基于 D-Wave Advantage 量子退火算法的 90 比特 RSA 整数分解研究

洪春雷 裴 植 王 潮

（上海大学特种光纤与光接入网重点实验室 上海 200444）

摘 要 业内认为在当前量子计算攻击密码整体进展缓慢背景下，RSA 整数分解进展每提升 1 比特都面临挑战。根据《Nature》文章报道，2019～2023 年谷歌不断改进其量子芯片，但依旧不能用于密码破译。谷歌等公司近期的研究表明：尽管亚线性量子资源方法分解 RSA 整数可以降低量子资源的消耗，但是即使“完美的量子优化算法＋Babai 算法”也不足以有效地分解 80 比特以上的 RSA 整数。量子退火算法凭借其独特的量子隧穿效应，可跳出传统智能优化算法极易陷入的局部极值，快速逼近全局最优解。鉴于 D-Wave Advantage 的量子资源已达到 5000＋量子比特，本文通过使用更多的量子资源，提出一种量子退火算法结合经典密码算法分解 RSA 整数的混合架构。通过提高最近向量问题（Closest Vector Problem，CVP）的规模，从而提升搜索用于分解 80 比特以上 RSA 整数光滑对的能力；本文使用 Block Korkin-Zolotarev（BKZ）算法对 CVP 的格基进行约化，获得较 LLL 算法更优的归约基。利用更优的归约基，Babai 算法可以获得更优的 CVP 的解。在此基础上，本文利用量子退火算法的隧穿效应进一步优化 Babai 算法对 CVP 的求解，获得较 Babai 算法更优的解，从而提高光滑对的搜索效率，加速 RSA 整数分解。最后，本文在 D-Wave Advantage 上首次完成量子计算分解 80 比特以上的 RSA 整数，最大分解 90 比特 RSA 整数：629367860625666765619139989＝6398047085669×98368744743281，大幅度超出富士通、洛克希德马丁公司、普渡大学的实验指标。实验结果表明：研究量子智能算法和量子位数较多的量子计算机攻击密码是有意义的，未来需要重视量子隧穿推进 CVP 等 NP 难题求解的潜力，其全局寻优能力可能成为密码攻击的关键。

关键词 RSA 整数；Block Korkin-Zolotarev 算法；Babai 算法；最近向量问题；量子退火；D-Wave Advantage

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2025.01733

Research on 90-bit RSA Integer Factorization Based on D-Wave Advantage Quantum Annealing Algorithm

HONG Chun-Lei PEI Zhi WANG Chao

（Key Laboratory of Specially Fiber Optics and Optical Access Networks，Shanghai University，Shanghai 200444）

Abstract RSA encryption algorithm，based on the NP-hard problem of large integer factorization，serves as the cornerstone of public-key encryption and is widely used in information security. Quantum computing is a novel computational paradigm that transcends classical computing，leveraging the laws of quantum mechanics to accelerate the solving of complex computational problems. Given the slow progress in quantum computing applications for cryptographic attacks，the scientific community agrees that achieving even a 1-bit improvement in RSA integer factorization is highly challenging. As reported in《Nature》，Google has been continuously improving its quantum chips from 2019 to 2023，but it is still not suitable for cryptanalysis. Although the method utilizing sublinear quantum resources for the factorization of RSA integers reduces the consumption of quantum resources，even the integration of a perfectly quantum-optimized algorithm with the

收稿日期：2024-12-04；在线发布日期：2025-03-12。**洪春雷**，博士研究生，主要研究方向为网络空间安全、量子计算密码。E-mail：hcl@shu.edu.cn。**裴 植**，博士研究生，主要研究方向为网络空间安全、量子计算密码。**王 潮**（通信作者），博士，教授，中国计算机学会（CCF）高级会员，主要研究领域为人工智能、网络空间安全、量子计算密码。E-mail：wangchao@shu.edu.cn。

Babai algorithm remains insufficient to effectively factorize RSA integers larger than 80-bit. Quantum Annealing, with its unique quantum tunneling effect, can escape from the local minima that traditional intelligent optimization algorithms are prone to easily, and quickly approach the global optimal. Considering that the quantum resources of the D-Wave Advantage have reached over 5000+ qubits, we propose a hybrid architecture that integrates a quantum annealing algorithm with classical cryptographic algorithms for the factorization of RSA integers by using more quantum resources. Expanding the scale of the Closest Vector Problem (CVP) can enhance the ability to search for smooth pairs that factorize RSA integers above 80-bit. We use the Block Korkin-Zolotarev (BKZ) algorithm to reduce the lattice basis of the CVP, obtaining a more accurate lattice reduction basis than that provided by the LLL algorithm. The application of an improved lattice reduction basis enables Babai's algorithm to obtain more accurate solutions for the CVP. Building on this foundation, the tunneling effect of the quantum annealing algorithm is employed to further optimize the Babai algorithm for solving the CVP. This optimization yields solutions that surpass those of the original Babai algorithm, significantly enhancing the search efficiency for smooth pairs. As a result, the process of RSA integer factorization is accelerated, demonstrating the potential of quantum techniques to revolutionize classical cryptographic challenges. Finally, we use quantum computing to factorize RSA integers above 80-bit for the first time on the D-Wave Advantage, and the maximum factorization of 90-bit RSA integer: 629367860625666765619139989=6398047085669×98368744743281. The experimental indicators significantly exceed those of Fujitsu, Lockheed Martin, and Purdue University, demonstrating the superior performance and effectiveness of the proposed approach. The experimental results clearly show that the importance of studying quantum intelligent algorithms and quantum computers equipped with more qubits to effectively attack cryptographic algorithms cannot be overstated. In the future, it is absolutely essential to focus on the potential of quantum tunneling to facilitate the efficient resolution of NP-hard problems such as the CVP. Its capability for global optimization could prove crucial in significantly advancing cryptographic attacks, particularly in various challenging scenarios where classical methods face insurmountable computational bottlenecks. Additionally, exploring the integration of quantum tunneling with classical techniques may unlock new transformative avenues for solving complex optimization problems more effectively.

