子集和问题的量子中间相遇搜索算法

鲍皖苏[1],宋 震[1],钟普查[2],付向群[1]
(1.解放军信息工程大学电子技术学院,河南郑州 450004; 2.湖南省岳阳军分区,湖南岳阳 414000)

摘 要: 子集和问题是 NP 完全问题,该问题是背包公钥的基础.现有最优的经典算法求解规模为 n 的子集和问题需要 $O(n2^{n/2})$步运算.本文提出了基于时空折衷思想的量子中间相遇搜索算法,该算法可以在 $O(n2^{n/3})$步求解规模为 n 的子集和问题,其存储复杂性为 $O(2^{n/3})$.由于 NP 完全问题可以在多项式时间内可相互归约,所以,在存储复杂性为 $O(2^{n/3})$的条件下,量子中间相遇搜索算法使得 NP 完全问题的计算复杂性降为 $O(n2^{n/3})$.

关键词: 量子算法; 子集和问题; 计算复杂性; 中间相遇
中图分类号: TP 301.6 **文献标识码:** A **文章编号:** 0372-2112 (2011) 01-0128-05

Quantum Mechanical Meet-in-the-middle Algorithm for Subset Sum Problem

BAO Wan-su[1], SONG Zhen[1], ZHONG Pu-cha[2], FU Xiang-qun[1]
(1. *Institute of Electronic Technology, The PLA Information Engineering University, Zhengzhou, Henan* 450004, *China*;
2. *Yueyang Army Subarea of Hunan Province, Yueyang, Hunan* 414000, *China*)

Abstract: Subset sum problem is one of the NP complete problems, which is the foundation of knapsack encryption schemes. Its computational complexity is $O(n2\exp(n/2))$ in classical algorithms. We present the quantum mechanical meet-in-the-middle algorithm, which can solve the subset sum problem in $O(n2\exp(n/3))$ with $O(2\exp(n/3))$ memory cost, and $O(2\exp(n/2))$ in quantum mechanical algorithm. The NP complete questions are minimized in $O(n2\exp(n/3))$ under this algorithm because of their equivalence.

Key words: quantum algorithm; subset sum problem; computational complexity; meet-in-the-middle

1 引言

随着物理学原理和计算机科学的交融和相互促进,量子信息与量子计算理论科学逐步发展起来.自从 Feyman[1]制造了一个可示范利用量子系统做运算的抽象模型以后,很多学者对量子计算进行了深入的研究.1985 年,Deutsch[2]证明了量子计算机比经典计算机有更强大计算能力.1994 年,Shor[3]在 Simon[4]研究的基础上提出了量子计算机上的大数质因子分解算法,该算法能够在多项式时间内完成,这对基于大数质因子分解和离散对数问题的公钥密码如 RSA 等提出了巨大的挑战.1996 年,Grover[5]提出了量子计算机上未加整理数据库的搜索算法,相对于经典的算法,它提供了二次加速(quadratic speed-up),即由 $O(2^n)$降为 $O(2^{n/2})$.Grover 量子搜索算法虽然没有实现多项式时间加速,但它是量子计算机上的穷举算法,可以用来加速一切 NP 问题的求解.量子计算算法,特别是 Shor 算法与 Grover 量子搜索算法引起了世界物理学家、密码学家和各国政府对量子计算机和量子计算算法研究的重视,促进了量子计算研究的迅猛发展[6~9].长期以来,国内外学者对量子搜索算法在一些具体的 NP 问题中的应用十分关注,他们试图以多项式时间解决某个特定的 NP 完全问题[10,11],或者利用具体问题中的一些结构来设计量子算法,使得新算法的计算速度优于 Grover 量子搜索算法.但是分析发现这样算法[12,13]或多或少存在一些问题,如 Brassard[12]等人提出的量子碰撞问题算法就错误地将子集中元素都当成问题的解,而实际上子集中只含一个解.本文利用时空折衷法,提出了子集和问题的量子中间相遇搜索算法,该算法能够在 $O(n2^{n/3})$步求解规模为 n 的子集和问题,其存储复杂性为 $O(2^{n/3})$.该算法把握子集和问题的密钥可分特性,主要通过 Oracle 的设计,实现了存储复杂性与计算复杂性转换,已经不是一个基于简单搜索[14]的量子算法.由于 NP 完成问题可以相互归结,所以,在存储复杂性为 $O(2^{n/3})$的条件下,量子中间相遇搜索算法使得 NP 完全问题的计算复杂性降为 $O(n2^{n/3})$.

