

$$I_v = 1 + C_{i_1}^1 + C_{i_2}^2 + \cdots + C_{i_d}^d,$$

并称需要求解的目标向量的标签为"目标标签".

由定义 1 可以将所有重量为 $d$ 的 $n$ 维布尔向量的标签都计算出来, 并且向量与其标签之间是一一对应的. 为了估计此时搜索目标标签所需要的量子比特数以及搜索空间的大小, 给出如下引理.

**引理 1** 在所有重量为 $d$ 的 $n$ 维二元向量的标签中, 其最小值与最大值分别为

$$\min I_v = 1 + C_1^1 + C_2^2 + \cdots + C_d^d = C_{d+1}^d,$$

$$\max I_v = 1 + C_{n-d+1}^1 + C_{n-d+2}^2 + \cdots + C_n^d = C_{n+1}^d.$$

由组合数学的知识可以很容易的证明引理 1. 而由引理 1 可知, 重量固定的向量标签的最大值决定了对标签进行搜索时所需要的量子比特数. 因此, 如果令

$$t = \min\{k \mid C_{n+1}^d \leqslant 2^k\},$$

则在对向量标签进行搜索时所需要的量子比特数就是 $t$ 个, 即对向量标签进行量子搜索的空间大小就是 $2^t$, 相对于直接对 $n$ 维布尔向量进行搜索的个数 $2^n$ 要小得多.

但是, 此时利用 Grover 量子搜索算法对向量标签进行搜索后, 得到的结果是重量固定目标解所对应的目标标签, 要得到最终的目标解, 还需要将目标标签还原成目标解的形式, 而且在对向量标签进行搜索的过程中, 向量标签本身是无法直接参与到函数 $f(x)$的运算中, 而要将标签还原成重量为 $d$ 的 $n$ 维布尔向量的形式, 因此需要给出重量固定的向量标签还原算法(算法 S1).

由于 $t$ 维布尔向量空间中的元素并不都是真正的向量标签, 因此在算法 S1 中, 如果输入的整数是重量为 $d$ 的 $n$ 维布尔向量的标签, 则可以输出该向量; 否则就会输出一个全 0 向量.

算法 S1 的计算主要集中在第 3 步到第 8 步的循环中, 而循环次数至多是 $n$ 次, 因此算法 S1 的计算复杂性是 $O(n)$. 也就是说如果已知一个重量为 $d$ 的 $n$ 维布尔向量的标签, 可以很容易地利用算法 S1 将该向量还原出来.

## 4 重量固定的目标解量子搜索算法

Grover 量子搜索算法中的 oracle 为一个简单的判断函数, 但是在输入重量固定的条件下, 为充分利用目标解重量固定这一条件, 此时 oracle 不可能再是一个简单判断函数. 此时, 是对向量的标签进行搜索, 因此在量子搜索算法中的 oracle 需要重新定义.

假设 oracle 的输入是 $t$ 维布尔向量 $b$, 则可调用标签还原算法求出标签 $b$ 对应的向量 $v$, 然后判定下式, 并给出 oracle 的输出:

$$\begin{cases} F_b = 0 & \text{if } f(v) \neq a, \\ F_b = 1 & \text{if } f(v) = a. \end{cases}$$

利用上述 oracle 可以给出重量固定的目标解量子搜索算法(算法 S2).

## 5 NTRU 私钥求解问题的量子搜索算法

为说明 NTRU 算法私钥求解问题, 首先简单介绍 NTRU 算法的密钥生成方法以及其推荐参数, 对于 NTRU 体制的加解密算法的具体过程, 可参见文献[15].

### 5.1 NTRU 密钥生成方法及推荐参数

NTRU 算法中需要 3 个整数参数($N$, $p$, $q$)和多项式环 $R=Z_q[X]/(X^N-1)$的 4 个子集 $L_F,L_g,L_r,L_m$, 其中 $F \in L_F = L(d_F)$ 表示多项式 $F$ 中只有 $d_F$ 个系数为 1, 其余系数为 0; 同样地, $L_g=L(d_g)$, $L_r=L(d_r)$; 而明文多项式 $m \in L_m$ 的系数也是在集合{0, 1}中. 在实际应用中为增加安全性, 要求 $N$ 是素数, 并且 gcd($p$, $q$)=1, $p<<q$.