Keywords RSA integers; Block Korkin-Zolotarev algorithm; Babai algorithm; closest vector problem; quantum annealing; D-Wave Advantage

1 引 言

RSA 密码体制[1]是目前广泛应用的公钥密码体制，致力于维护网络通信的安全，其安全性依赖于大整数分解的困难性（分解整数 $N=P\times Q$，P 和 Q 为两个大素数）。1994 年 Shor[2]提出一种理论上可以在多项式时间内对任意整数进行质因子分解的算法，对公钥密码 RSA 的安全性造成了潜在威胁。近年来，各大科研机构以及高校也在致力于 RSA 攻击的研究。2018 年普渡大学 Jiang 等人[3]使用一种改进乘法表的算法，在 D-Wave 量子计算机上分解最大为 19 比特的 RSA 整数 376289；2021 年上海大学理学院 Hegade 等人[4]使用数字化绝热量子分解算法分解最大为 12 比特 RSA 整数 2479；2022 年 Yan 等人[5]在超导量子计算机上使用亚线性量子资源分解了最大为 48 比特的 RSA 整数 261980999226229。此外，各大公司也逐步开展对RSA攻击的研究。2019 年洛克希德马丁公司[6]通过量子退火算法分解最大为 13 比特 RSA 整数 7781；同年，谷歌[7]推出了量子霸权芯片 Sycamore，但是它无法用于密码破译；2023 年富士通公司使用量子模拟器分解 8 比特 RSA 整数 253[8]；同年，《Nature》文章报道谷歌进一步改进了量子霸权芯片 Sycamore，但仍无法用于密

码破译[9]。2023 年 6 月份谷歌[10]研究员表示使用文献[5]中的亚线性量子资源难以有效地分解 80 比特以上 RSA 整数。2023 年 7 月份 IonQ 公司[11]研究员表示在当前量子计算机硬件条件下“QAOA+Babai 算法”无法分解 80 比特以上的 RSA 整数。本源量子公司[12]曾指出：在当前量子计算攻击密码整体进展缓慢背景下，每提升 1 比特的 RSA 整数分解进展都极其困难。因此，需要探索一种适合当前量子计算机现实情况的密码攻击技术路线。

D-Wave 量子计算机的原理——量子退火算法[13]凭借其独特的量子隧穿效应，可跳出传统搜索算法易陷入的局部极值，可视为一类具有全局寻优能力的人工智能算法。受启发于 1999 年张焕国等人[14]提出的演化密码的思想，本团队深入探索一种不同于通用量子计算的量子隧穿驱动人工智能优化算法的量子退火算法进行密码分析和设计，并提出了如下两类技术路线。

(1) 第一类技术路线：2012 年本团队提出基于量子退火原理的 D-Wave 量子计算机对密码的设计与攻击可能会产生极大的影响[15]。2016 年本团队提出将密码设计或者攻击的问题转化为组合优化问题，并映射至 Ising 或 QUBO 模型。借助 D-Wave 量子退火算法的指数级解空间搜索的优势，实现密码的设计和攻击。通过该技术路线，本团队于 2019 年采用量子退火算法分解了最大为 20 比特[16]的 RSA 整数，实现了当年各类量子计算攻击 RSA 的最高实验指标。2020 年本团队优化 Ising 模型的参数，分解了一个更大的 20 比特 RSA 整数[17]。在此基础上，本团队进一步优化参数，2021 年分解了一个最大为 21 比特的 RSA 整数[18]。2022 年通过乘法表的高位优化模型，并利用量子退火分解了 22 比特 RSA 整数[19]。

(2) 第二类技术路线：由于量子计算的独特优势，2015 年本团队提出将量子计算与经典密码设计和分析方法结合的新型技术路线。基于独特的隧穿效应，优化密码函数设计和密码部件攻击中的数学难题求解。2017 年使用 Grover 算法结合侧信道攻击的方法攻击 ECC[20]；同年年底，通过量子退火算法结合经典密码分析方法在国际上首次完成基于 D-Wave 2000Q 的抗多种密码攻击的布尔函数设计实验[21]；2023 年 1 月通过该技术路线完成了 50 比特 RSA 的攻击[19]；同年，本团队使用量子退火算法对对称密码进行积分攻击，搜索到与经典算法攻击轮数持平的积分区分器[22]，初步验证了量子退火结合经典算法架构的可行性。

相较于通用量子计算机以及 Noisy Intermediate-Scale Quantum（NISQ）量子计算机有限的量子比特，D-Wave Advantage 具有高达 5760 个量子比特。基于 D-Wave Advantage 量子资源较多的优势，本文在 D-Wave 量子计算机上使用超出文献[5]的量子资源，开展量子退火算法结合经典算法架构分解 80 比特以上 RSA 整数的研究。首先，在 RSA 整数分解问题转化为最近向量问题（Closest Vector Problem，CVP）的过程中，本文通过提高文献[5，19]中 CVP 规模，从而提高搜索光滑对的能力。高质量 CVP 问题的解，能提高搜索到光滑对的概率。因此，本文旨在获得 CVP 问题更优的解，而在 Babai 算法[23]求解 CVP 的过程中，格归约基的质量会影响 CVP 的解。鉴于 Block Korkin-Zolotarev（BKZ）算法[24]非常适合处理高维度的格约化问题，本文使用 BKZ 算法对 CVP 的格基进行约化，获得较文献[5，19]中使用 LLL 算法[25]更优的归约基。基于该归约基，Babai 算法求解 CVP 问题可以获得较 LLL 算法归约基更优的解，即可以优化文献[5，19]中利用经典 Babai 算法对 CVP 问题的求解。在此基础上，本文利用量子退火算法的隧穿效应进一步优化 Babai 算法对 CVP 问题的求解，获得较 Babai 算法更优的解，提高光滑对的搜索效率，从而加速 RSA 整数的分解。最后，本团队在 D-Wave Advantage 上完成最大为 90 比特 RSA 整数 629367860625666765619139989 的分解。

2　预备知识

首先，将介绍本文需要使用的一些符号。$\mathbb{R}$ 和 $\mathbb{Z}$ 分别表示实数集和整数集。$\|\boldsymbol{a}\|$ 表示在 m 维欧几里得空间中向量 $\boldsymbol{a}\in\mathbb{R}^m$ 的欧氏范数。接下来，将分别介绍与本文工作相关的格理论基础，经典格算法以及量子退火算法。

2.1　格理论基础

定义 1.　格（Lattice）. 假设 $\boldsymbol{B}=[\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n]\in\mathbb{R}^{m\times n}$ 为 m 维欧几里得空间 $\mathbb{R}^m$ 中 n 个线性无关列向量构成的矩阵，则格 $\mathcal{L}(\boldsymbol{B})$ 表示为

$$\mathcal{L}(\boldsymbol{B})=\{\boldsymbol{Bx}:\boldsymbol{x}\in\mathbb{Z}^n\}=\left\{\sum_{i=1}^{n}x_i\boldsymbol{b}_i:x_i\in\mathbb{Z}\right\},$$

即格 $\mathcal{L}(\boldsymbol{B})$ 是列向量 $\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n$ 的所有整数系数线性组合构成的集合，其中 m 和 n 分别称为格 $\mathcal{L}(\boldsymbol{B})$ 的维数和秩，矩阵 $\boldsymbol{B}$ 称为格 $\mathcal{L}(\boldsymbol{B})$ 的基。