收稿日期: 2009-12-28; 修回日期: 2010-07-27

2 子集和问题

子集和问题是背包公钥密码的基础,其数学描述如下:

定义 1[15] 已知正整数向量 $\boldsymbol{B}=(b_1,b_2,\cdots,b_n)$和正整数 S,其中 $n\geqslant 3$,求解满足 $\sum_{i=1}^{n} b_i x_i = S(\boldsymbol{B}\cdot\boldsymbol{X}_j = S)$ 的向量 $\boldsymbol{X}_j=(x_1,x_2,\cdots,x_n)$的问题称为子集和问题,其中 $1\leqslant j\leqslant 2^n, x_i\in\{0,1\}, N=2^n$.

背包公钥加密的基本思想是选择一个特殊的子集和问题实例,然后将它伪装成一个很难求解的一般子集和问题实例.大多数背包公钥密码仅仅利用了子集和问题中的一些特例作为私钥加密,如超递增背包,因此存在很多安全隐患,但子集和问题本身确实是一个NP完全问题.求解子集和问题传统的方法是对向量 X_j 逐个赋值检验,即穷举攻击,该方法平均要尝试 $N/2$ 次才能找到正确答案,最坏情况下需要 N 次,所以其计算复杂度为 $O(2^n)$.

3 经典中间相遇攻击求解子集和问题

中间相遇攻击是对公钥密码、分组密码和序列密码的一种常用攻击方法.中间相遇攻击是一类时空折衷算法,它是通过增加存储复杂性来降低计算复杂性,其本质仍是穷举攻击方法.时空折衷思想由 Hellman[16] 提出,并成为的密码分析的一个有效手段,在多方面得到了应用[17~21].文献[21]给出了基于中间相遇攻击思想的经典算法,该算法能够在 $O(n2^{n/2})$步求解子集和问题,是已知的经典计算下求解子集和问题的最优算法.

3.1 算法描述

输入:正整数集合$\{b_1,b_2,\cdots,b_n\}$,正整数 S.

输出:$\boldsymbol{X}_j=(x_1,x_2,\cdots,x_n), x_i\in\{0,1\}$,使得 $\sum_{i=1}^{n} b_i x_i = S$,如果这样的 $\boldsymbol{X}_j$ 存在.

Step 1.设 $t\leftarrow\lfloor n/2\rfloor$;

Step 2.建立一个表,表中的内容为$(\sum_{i=1}^{t} b_i x_i,(x_1,x_2,\cdots,x_t))$,其中有$(x_1,x_2,\cdots,x_t)\in(Z_2)^t$;按第一项对表排序.

Step 3.对每个$(x_{t+1},x_{t+2},\cdots,x_n)\in(Z_2)^{n-t}$,执行如下计算:

①计算 $l=S-\sum_{i=t+1}^{n} b_i x_i$ 并使用二分查找法检验 l 是否是表中第一项的某个条目.

②若 $l=\sum_{i=1}^{t} b_i x_i$,则返回(一个解为$(x_1,x_2,\cdots,x_n)$).

Step 4.返回(无解).

3.2 算法分析

在上面算法中第二步需要 $O(2^t)$的存储复杂性,同时,对 2^t 项排序所需要 $O(2^t\log 2^t)$步计算.在算法第三步使用二分查找法确定一个 l 是否在表中需要$O(\log 2^t)$步计算,因此,对 2^{n-t}个条目的判定需要 $O(2^{n-t}\log 2^t)$步计算.由于 $t\leftarrow\lfloor n/2\rfloor$,所以整个算法的存储复杂性约为 $O(2^{n/2})$,计算复杂性

$$O(2^t\log 2^t)+O(2^{n-t}\log 2^t)=O(2^{n/2}\log 2^{n/2})+O(2^{n-n/2}\log 2^{n/2})$$

为 $O(n2^{n/2})$.该算法虽然还是指数时间算法,但相对于简单穷举攻击已经有了很大的加速.