NTRU 算法的密钥生成过程: 随机的从集合 $L_F$ 和 $L_g$ 中选取多项式 $F$, $g$, 计算 $f=1+pF$, 并分别计算 $f$ 在环 $R$ 上模 $q$ 和模 $p$ 的逆元 $f_q$ 和 $f_p$, 如果所选取的多项式 $f$ 没有逆元, 则重新选取多项式 $F$, 直到选出符合条件的多项式. 事实上, 对于合适的参数 $p$ 和 $q$, 绝大多数多项式都符合这一条件; 然后计算: $h=f_q*g \pmod q$. 则多项式 $f$ 和 $f_p$ 即为 NTRU 算法的私钥(为了安全 $g$ 也要保密), 多项式 $h$ 为算法的公钥, ($N$, $p$, $q$)是公开参数. 2005 年, NTRU 算法的设计者给出了最新的推荐参数表, 如表 1 所示(详见 www.ntru.com).

表 1 NTRU 体制推荐参数

| | $N$ | $d_F$ | $d_g$ | $d_r$ | $p$ | $q$ |
|---|---|---|---|---|---|---|
| NTRU251 | 251 | 48 | 125 | 48 | 2 | 197 |
| NTRU347 | 347 | 66 | 173 | 66 | 2 | 269 |
| NTRU491 | 491 | 91 | 245 | 91 | 2 | 367 |
| NTRU587 | 587 | 108 | 293 | 108 | 2 | 439 |

### 5.2 NTRU 体制现有攻击方法

在 NTRU 经典攻击方法研究方面, 1998 年, Jeffstey 等人[15]提出了对 NTRU 的强力攻击, 并在此基础上给出了 NTRU 体制中间相遇攻击, 有效地降低了攻击 NTRU 体制的计算复杂性, 但攻击算法中需要大量地存储空间; 1997 年, Don 等人[17]通过分析 NTRU 体制公钥与私钥之间的关系, 给出了 NTRU 的格规约攻击方法; 此后 Alexander 根据 NTRU 体制私钥的特殊形式, 给出了改进的 NTRU 格规约攻击方法, 降低了格规约攻击的计算复杂性(详见 http://citeseer. ist.psu.edu/article/may99cryptanalysis.html).

在 NTRU 量子攻击方法研究方面, 2003 年, Ludwig 将 Grover 量子搜索算法应用到格规约算法中, 并给出 NTRU 的量子格规约攻击方法计算复杂性的估计; 但在 2005 年, Nick 等人在详细分析 NTRU 格规约攻击的基础上, 给出 NTRU 格规约攻击的计算复杂性的实验估计与理论估计, 并说明 Ludwig 的格规约攻击方法的计算复杂性比中间相遇攻击的计算复杂性大得多, 即使是利用量子格规约方法对 NTRU 进行攻击的计算复杂性仍然比中间相遇攻击的大. 表 2 给出了目前对 NTRU 体制几种常用攻击方法的计算复杂性.

由表 2 可知, 中间相遇攻击目前仍是对 NTRU 体制最有效的攻击方法.

表 2 NTRU 体制不同攻击方法的计算复杂性比较

| 攻击方法 | 强力攻击 | 中间相遇攻击 | 格规约攻击 | 量子格规约攻击 |
|---|---|---|---|---|
| 计算复杂性 | $O\left(C_N^d\right)$ | $O\left(C_{N/2}^{d/2}/\sqrt{N}\right)$ | $O\left(N^3\left(k/6\right)^{k/4}\right)$ | $O\left(n^3\left(k/6\right)^{k/8}\right)$ |
| NTRU251 | $2^{172.7}$ | $2^{80}$ | $2^{520}$ | $2^{260}$ |
| NTRU347 | $2^{239.4}$ | $2^{112}$ | $2^{797}$ | $2^{398.5}$ |
| NTRU491 | $2^{335.1}$ | $2^{160}$ | $2^{949}$ | $2^{474.5}$ |
| NTRU587 | $2^{399.7}$ | $2^{192}$ | $2^{1581}$ | $2^{790.5}$ |