定义 2. 最近向量问题(Closest Vector Problem, CVP). 给定格$\mathcal{L}(\boldsymbol{B})$和目标向量 $\boldsymbol{t}\in\mathbb{R}^m\setminus\mathcal{L}$,搜索一个距离目标向量 $\boldsymbol{t}$ 最近的非零格向量 $\boldsymbol{v}\in\mathcal{L}$,使得对任意非零格向量 $\boldsymbol{u}\in\mathcal{L}$,一定满足:$\|\boldsymbol{v}-\boldsymbol{t}\|\leq\|\boldsymbol{u}-\boldsymbol{t}\|$。

2.2 经典格算法

LLL 算法[25]:给定一组格基,该算法可将该格基转化为一组正交性较好的近似正交基,并使得该近似正交基中的各个向量尽可能最短,具体步骤可参考算法 1。

算法 1. LLL 算法

输入:一组格基 $\boldsymbol{B}=[\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n]\in\mathbb{R}^{m\times n}$,参数 $\sigma=0.75$

输出:一组 LLL 算法归约基 $\boldsymbol{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n]\in\mathbb{R}^{m\times n}$

1. 对格基$\boldsymbol{B}=[\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n]$进行施密特正交化可以得到 $\hat{\boldsymbol{B}}=[\hat{\boldsymbol{b}}_1,\hat{\boldsymbol{b}}_2,\cdots,\hat{\boldsymbol{b}}_n]\in\mathbb{R}^{m\times n}$
2. FOR $i=2$ TO n DO
3. FOR $j=i-1$ TO 1 DO
4. $\boldsymbol{b}_i\leftarrow\boldsymbol{b}_i-\mu_{i,j}\boldsymbol{b}_j$,其中 $\mu_{i,j}=\lceil\langle\boldsymbol{b}_i,\hat{\boldsymbol{b}}_j\rangle/\langle\hat{\boldsymbol{b}}_j,\hat{\boldsymbol{b}}_j\rangle\rfloor$
5. END
6. END
7. IF $\exists\sigma\|\hat{\boldsymbol{b}}_i\|^2>\|\mu_{i+1,i}\hat{\boldsymbol{b}}_i+\hat{\boldsymbol{b}}_{i+1}\|^2$ THEN
8. $\boldsymbol{b}_i\leftrightarrow\boldsymbol{b}_{i+1}$,
9. 返回第一行,
   END
10. 输出 $\boldsymbol{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n]$

BKZ 算法[24]:BKZ 算法是 LLL 算法的一种扩展,该算法通过处理格基中固定大小的子块来提高格基的约化质量,尝试找到更短且更接近正交的向量组成的格基。具体步骤可参考算法 2。

算法 2. BKZ 算法

输入:一组格基 $\boldsymbol{B}=[\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n]\in\mathbb{R}^{m\times n}$,分块大小 $\beta\in\{2,3,\cdots,n\}$,施密特正交矩阵 $\boldsymbol{\mu}$ 以及 $\|\boldsymbol{b}_1^*\|^2$, $\|\boldsymbol{b}_2^*\|^2,\cdots,\|\boldsymbol{b}_n^*\|^2$

输出:BKZ 算法规约基 $\boldsymbol{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n]\in\mathbb{R}^{m\times n}$

1. $z\leftarrow0$; $j\leftarrow0$; $LLL(\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n,\boldsymbol{\mu})$;
2. WHILE $z<n-1$ DO
3. $j\leftarrow(j \bmod (n-1))+1$; $k\leftarrow\min(j+\beta-1,n)$; $h\leftarrow\min(k+1,n)$;
4. $\boldsymbol{v}\leftarrow Enum(\mu_{[j,k]},\|\boldsymbol{b}_j^*\|^2,\|\boldsymbol{b}_{j+1}^*\|^2,\cdots,\|\boldsymbol{b}_k^*\|^2)$;
5. IF $\boldsymbol{v}\neq(1,0,\cdots,0)$ THEN
6. $z\leftarrow0$; 阶段 j 对基向量进行 LLL 约化得到 $$LLL\left(\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\sum_{i=j}^{k}v_i\boldsymbol{b}_i,\boldsymbol{b}_j,\cdots,\boldsymbol{b}_h,\boldsymbol{\mu}\right)$$
7. ELSE
8. $z\leftarrow z+1$; $h-1$ 阶段对基向量进行 LLL 约化得到 $LLL(\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_h,\boldsymbol{\mu})$
9. END IF
10. END WHILE
11. 输出 $\boldsymbol{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n]$

Babai's 最近向量算法[23]:假设给定一个 n 维格 $\mathcal{L}$,一组近似正交基 $\boldsymbol{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n]$以及一个目标向量 $\boldsymbol{t}\in\mathbb{R}^n\setminus\mathcal{L}$。该算法可计算在近似正交基 $\boldsymbol{D}$ 下最接近目标向量 $\boldsymbol{t}$ 的整数系数的线性组合,具体步骤可参考算法 3。

算法 3. Babai's 最近向量算法

输入:一组近似正交基 $\boldsymbol{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n]\in\mathbb{R}^{m\times n}$,目标向量 $\boldsymbol{t}\in\mathbb{R}^n\setminus\mathcal{L}$

输出:向量 $\boldsymbol{v}\in\mathcal{L}$, $\boldsymbol{v}-\boldsymbol{t}$

1. 对$\boldsymbol{D}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n]$进行施密特正交化得到$\hat{\boldsymbol{D}}=[\hat{\boldsymbol{d}}_1,\hat{\boldsymbol{d}}_2,\cdots,\hat{\boldsymbol{d}}_n]\in\mathbb{R}^{m\times n}$
2. $\boldsymbol{v}\leftarrow\boldsymbol{t}$
3. FOR $i=n$ TO 1 DO
4. $\boldsymbol{v}\leftarrow\boldsymbol{v}-\lceil\mu_i\rfloor\boldsymbol{d}_i$, 其中 $\mu_i=\langle\boldsymbol{v},\hat{\boldsymbol{d}}_i\rangle/\langle\hat{\boldsymbol{d}}_i,\hat{\boldsymbol{d}}_i\rangle$
5. END
6. 输出 $\boldsymbol{v}$, $\boldsymbol{v}-\boldsymbol{t}$

2.3 量子退火算法

量子退火算法[13]是一种基于量子绝热演化原理的优化算法,它利用量子比特的叠加态、纠缠态以及量子隧穿效应来寻找问题的全局最优解。通过量子波动产生的量子隧穿效应可以使得量子退火算法具有穿越高势垒的能力,克服传统智能优化算法仅能翻越势垒,从而陷入局部极值的问题。量子退火算法可以跳出传统智能优化算法(以模拟退火为例)易陷入的局部极值(如图 1),逼近甚至达到全局最优解。