4 Grover 量子搜索算法

Grover 量子搜索算法[5]在一个未整理的数据库中搜索满足条件 $f(x)=a$ 的一个解 x.

4.1 算法描述

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

(1)制造 n 量子比特的均匀叠加态

对第 1 个寄存器中的每一量子位$|0\rangle$进行 Hadmard 变换,有以下结果:

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

通过上述操作,实现了 n 个量子位的均匀叠加,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重新改变为一个判定问题,给出一个 Oracle,它可以迅速计算函数值 $f(x)$并与 a 比较,给出结果:

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

Oracle 的作用可以写成

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

我们说 Oracle 通过改变解的相位,标记了搜索问题的解.

(3)执行 Grover 迭代$\sqrt{N}$次

Grover 迭代包括下面两个步骤:

①执行 Oracle;

②对叠加态$|s\rangle$执行酉变换

$$I_\varphi=2|\varphi\rangle\langle\varphi|-I,|\varphi\rangle=\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle.$$

执行 Oracle 的作用在于从叠加态中挑选出搜索问

题的解,我们称为目标元素;酉变换 I_φ 的作用在于对目标元素的几率幅进行放大,进而减小非目标元素的几率幅.通过执行 Grover 迭代多次,就可以使得满足条件的目标元素的几率幅达到最大.

(4)测量第 1 个寄存器得到一个搜索问题的解

4.2 算法分析

Grover 量子搜索算法的 Oracle 是可以简单实现的,因此不考虑其计算复杂性,整个算法的计算复杂性由迭代次数决定,所以,Grover 量子搜索算法的计算复杂性为 $O(2^{n/2})$,且算法无存储复杂性.

5 子集和问题的量子中间相遇搜索算法

直接应用 Grover 量子搜索算法解决子集和问题只能提供二次加速,即计算复杂性仍然是 $O(2^{n/2})$.本节结合经典中间相遇攻击方法和 Grover 量子搜索算法,给出求解子集和问题的量子中间相遇搜索算法,其基本思想是先计算 $\boldsymbol{B}\cdot\boldsymbol{X}_j$ 前 $k=\lfloor n/3\rfloor$ 比特 $\sum_{i=1}^{k}b_ix_i$,其中 $\boldsymbol{B}=(b_1,b_2,\cdots,b_n)$,$\boldsymbol{X}_j=(x_1,x_2,\cdots,x_n)$,制成表 L,表中的内容为$(\sum_{i=1}^{k}b_ix_i,(x_1,x_2,\cdots,x_k))$,按第一项对表 L 排序;然后使用 Grover 算法在 2^{n-k} 个向量中找出满足 $f=S-\sum_{i=k+1}^{n}b_ix_i$ 且 f 属于表 L 的第一项的向量$(x'_{k+1},x'_{k+2},\cdots,x'_n)$,最后由 $f=S-\sum_{i=k+1}^{n}b_ix_i=\sum_{i=1}^{k}b_ix_i$ 确定前 k 比特.

5.1 算法描述

该算法需要三个寄存器,第 1 个寄存器存放 $n-k$ 个量子位并初始化为$|0\rangle^{\otimes(n-k)}$,第 2 个存储器存放 1 个量子位并初始化为$|1\rangle$,第 3 个寄存器为 k 比特.