### 5.3 NTRU 私钥求解的量子搜索算法

由 NTRU 算法设计可知, 其私钥多项式是 $f=1+pF$ 形式, 其中多项式 $F$ 中只有 $d_F$ 个系数为 1, 其余系数为 0, 且次数小于 $N$, 因此只要能求出多项式 $F$ 也就可以求出 NTRU 体制的私钥. 而每一个多项式 $F$ 都可以表示成一个 $N$ 维布尔向量的形式, 而且是重量为 $d_F$ 的 $N$ 维布尔向量. 因此, NTRU 的私钥求解问题即可转化为重量固定的目标解搜索问题, 就可以利用重量固定的目标解量子搜索算法进行求解. 表 3 给出了直接利用 Grover 量子搜索算法与利用重量固定的目标解量子搜索算法来求解 NTRU 私钥时, 在不同参数下的计算复杂性.

由表 3 可知, 重量固定的目标解量子搜索攻击比 NTRU 经典强力攻击的计算复杂性显著降低, 而与表 2 中的中间相遇攻击的计算复杂性相比要稍大. 但需要注意的是中间相遇攻击方法还需要大量的存储空间, 其存储复杂性大约是 $O\left(C_{N/2}^{d/2}\right)$, 而本文提出的重量固定的目标解量子搜索攻击方法不需要存储. 因此, 新算法对 NTRU 体制的攻击效果要优于目前已知的最好攻击方法.

## 6 结语

本文针对重量固定的目标解搜索问题, 提出了计算复杂性优于 Grover 算法的量子搜索算法, 与重量固定的目标解搜索问题的经典求解算法相比, 新算法的计算复杂性有着显著的降低, 特别是当目标解的重量较小时, 新算法的效率更为明显. 并以 NTRU 公钥密码体制的私钥求解问题为例, 验证了新算法对重量固定的目标解搜索问题求解的有效性.

表 3 利用重量固定目标解量子搜索算法攻击 NTRU 体制的计算复杂性

| NTRU 参数 | $N$, $d_F$ | NTRU251 | NTRU347 | NTRU491 | NTRU587 |
|---|---|---|---|---|---|
| 计算复杂性 | $O\left(\sqrt{C_{N+1}^{d_F}}\right)$ | $2^{86.5}$ | $2^{119.8}$ | $2^{167.7}$ | $2^{200}$ |

## 参考文献

1 Paul B. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by turing machines. J Stat Phys, 1980, 22: 563—591

2 Feynman R. Simulating physics with computers. Int Theor Phys, 1982, 21: 467—488
3 Deutsch D, Jozsa R. Rapid solution of problems by quantum computation. Proc R Soc London A, 1992, 439: 553—558
4 Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput, 1997, 26: 1484—1509
5 Grover L K. A fast quantum mechanics algorithm for database search. In: Proceeding of the 28th ACM Symposium on Theory of Computation. New York: ACM Press, 1996. 212—219
6 陈巍, 韩正甫, 莫小范, 等. 量子密钥传输系统的主动相位补偿. 科学通报, 2007, 52: 2221—2225
7 高飞, 郭奋卓, 温巧燕, 等. Ping-pong 协议中不同检测策略的效率比较. 中国科学 G 辑: 物理学 力学 天文学, 2009, 39: 161—166
8 Xu F X, Chen W, Wang S, et al. Field experiment on a robust hierarchical metropolitan quantum cryptography network. Chinese Sci Bull, 2009, 54: 2991—2997
9 付向群, 鲍皖苏, 周淳. Shor 整数分解量子算法的加速实现. 科学通报, 2010, 55: 322—327
10 Long G L. Grover algorithm with zero theoretical failure rate. Phys Rev A, 2001, 64: 022307
11 Long G L, Zhang W L, Li Y S, et al. Arbitrary phase rotation can not be used in Grover's quantum search algorithm. Commun Theor Phys, 1999, 32: 335—338
12 Long G L, Li Y S, Zhang W L, et al. Phase matching in quantum searching. Phys Lett A, 1999, 262: 27—34
13 Long G L, Xiao L, Sun Y. Phase matching condition for quantum search with a generalized quantum database. Phys Lett A, 2002, 294: 143—152
14 Zhong P C, Bao W C. Quantum mechanical meet-in-the-middle search algorithm for Triple-DES. Chinese Sci Bull, 2010, 55: 321—325
15 Jeffstey H, Jill P, Joseph H S. NTRU: A ring based public key cryptosystem. ANTS III, 1998, 1423: 267—288
16 Ludwig C. A faster lattice reduction method using quantum search. Algor Comput, 2003, 2906: 199—208
17 Don C, Adi S. Lattice Attacks on NTRU. Germany: Springer-Verlage Press, 1997. 52—61