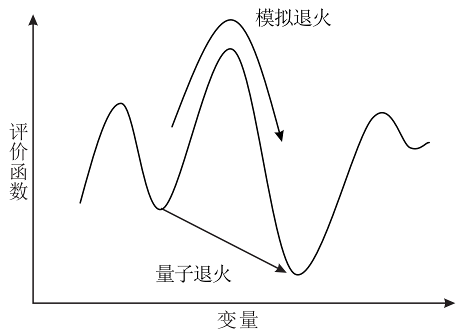

图 1 量子退火与模拟退火跳出局部极值的对比

量子退火的原理在于系统能够通过绝热演化保持在哈密顿量基态,从而实现对复杂能量景观的有效探索。量子退火算法通过将初始穿透场能量保持一个较大值,使得粒子具有足够大的波动,从而能够探索整个系统的能量空间。随后,缓慢地减小穿透场的能量,直至其能量为零。在减小穿透场能量的过程中,量子系统逐渐趋于稳定,最终粒子停留在基态,即能量最低态。因此,量子退火算法常被用于寻

找经典物理系统中能量最低态(基态),该过程可以通过由哈密顿量初态和哈密顿量终态构建的哈密顿量系统描述：

$$H(t)=\left(1-L\left(\frac{t}{T}\right)\right)H_{\text{init}}+L\left(\frac{t}{T}\right)H_{\text{final}},$$

式中,H_{init}为哈密顿量初态,H_{final}为哈密顿量终态,T 为演化过程的总时间,$t\in[0,T]$,L 函数为单调递增函数。

通过 D-Wave 量子退火机求解问题时,需要将该问题映射为 Ising 自旋模型。D-Wave 时变哈密顿量$H(t)$可以视为横向场 Ising 自旋模型,$H(t)$中哈密顿量终态由实际求解问题决定,可将其表述为以下形式：

$$H(t)=\sum_i h_i(t)\sigma_i^z+\sum_{i,j}J_{i,j}(t)\sigma_i^z\sigma_j^z+\sum_i\Delta_i(t)\sigma_i^x \quad (1)$$

其中 $\sigma_i^z=\boldsymbol{I}\otimes\boldsymbol{I}\otimes\cdots\otimes\boldsymbol{\sigma}^z\otimes\cdots\otimes\boldsymbol{I}$,$\boldsymbol{\sigma}^z$处于第 i 个位置(σ_i^x可类似定义),$\boldsymbol{I}$ 为二维单位矩阵,h_i代表第 i 个量子比特的权重,$J_{i,j}$表示第 i 和第 j 个量子比特的耦合强度,“$\otimes$”表示张量积,$\boldsymbol{\sigma}^z$和$\boldsymbol{\sigma}^x$为泡利矩阵,其形式如下：

$$\boldsymbol{\sigma}^z=\begin{bmatrix}1&0\\0&-1\end{bmatrix},\ \boldsymbol{\sigma}^x=\begin{bmatrix}0&1\\1&0\end{bmatrix}。$$

表达式(1)中$\sum_i\Delta_i(t)\sigma_i^x$表示初始哈密顿量,$\sum_i h_i(t)\sigma_i^z+\sum_{i,j}J_{i,j}(t)\sigma_i^z\sigma_j^z$表示终态哈密顿量$H_{\text{final}}$。通过对参数 t 的控制,可以使得整个系统的退火过程由初始哈密顿量向终态哈密顿量演化,若演化过程足够慢,则 Ising 自旋模型最终所处的状态为H_{final}基态。

3　量子退火结合经典算法分解整数

受启发于 Schnorr 算法[26]以及文献[5],本文通过量子退火算法结合经典密码算法的混合架构完成 80 比特以上的 RSA 整数分解。本章将分别介绍 CVP 规模对光滑对搜索的影响、CVP 解的质量对光滑对搜索的影响、量子退火算法优化 Babai 算法对 CVP 的求解以及量子退火融合经典算法架构分解整数的流程。

3.1　CVP 规模对光滑对搜索的影响

2023 年谷歌研究员[10]表示使用文献[5]中的亚线性量子资源难以有效地分解 80 比特以上的 RSA 整数,即整数规模的增加与 CVP 规模呈文献[5]中的亚线性增加关系时,即使完美的量子优化器+Babai 算法也难以完成 80 比特以上 RSA 整数分解。

光滑对的搜索是分解 RSA 整数过程中最重要且最耗时的步骤,本节的重点是分析CVP规模对光滑对搜索的影响。接下来,将介绍光滑对的定义：

光滑对：假设$\{p_1,p_2,\cdots,p_m\}$为 m 个最小的质数,如果满足 $u=\sum_{i=1}^{m}p_i^{e_i}$,其中 $e_i\in\mathbb{Z}$,那么 u 一定是 $p_m-smooth$。给定一个关系对(u,v),若 u 和 v 均是 $p_m-smooth$,那么可以称(u,v)为 $p_m-smooth$ 关系对;若 u 是 $p_m-smooth$,v 是 $p_{m'}-smooth$,则称(u,v)为 $p_{m,m'}-smooth$ 关系对。

接下来,以 48 比特 RSA 整数 220190119273699、64 比特 RSA 整数 11769431398084725929 以及 80 比特 RSA 整数 1034879359475633166138643 为例,讨论 CVP 问题规模与光滑对之间的关系。根据算法 3,我们可以发现 Babai 算法得到的距离目标向量最近的向量 $\boldsymbol{v}$ 如下：

$$\boldsymbol{v}=\sum_{i=1}^{n}k_i\boldsymbol{d}_i,\ k_i=\lceil\mu_i\rfloor=\lceil\langle\boldsymbol{d},\hat{\boldsymbol{d}}_i\rangle/\langle\hat{\boldsymbol{d}}_i,\hat{\boldsymbol{d}}_i\rangle\rfloor。$$

根据上式可以观察到k_i的值是将μ_i的值四舍五入,因此 Babai 算法求得的解向量可能存在不够精确的问题。如果同时考虑将 μ_i的值向上且向下取整,则可能获得比 Babai 算法距离目标向量更近的向量。接下来,我们通过穷举的方式同时考虑将 μ_i的值向上且向下取整,从而搜索每组 CVP 存在的所有光滑对数量,其中 CVP 规模设置过高会急剧增加 CVP 的搜索空间。当 CVP 规模为$Dim=10,15,20$时,分别随机生成各维度下的 30 组不同 CVP,其中光滑界为 $p_{Dim,2\times Dim^2}$。表 1 将展示各维度下单个 CVP 平均能搜索到 48 比特、64 比特以及 80 比特 RSA 整数的光滑对数量。