定义 Oracle 为 $F_x\to\{0,1\}$,并满足以下关系的函数:

$$\begin{cases}F_x=0,\text{if } f'(x)=S-\sum_{i=k+1}^{n}b_ix_i\in L;\\ F_x=1,\text{if } f'(x)=S-\sum_{i=k+1}^{n}b_ix_i\notin L.\end{cases}$$

通过 Oracle 就可以标记满足子集和问题的解的 $n-k$ 比特$(x_{k+1},x_{k+2},\cdots,x_n)$,此时 Oracle 需要在已排序的表 L 中使用二分查找法检验 $S-\sum_{i=k+1}^{n}b_ix_i$ 是否是表 L 中第一项的某个条目,Oracle 的设计是实现时空折衷的关键.算法步骤如下:

Step 1. 设 $k\leftarrow\lfloor n/3\rfloor$;

Step 2. 建立一个表 L,表中的内容为$(\sum_{i=1}^{k}b_ix_i,(x_1,x_2,\cdots,x_k))$,其中有$(x_1,x_2,\cdots,x_k)\in(Z_2)^k$,并按第一项对表 L 进行排序.

Step 3. 制造 $n-k$ 量子比特的均匀叠加态$|s'\rangle$

对第 1 个量子寄存器中的每一量子位进行 Hadmard 变换,有以下结果:

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

通过上述操作,实现了 $n-k$ 个量子位的均匀叠加,得到状态$|s'\rangle$.

Step 4. 执行 Grover 迭代$\sqrt{2^{n-k}}$次

Grover 迭代包括下面两个步骤:

①执行 Oracle;

②对叠加态$|s'\rangle$执行酉变换 $I'_\varphi=2|\varphi'\rangle\langle\varphi'|-I$,$|\varphi'\rangle=\frac{1}{2^{(n-k)/2}}\sum_{x=0}^{2^{(n-k)}-1}|x\rangle$.

Step 5. 输出$(x'_{t+1},x'_{t+2},\cdots,x'_n)$. 此时,$S-\sum_{i=k+1}^{n}b_ix'_i$ 是表 L 中第一项的一个条目.

Step 6. 利用 $S-\sum_{i=k+1}^{n}b_ix'_i=\sum_{i=1}^{k}b_ix'_i$ 在表 L 中找出 $\sum_{i=1}^{k}b_ix'_i$ 并确定$(x'_1,x'_2,\cdots,x'_k)$.

Step 7. 返回(一个为解$(x'_1,x'_2,\cdots,x'_n)$).

5.2 算法分析

算法的 Step 2 所需要的存储复杂性为 $O(2^k)$,同时,对表 L 进行排序的计算复杂性为 $O(2^k\log2^k)$;Step 4 中 Oracle 在已排序的表 L 中检验 $f'(x)=S-\sum_{i=k+1}^{n}b_ix_i$ 是否是表 L 中第一项的某个条目的计算复杂性为 $O(\log2^k)$,由于共执行了$\sqrt{2^{n-k}}$次迭代,所以 Step 4 的计算复杂性为 $O(\sqrt{2^{n-k}}\log2^k)$;Step 6 确定$(x'_1,x'_2,\cdots,x'_k)$所需要的计算复杂度为 $O(\log2^k)$;其它步骤不涉及计算复杂性与存储复杂性考虑;所以,整个算法的计算复杂性为 $O(2^k\log2^k)+O(\sqrt{2^{n-k}}\log2^k)+O(\log2^k)$,由于$k=\lfloor n/3\rfloor$,得到

$$\begin{aligned}&O(2^k\log2^k)+O(\sqrt{2^{n-k}}\log2^k)+O(\log2^k)\\&=O(2^{n/3}\log2^{n/3})+O(\sqrt{2^{n-n/3}}\log2^{n/3})+O(\log2^{n/3})\\&\cdot O(2^{n/3}\log2^{n/3})=O(2^{n/3}\log2^{n/3})=O(2^{n/3}n)\end{aligned}$$

所以,算法的计算复杂性为 $O(2^k\log2^k)=O(2^{n/3}n)$,算法的存储复杂性为 $O(2^k)=O(2^{n/3})$.

与经典中间相遇攻击相比,新算法在存储复杂性相等的条件下计算复杂性由 $O(n2^{n/2})$降为 $O(2^{n/3}n)$.与相比 Grover 量子搜索算法,新算法在增加了存储复杂性的条件下实现了降低计算复杂性的目标.同时,由 Grover 量子搜索算的确定性,新算法的成功率也是 1.