## 补充材料

**算法 S1　重量固定的向量标签还原算法**

**算法 S2　重量固定的目标解量子搜索算法**

本文的以上补充材料见网络版 csb.scichina.com. 补充材料为作者提供的原始数据, 作者对其学术质量和内容负责.

**算法 S1 重量固定的向量标签还原算法**

输入: 整数 $0 \leqslant b \leqslant 2^t - 1$, 整数 $n$ 和 $d$;

输出: $n$ 维布尔向量 $v$;

1: 令 $I_v=b$, 若 $C_{d+1}^{d} \leqslant I_v \leqslant C_{n+1}^{d}$ 则执行 Step2; 否则输出全 0 向量;

2: 令 $k=0$, $i_1=i_2=\cdots=i_d=0$;

3: for $j=0$ to $n-1$

4: if $k<d$ and $n-j \geq d-k$

5: if $I_v > C_{n-j}^{d-k}$

6: $i_{d-k}=n-j$;

7: $I_v = I_v - C_{n-j}^{d-k}$;

8: $k=k+1$;

9: 令 $n$ 维向量 $v=(v_1,v_2,\cdots,v_n)$为全 0 向量; 如果 $i_1 \neq 0$ 并且 $I_v=1$, 则执行 Step10; 否则执行 Step11

10: 令 $v_{i_1} = v_{i_2} = \cdots = v_{i_d} = 1$;

11: 输出 $n$ 维布尔向量 $v$.

**算法 S2 重量固定的目标解量子搜索算法**

输入: 函数 $f$, 常数 $a$ 以及目标解的维数 $n$ 和重量 $d$;

输出: 使得 $f(x)=a$ 成立的解 $x$;

1: 计算标签的最大值 $C_{n+1}^{d}$; 令 $t = \min\{k \mid C_{n+1}^{d} \leqslant 2^k\}$;

2: 利用 Grover 量子搜索算法对 $t$ 维布尔向量进行搜索, 得到输出 $b_0$;

3: 调用标签还原算法求出标签 $b_0$ 对应的 $n$ 维布尔向量 $v_0$;

4: 令 $x=v_0$;

5: 输出目标解 $x$.

**算法分析:**

由上述过程可知, 算法 2 中也不需要存储, 且成功率一定为 1. 下面分析算法 2 的计算复杂性.

算法 2 的关键步骤是第二步, 即利用量子搜索算法对 $t$ 维布尔向量进行搜索, 其他步骤中的计算都为多项式时间, 因此算法 2 的计算复杂性为 $O(2^{t/2})$. 而由 $t = \min\{k \mid C_{n+1}^{d} \leqslant 2^k\}$, 因此算法 2 的计算复杂性的近似估计为 $O\left(\sqrt{C_{n+1}^{d}}\right)$.

由表 1 可以知, 重量固定的目标解量子搜索算法的计算复杂性与经典搜索算法相比有着显著的降低. 为进一步说明这点, 本文以 NTRU 公钥密码体制的私钥求解问题为例, 说明重量固定的目标解量子搜索算法在密码分析的实际应用中的重要作用.

**表 1 重量固定目标解搜索问题不同求解算法的计算复杂性比较**

| 搜索算法 | 经典搜索算法 | 重量固定的目标解量子搜索算法 |
|---|---|---|
| 计算复杂性 | $O\left(C_n^d\right)$ | $O\left(\sqrt{C_{n+1}^{d}}\right)$ |