表 1　单个 CVP 的平均光滑对数量

CVP 维度	48 比特	64 比特	80 比特
10	0.66	0	0
15	41.9	1.2	0
20	145.6	25.5	0
30	36355	3570	26

注：维度为 30 的 CVP 的解空间高达 2^{30},遍历搜索其光滑对极其耗时。因此,仅随机搜索 1 个 CVP 的光滑对,并记录其数量。

根据表 1,可以发现：相同规模下的 RSA 整数,随着 CVP 规模的增加,可搜索到的光滑对数量也在增加,然而当 CVP 规模过小时,会导致难以搜索到光滑对。因此,可以得出结论：当 CVP 规模降低到一定程度,在整个解空间中也无法搜索到光滑对。

尽管规模增大之后，解空间会指数级增大，但可以搜索到光滑对。

3.2 CVP 解的质量对光滑对搜索的影响

给定一个待分解的整数 N，可以将其转化为 CVP 问题，假设通过计算可以得到满足如下条件的整数 e_i：

$$\varphi := \left|\sum_{i=1}^{m} e_i \ln p_i - \ln N\right| \approx 0。$$

为了便于计算，可以通过以下表达式定义 u,v：

$$u=\prod_{e_i \geq 0} p_i^{e_i},\ v=\prod_{e_i<0} p_j^{-e_i}。$$

经过推导可以得到：

$$\left|\ln\left(\frac{u}{vN}\right)\right| = \varphi,$$

由泰勒定理可以得出 $u-vN=vN(e^{\varphi}-1)\approx\varphi vN$。当 φ 越小，$u-vN$ 越小，那么 $(u,|u-vN|)$ 越有可能是光滑对。因此，CVP 解的质量将影响光滑对的搜索，即距离目标向量越近的解，越有可能是光滑对。

接下来，通过一个实例更加清晰地展示 CVP 解质量与光滑对之间的关系。随机选取一个 26 比特 RSA 整数 42904391，初始格基的维数设置为 7，并通过修改参数选择 10 组不同的初始格基。对于不同的初始格基对应的 CVP，通过穷举的方法同时考虑将 Babai 算法求解过程中系数的值向上且向下取整，即遍历所有的 $2^7=128$ 种不同情况的解搜索光滑对。对于上述 10 组不同的初始格基，可以得到 RSA 整数 42904391 的 10 组实验对应的光滑对分布情况如图 2 所示。

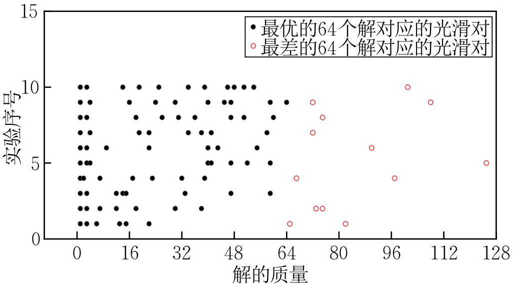

图 2 光滑对与解质量的关系

图 2 中横坐标表示每个 CVP 的 128 种不同解的质量排序，其中横坐标“0”表示距离目标向量最优的解，横坐标“64”表示距离目标向量最优的第 65 个解，以此类推。纵坐标表示 10 组实验的序号。根据上图可以发现 10 组不同的初始格基，搜索到的光滑对数量会有所不同。然而，对于每一组初始格基对应的最优的前 8 个解中均包含了至少 2 个光滑对。通过观察图 2，可以发现这 10 组不同初始格基对应的光滑对更加稠密的分布在前 64 个最优的解中，而最差的 64 个解中分布的光滑对则更加稀疏。因此，通过该例子可以清晰地观察到距离目标向量越近的解，越有可能是光滑对。

3.3 量子退火优化 Babai 算法对 CVP 的求解

本文通过量子退火（QA）算法优化 Babai 算法对 CVP 的求解，从而提高搜索到光滑对的概率。接下来，将分别介绍 QA＋Babai 算法对 CVP 求解的影响，并对比QA＋LLL＋Babai算法和QA＋BKZ＋Babai 算法对 CVP 的求解。

(1) QA＋Babai 算法对 CVP 求解的影响

本文采用 Babai 算法求解 CVP，并利用量子退火算法优化该解。首先，需要使用格约化算法对 CVP 的格基进行约化，获得一组近似正交且较短的格归约基，然后再使用 Babai 算法获得近似最近向量。根据算法 3，可以观察到 Babai 算法通过如下公式求解近似最近向量：

$$\boldsymbol{v}=\sum_{i=1}^{n} k_i \boldsymbol{d}_i,\ k_i=\lceil \mu_i \rfloor=\lceil \langle \boldsymbol{d},\hat{\boldsymbol{d}}_i\rangle / \langle \hat{\boldsymbol{d}}_i,\hat{\boldsymbol{d}}_i\rangle \rfloor。$$

由于解分布的随机性，若通过遍历的方法同时考虑将 μ_i 的值向上且向下取整，从而寻找相较于 Babai 算法更优的解，将会指数级的增加经典计算的复杂度。量子退火算法凭借其独特的隧穿优势，可以快速地逼近指数级解空间中的最优解，即快速的得到相较于 Babai 算法更优的解。

本文随机产生 50 组维度为 8，初始格基不同的 CVP 样本，并分别采用 Babai 算法以及 Babai＋QA 算法对上述 CVP 求解。Babai 算法与目标向量的欧式距离平方以及 Babai＋QA 与目标向量的欧式距离平方的对比如图 3 所示。

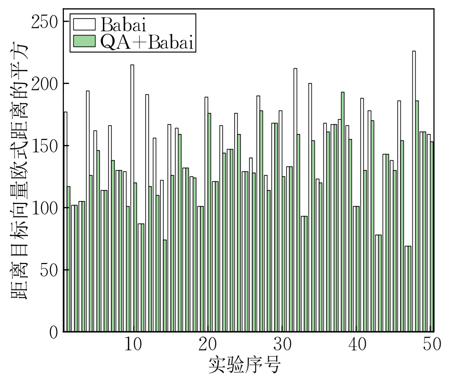

图 3 对比 QA＋Babai 与 Babai 算法求解 CVP

图 3 中横坐标表示 50 组随机样本的序号，纵坐标表示对 CVP 求解后与目标向量的欧式距离平方，其中橙色部分表示 Babai 算法的结果，青色部分表示用量子退火算法优化后的结果。根据图 3 可以观