5.3 最优性证明

5.2 节给出量子中间相遇算法求解子集和问题的

计算复杂性为 $O(2^{n/3}n)$,下面证明该复杂性为算法复杂性的最小值.

定理 1　假设子集和问题的规模为 n,给定整数 k,则利用量子中间相遇搜索算法解决子集和问题的计算复杂性为 $O(2^k\log 2^k)+O\left(\sqrt{2^{n-k}}\log 2^k\right)+O(\log 2^k)$,且存储复杂性为 $O(2^k)$.特别地当 $k=n/3$ 时,算法计算复杂性取到最小值 $O(2^{n/3}n)$,此时存储复杂性为 $O(2^{n/3})$.

证明:从算法的描述与分析得到,算法的计算复杂性为 $O(2^k\log 2^k)+O\left(\sqrt{2^{n-k}}\log 2^k\right)+O(\log 2^k)$,下面证明在 $k=n/3$ 取得最小值 $O(2^{n/3}n)$.

因为 $O(\log 2^k)\leqslant O(2^k\log 2^k)$,所以

$$O(2^k\log 2^k)+O\left(\sqrt{2^{n-k}}\log 2^k\right)+O(\log 2^k)$$
$$=O(2^k\log 2^k)+O(2^{(n-k)/2}\log 2^k)$$

又因为 $2^k+2^{(n-k)/2}\geqslant 2\sqrt{2^k\cdot 2^{(n-k)/2}}$,当 $2^k=2^{(n-k)/2}$时取得最小值,此时 $k=n/3$,

所以 $O(2^k\log 2^k)+O(2^{(n-k)/2}\log 2^k)$在 $k=n/3$ 取得最小值 $O(2^{n/3}n)$,此时存储复杂性为 $O(2^{n/3})$.

证毕

定理 1 证明了利用中间相遇攻击思想在量子计算机上解决子集和问题不可能存在计算复杂性低于 $O(2^{n/3}n)$的算法.

6　结束语

本文提出了子集和问题的量子中间相遇算法,该算法结合了经典中间相遇攻击算法和 Grover 量子搜索算法的特点,算法的 Oracle 已经不是简单的数值的对比,这也为今后利用 Oracle 的特性设计量子计算算法提供了新的思路.与已有的子集和问题算法相比,该算法以一定的存储复杂性为代价,显著降低了计算复杂性,因此为抵抗量子计算机的攻击,未来新型背包公钥密码体制的向量维数应结合量子计算机的实际计算能力相应增大.

参考文献:

[1] R Feynman. Simulating physics with computers[J]. Int Theor Phys, 1982, 21: 467 – 470.

[2] D Deutsch. Quantum theory, the Church-Turing principle and universal quantum computer[J]. Proc R Soc London, A 1985, 400: 97 – 117.

[3] D R Simon. On the power of quantum computation[A]. Proceeding of the 35th Annual IEEE Computer Society[C]. Los Alamitos: IEEE Press, 1994. 116 – 123.

[4] P W Shor. Polynomial-Time Algorithms for prime factorization and discrete logarithms on a quantum computer[J]. SIAM Journal on Computing, 1997, 26(5): 1484 – 1509. (preliminary version in FOCS 1994).

[5] L K Grover. A fast quantum mechanics algorithm for database search[A]. Proceedings 28th ACM Symposium on Theory of Computation[C]. New York: ACM Press, 1996. 212 – 219.

[6] 孙莹,温巧燕,朱甫臣.基于可重用基序列的量子安全通信方案[J].电子学报,2010,38(1):111 – 116.
Sun Ying, Wen Qiao-yan, Zhu Fu-chen. Quantum secure communication based on the reusable bases sequences[J]. Acta Electronica Sinica, 2010, 38(1): 111 – 116. (in Chinese)

[7] 温晓军,田原,牛夏牧.一种基于秘密共享的量子强盲签名协议.电子学报,2010,38(1):720 – 724.
Wen Xiao-jun, Tian Yuan, Niu Xia-mu. A strong blind quantum signature protocol based on secret sharing[J]. Acta Electronica Sinica, 2010, 38(1): 720 – 724. (in Chinese)