察到：QA＋Babai 的解与目标向量的欧式距离在大部分情况下均小于 Babai 算法对 CVP 的求解。因此，可以得出结论：量子退火算法优化 Babai 算法对 CVP 的求解在大部分情况下均优于 Babai 算法对 CVP 的求解。

(2) 对比QA＋LLL＋Babai算法和QA＋BKZ＋Babai 算法对 CVP 的求解

通过 Babai 算法对 CVP 求解时，需要先使用格约化算法对其格基进行约化，获得正交性较好的格归约基，其中格归约基的质量会影响 Babai 算法对 CVP 的求解。基于 Babai 算法的解，量子退火算法可以进一步优化 Babai 算法对 CVP 的求解。

以随机选取的格基维度为 40、45 和 50 的 CVP 为例，分别选取 LLL 算法和 BKZ 算法对格基进行约化，对比基于 LLL 算法和 BKZ 算法的 Babai 算法求解 CVP 的解质量以及通过 QA 算法对其优化的解如表 2 所示。

表 2　不同方法求解 CVP 与目标向量的欧式距离

维度	LLL＋Babai	BKZ＋Babai	QA＋LLL＋Babai	QA＋BKZ＋Babai
40	2038	1710	1275	1063
45	3611	2012	2145	1937
50	4631	2973	3608	1794

根据表 2 可以发现：BKZ＋Babai 算法求解 CVP 得到的解优于 LLL＋Babai 算法的解。在此基础上，QA 算法优化基于 BKZ＋Babai 算法求解 CVP 得到的解也优于 QA 算法优化基于 LLL＋Babai 算法的解。

3.4　量子退火结合经典算法分解整数概述

本节将分别介绍量子退火算法优化求解CVP，光滑对搜索和使用光滑对进行整数分解的过程。

(1) 阶段 1：基于量子退火算法优化求解 CVP

给定一个待分解的整数 N，可以将其转化为寻找目标向量 $\boldsymbol{N}\in\mathbb{R}^{n+1}$ 的 CVP，其中初始格基表示为 $\boldsymbol{B}_{n,c}=[\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n]\in\mathbb{R}^{n+1,n}$，$c>0$ 是一个可调参数。

$$\boldsymbol{B}_{n,c}=\begin{pmatrix} f(1) & 0 & \cdots & 0 \\ 0 & f(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f(n) \\ \lceil 5^c\ln p_1\rfloor & \lceil 5^c\ln p_2\rfloor & \cdots & \lceil 5^c\ln p_n\rfloor \end{pmatrix},$$

$$\boldsymbol{N}=\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \lceil 5^c\ln N\rfloor \end{pmatrix},$$

其中，$f(i)$ 表示 $f:[1,2,\cdots,n]\to[1,2,\cdots,n]$ 的随机置换，每种不同的置换对应不同的一组初始格基，即通过更换可调参数 c 或者 $f(i)$ 的随机置换可以获得不同的初始格基 $\boldsymbol{B}_{n,c}$ 以及对应的目标向量 $\boldsymbol{N}$。

在求解初始格基 $\boldsymbol{B}_{n,c}=[\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n]$ 距离目标向量 $\boldsymbol{N}$ 最近的格向量 $\boldsymbol{b}\in\mathcal{L}(\boldsymbol{B}_{n,c})$ 时，首先需要使用 BKZ 算法得到一个近似正交的归约基 $\boldsymbol{D}_{n,c}=[\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n]\in\mathbb{R}^{n+1,n}$，它的施密特正交化向量为 $\hat{\boldsymbol{D}}_{n,c}=[\hat{\boldsymbol{d}}_1,\hat{\boldsymbol{d}}_2,\cdots,\hat{\boldsymbol{d}}_n]\in\mathbb{R}^{n+1,n}$。接下来，通过 Babai 算法可以获得距离目标向量 $\boldsymbol{N}$ 近似最近的格向量 $\boldsymbol{b}\in\mathcal{L}(\boldsymbol{B}_{n,c})$，该向量表示如下：

$$\boldsymbol{b}=\sum_{i=1}^{n}k_i\boldsymbol{d}_i,\ k_i=\lceil u_i\rfloor=\lceil\langle\boldsymbol{d},\hat{\boldsymbol{d}}_i\rangle,\langle\hat{\boldsymbol{d}}_i,\hat{\boldsymbol{d}}_i\rangle\rfloor,$$

根据上式，可以观察到 k_i 的值只是将 u_i 的值进行四舍五入操作，可能出现格向量 $\boldsymbol{b}$ 距离目标向量 $\boldsymbol{N}$ 较大的情况。因此，本文利用量子比特的叠加效应，对 Babai 算法中 u_i 的值同时进行上下取整，并同时进行编码，从而构造距离目标向量 $\boldsymbol{N}$ 更近的格向量 $\boldsymbol{b}_{\text{new}}$ 如下：

$$\boldsymbol{b}_{\text{new}}=\sum_{i=1}^{n}(k_i+y_i)\boldsymbol{d}_i=\sum_{i=1}^{n}y_i\boldsymbol{d}_i+\boldsymbol{b},\ y_i\in\{0,\pm1\},$$

那么，格向量 $\boldsymbol{b}_{\text{new}}$ 距离目标向量 $\boldsymbol{N}$ 欧式距离的平方可以用函数 $F(x_1,x_2,\cdots,x_n)$ 表示如下：

$$F(x_1,x_2,\cdots,x_n)=\|\boldsymbol{N}-\boldsymbol{b}_{\text{new}}\|^2=\left\|\boldsymbol{N}-\boldsymbol{b}-\sum_{i=1}^{n}y_i\boldsymbol{d}_i\right\|^2,$$

因此，$F(x_1,x_2,\cdots,x_n)$ 的值越小，那么格向量 $\boldsymbol{b}_{\text{new}}$ 距离目标向量 $\boldsymbol{N}$ 越近。通过将变量 y_i 映射为泡利矩阵，则可以构建 $F(x_1,x_2,\cdots,x_n)$ 函数的哈密顿量如下：

$$H_b=\left\|\boldsymbol{N}-\boldsymbol{b}-\sum_{i=1}^{n}\hat{\boldsymbol{y}}_i\boldsymbol{d}_i\right\|^2=\sum_{j=1}^{n+1}\left|N_j-b_j-\sum_{i=1}^{n}\hat{\boldsymbol{y}}_i\boldsymbol{d}_{i,j}\right|^2,$$

其中 $\hat{\boldsymbol{y}}_i\in\{0,1,-1\}$ 是根据单量子比特编码规则映射到泡利矩阵的一个量子操作符，映射规则由 Babai 算法中正交系数 u_i 与 k_i 之间的大小决定，即

$$\hat{\boldsymbol{y}}_i=\begin{cases}(I-\delta_z^i)/2,\ k_i\le\mu_i\\(\delta_z^i-I)/2,\ k_i>\mu_i\end{cases}。$$