[8] 李志强,陈汉武,徐宝文,肖芳英,薛希玲.四量子可逆逻辑电路快速综合算法[J].电子学报,2008,36(11):2081 – 2089.
Li Zhi-qiang, Chen Han-wu, Xu Bao-wen, Xiao Fang-ying, Xue Xi-ling. Fast algorithms for 4-qubit reversible logic circuits synthesis[J]. Acta Electronica Sinica, 2008, 36(11): 2081 – 2089. (in Chinese)

[9] 徐文旭,廖明宏.最长公共子序列的量子算法[J].电子学报,2007,35(12A):99 – 103.
Xu Wen-xu, Liao Ming-hong. Quantum algorithm for longest common subsequence[J]. Acta Electronica Sinica, 2007, 35(12A): 99-103. (in Chinese)

[10] 胡劲松,陈国良,郭光灿.在量子计算机上求解 0/1 背包问题[J].计算机学报,1999,22(12):1314 – 1316.
Hu Jin-song, Chen Guo-liang, Guo Guang-can. Solving the 0/1-knapsack problem in quantum quantum computing[J]. Chinese J. Computers, 1999, 22(12): 1314 – 1316. (in Chinese)

[11] 吕欣,冯登国.背包问题的量子算法分析[J].北京航空航天大学学报,2004,30(11):1088 – 1091.
Lü Xin, Feng Deng-guo. Quantum algorithm analysis of knapsack problem[J]. Journal of Beijing University of Aeronautics and Astronautic, 2004, 30(11): 1088 – 1091. (in Chinese)

[12] G Brassard, P Hoyer, A Tapp. Quantum algorithm for the collision problem[DB]. quant-ph /9705002, 1997.

[13] Miao Xijia. Solving the quantum search problem in polynomial time on an NMR quantum computer[DB]. Qunat-ph/0206102, 2002.

[14] M A Nielson, I L Chuang. Quantum Computation and Quantum Information[M]. Cambridge: Cambridge University, 2000.

[15] B Schneier. Applied Cryptography (second edition)[M]. New York: John Wiley & Sons Press 1996. 331 – 340.

[16] E Martin, A Hellman. Cryptanalytic time-memory tradeoff[J]. IEEE Transactions on Information Theory, 1980, 26(4): 401 – 406.

[17] Jin Hong, Palash Sarkar. Rediscovery of Time Memory Trade-

offs[OL]. http://eprint.iacr.org/,2005.
[18] Jin Hong, Palash Sarkar. New applications of time memory data tradeoffs[A]. Advances in Cryptology, Proceedings of Asiacrypt 2005, Lecture Notes in Computer Science 3788[C]. Berlin: Springer-Verlag, 2005. 353 – 372.
[19] Philippe Oechslin. Making a faster cryptanalytic time- memory trade-off[A]. Advances in Cryptology, Proceedings of CRYPTO 2003, Lecture Notes in Computer Science 2729[C]. Berlin: Springer-Verlag, 2003. 617 – 630.
[20] M O Saarinen. A time-memory tradeoff attack against LILI-128[A]. Proceedings of FSE 2002, Lecture Notes in Computer Science 2365[C]. Berlin: Springer-Verlag, 2002. 231 – 236.
[21] A J Menezes, P C Oorschot, S A Vanstone. Handbook of Applied Cryptography[M], Canda: CRC Press LLC, 1997. 117 – 118.

作者简介:

鲍皖苏 男,1966 年生于安徽天长,解放军信息工程大学电子技术学院教授、博士生导师,主要研究方向为序列密码、公钥密码、量子密码. E-mail:2010thzz@sina.com

宋 震 男,1976 年生于陕西三元,博士,主要研究方向为信息安全与可信计算.

钟普查 男,1982 年生于湖南岳阳,硕士,湖南省岳阳军分区参谋,研究方向为量子密码.

付向群 男,1985 年生于江西进贤,解放军信息工程大学电子技术学院博士研究生,研究方向为量子密码.