基于量子退火算法的隧穿效应，可以快速的逼近哈密顿量 H_b 的最小值，从而获得距离目标向量 $\boldsymbol{N}$ 最近的一组格向量 $\boldsymbol{b}_{\text{new}}$。

(2) 阶段 2：光滑对的搜索

光滑对的搜索是整数分解的关键步骤，基于格向量 $\boldsymbol{b}_{\text{new}}=\sum_{i=1}^{n}e_i\boldsymbol{b}_i\in\mathcal{L}(\boldsymbol{B}_{n,c})$ 可以得到整数系数 e_i，并通过以下表达式可以得到 u,v：

$$u=\prod_{e_i\ge0}p_i^{e_i},\ v=\prod_{e_i<0}p_i^{-e_i},$$

根据 u,v 的取值可以计算出 $|u-vN|$ 的值。为了权衡搜索光滑对以及方程组求解的效率，本文选取的光滑对均需要满足 $p_{n,2n^2}-smooth$。因此，若 u 和 $|u-vN|$ 的因子均不大于第 n 个质数和第 $2n^2$ 个质数，那么将关系对 $(u,|u-vN|)$ 保存并作为用于分解整数的光滑对。通过更换初始格基 $\boldsymbol{B}_{n,c}$ 以及对应的目标向量 $\boldsymbol{N}$ 可以得到不同的向量 $\boldsymbol{b}_{\text{new}}$，迭代上述操作可以得到多个不同的光滑对。

(3) 阶段三：利用光滑对分解整数

基于足够的光滑对，可以对任意一个 RSA 整数进行分解。假设关系对 $(u,|u-vN|)$ 的光滑界为 $p_{n,2n^2}$，那么通过收集 $2n^2+2$ 个光滑对可以完成整数分解。

对于一个光滑界为 $p_{n,2n^2}$ 的光滑对 $(u,|u-vN|)$，它的布尔函数指数表可以表示为一个 $1\times(2n^2+1)$ 的向量。第 1 列表示 $u-vN$ 的值是否为正，若为正则它的值为 0，否则为 1。第 2 到 $2n^2+1$ 列则表示 u 和 $|u-vN|$ 对应的质因子指数是否为偶数，即将 u 和 $|u-vN|$ 分别分解为 $u=\prod_{i<n+1} p_i^{e_i}$ 和 $|u-vN|=\prod_{k<2n^2+1} p_k^{e_k}$。若 e_i 或 e_k 的值为偶数，则第 i 列或者第 k 列的值为 0，否则为 1。因此，根据 $2n^2+2$ 个光滑对可以构建一个 $(2n^2+2)\times(2n^2+1)$ 的布尔函数指数表，它可以看作为一个二元 $(2n^2+2)\times(2n^2+1)$ 的向量组，并且它一定存在线性相关的向量。通过高斯消元的方法可以计算出线性相关的 n 个行向量，利用上述 n 个线性相关向量对应的光滑对 $(u_j,|u_j-v_jN|)$，可以得到 $X^2=\prod_{j=1}^{n} u_j$，$Y^2=\prod_{j=1}^{n}|u_j-v_jN|$。最后，利用辗转相除法可以得出整数 N 的两个质数因子 $P=\gcd(X+Y,N)$，$Q=\gcd(X-Y,N)$。

4 实验结果与分析

4.1 实验结果

基于 D-Wave Advantage 量子计算机，本文完成了随机生成的 85 比特 RSA 整数 29617538717470095144079807 以及 90 比特 RSA 整数 629367860625666765619139989 的分解。D-Wave Advantage 拥有 5760 个物理量子比特，采用 Pegasus 拓扑结构，运行时必须保持在接近绝对零度的低温，并与周围环境隔离。在实验中，退火时间为 D-Wave Advantage 的默认退火时间 20 μs，退火的迭代次数为 2000 次。此外，使用 D-Wave Advantage 的默认退火 Schedule 参数[[0,0],[20,1]]，表示退火过程从时间 0 μs 开始，退火参数（磁场 s）从 0 开始增加，然后线性增长至 1，直到时间达到 20 μs。在整个过程中，退火的速率保持线性变化。

对于 85 比特以及 90 比特 RSA 整数，本次实验分别使用了 40 和 50 个逻辑量子比特（分别使用了大约 1200 和 1600 个物理量子比特），光滑界分别设定为 3200 和 5000。通过量子退火的隧穿优势，可以搜索到每个 CVP 的多个近似最优解。基于多个 CVP 以及对应的多个近似最优解，可以搜索到足够多的光滑对，并构建对应的线性方程组。通过对线性方程组的求解，可以计算出 X 和 Y 的值，具体结果可见附录。最后，我们通过辗转相除法 $P=\gcd(X+Y,N)$ 和 $Q=\gcd(X-Y,N)$ 分解整数 $N_1=29617538717470095144079807 43$ 以及 $N_2=629367860625666765619139989$ 如下：

$$N_1=1899511627931\times 15592186055597,$$

$$N_2=6398047085669\times 98368744743281。$$

4.2 实验结果分析

经过大量实验，我们发现对于整数分解问题转化的 CVP 问题，D-Wave Advantage 量子计算机的求解规模大约为 170 维。因此，理论上量子退火与经典算法结合的混合架构可以应用在更大规模的整数分解问题。然而，量子计算机的实际执行能力、量子态维持时间、求解的成功率等因素仍需要进一步研究。

随着量子领域的快速发展，可以考虑通过以下策略突破当前的主要技术瓶颈，提升量子计算在实际应用中的可行性。

(1) 增加量子比特数量和质量：研发更高质量的量子比特，优化制造工艺，以提高比特的稳定性和抗干扰能力，减少退相干效应。

(2) 减少环境噪声和提高计算精度：采用更先进的冷却与屏蔽技术，降低热噪声和外部电磁干扰，从而提高系统的整体稳定性。此外，可探索适用于量子退火的误差校正方法，以减少计算过程中的累积误差，提升计算精度。

(3) 退火 Schedule 的优化：研究自适应退火 Schedule 优化方法，根据计算任务特性动态调整退火时间和路径，以提高解的质量。结合机器学习算法优化参数的选择，提升计算效率和求解能力。

5 总结与展望

在当前量子计算攻击密码整体进展缓慢背景

下，RSA 整数分解进展每提升 1 比特都具有挑战性。本文提出一种量子退火算法结合经典算法的架构分解最高为 90 比特的 RSA 整数。鉴于 D-Wave Advantage 量子计算机的 5000＋量子比特的优势，本文通过消耗更多的量子资源，即提升 CVP 问题的规模，从而提升光滑对的搜索能力。随后，针对较大规模的 CVP，通过 BKZ 算法对格基进行约化，获得较 LLL 算法更高质量的归约基，从而使得 Babai 算法可以获得 CVP 更优的解。在此基础上，本文通过量子退火的隧穿效应进一步优化 Babai 算法对 CVP 问题的求解，获得较 Babai 算法更优的解，提高光滑对的搜索效率，从而加速 RSA 整数的分解。最后，基于 D-Wave Advantage 量子计算机，成功分解了随机生成的最高为 90 比特的 RSA 整数。

D-Wave 量子计算机发展迅猛且稳定，《Nature》的文章[27-28]表明 D-Wave 相干性等关键技术不断优化，对大规模问题求解具有优势。2024 年推出的 D-Wave Advantage2 预计将扩展至 7000＋个量子比特。D-Wave 公司最近的性能基准测试表明：4400＋量子位的 Advantage2 处理器在计算上比目前的 Advantage 系统更强大。因此，充分利用量子智能算法和量子位数较多的量子计算机攻击密码，探索通用量子计算机和 NISQ 量子计算机之外的技术路线是有意义的。未来需要重视量子隧穿推进 CVP 等 NP 难题求解的潜力，其全局寻优能力不仅可能成为密码攻击的关键，也会对信息科学领域产生影响。

参　考　文　献

[1] Rivest R L, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 1978, 21(2): 120-126

[2] Shor P. Algorithms for quantum computation: Discrete logarithms and factoring//Proceedings of the Annual Symposium on Foundations of Computer Science. Washington, USA, 1994: 124-134

[3] Jiang S, Britt K A, McCaskey A J, et al. Quantum annealing for prime factorization. Scientific Reports, 2018, 8(1): 17667

[4] Hegade N N, Paul K, Albarrán-Arriagada F, et al. Digitized adiabatic quantum factorization. Physical Review A, 2021, 104(5): L050403

[5] Yan B, Tan Z, Wei S, et al. Factoring integers with sublinear resources on a superconducting quantum processor. arXiv preprint arXiv:2212.12372, 2022

[6] Warren R H. Factoring on a quantum annealing computer. Quantum Information & Computation, 2019, 19(3/4): 252-261

[7] Arute F, Arya K, Babbush R, et al. Quantum supremacy using a programmable superconducting processor. Nature, 2019, 574(7779): 505-510

[8] Fujitsu Study Says Quantum Decryption Threat Still Distant (FSSQDTSD). https://www.hpcwire.com/2023/01/23/fujitsu-study-says-quantum-decryption-threat-still-distant/

[9] Acharya R, Aleiner I, Allen R, et al. Suppressing quantum errors by scaling a surface code logical qubit. Nature, 2023, 614(7949): 676-681

[10] Tanuj K, Noureldin Y. A comment on "Factoring integers with sublinear resources on a superconducting quantum processor". arXiv preprint arXiv: 2307.09651, 2023

[11] Aboumrad W, Widdows D, Kaushik A. Quantum and classical combinatorial optimizations applied to lattice-based factorization. arXiv preprint arXiv: 2308.07804, 2023

[12] Cui Fu-Xin, Wang Bei, Liu Yan, et al. Research status and prospect of quantum attack in public key cryptography. Cyber Security and Data Governance, 2022, 41(9): 3-12(in Chinese)
(崔富鑫，王辈，刘焱等. 公钥密码的量子攻击研究现状与展望. 网络安全与数据治理，2022，41(9)：3-12)

[13] Finnila A B, Gomez M A, Sebenik C, et al. Quantum annealing: A new method for minimizing multidimensional functions. Chemical Physics Letters, 1994, 219(5/6): 343-348

[14] Zhang Huan-Guo, Qin Zhong-Ping. Evolutionary Cryptosystem. Wuhan: Wuhan University, 2011(in Chinese)
(张焕国，覃中平. 演化密码. 武汉：武汉大学出版社，2011)

[15] Wang Chao, Zhang Huan-Guo. The influence of Canadian commercial quantum computer in cryptography. Information Security and Communications Privacy, 2012, (2):31-32, 35 (in Chinese)
(王潮，张焕国. 加拿大商用量子计算机对密码学影响. 信息安全与通信保密，2012，(2)：31-32，35)

[16] Peng W C, Wang B N, Hu F, et al. Factoring larger integers with fewer qubits via quantum annealing with optimized parameters. SCIENCE CHINA Physics, Mechanics & Astronomy, 2019, 62: 1-8

[17] Wang B, Hu F, Yao H, et al. Prime factorization algorithm based on parameter optimization of Ising model. Scientific Reports, 2020, 10(1): 7106

[18] Wang C, Hu Q, Yao H, et al. Deciphering a million-plus RSA integer with ultralow local field coefficient h and coupling coefficient J of the Ising model by D-Wave 2000Q. Tsinghua Science and Technology, 2023, 29(3): 874-882

[19] Wang Chao, Wang Qi-Di, Hong Chun-Lei, et al. Quantum annealing public key cryptographic attack algorithm based on D-Wave Advantage. Chinese Journal of Computers, 2024, 47(5): 1030-1044(in Chinese)

(王潮，王启迪，洪春雷等. 基于 D-Wave Advantage 的量子退火公钥密码攻击算法研究. 计算机学报，2024，47(5)：1030-1044)

[20] Wang Chao, Cao Lin, Jia Hui-Hui, et al. ECC fault attack algorithm based on Grover's quantum search algorithm with 0.1π phase rotation. Journal on Communications, 2017, 38(8): 1-8(in Chinese)
(王潮，曹琳，贾徽徽等. 基于 0.1π 旋转相位 Grover 算法的 ECC 电压毛刺攻击算法. 通信学报，2017，38(8)：1-8)

[21] Hu F, Lamata L, Sanz M, et al. Quantum computing cryptography: Finding cryptographic Boolean functions with quantum annealing by a 2000 qubit D-wave quantum computer. Physics Letters A, 2020, 384(10): 126214

[22] Pei Zhi, Hong Chun-Lei, Wang Qi-Di, et al. Research on quantum computing for practical SPN structure symmetric ciphers attacks using the D-Wave Advantage. Chinese Journal of Computers, 2025, 48(1): 234-248(in Chinese)
(裴植，洪春雷，王启迪等. 基于 D-Wave Advantage 的量子计算实用化SPN结构对称密码攻击研究. 计算机学报，2025，48(1)：234-248)

[23] Babai L. On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica, 1986, 6: 1-13

[24] Schnorr C P, Euchner M. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical Programming, 1994, 66: 181-199

[25] Lenstra A K, Lenstra H W, Lovász L. Factoring polynomials with rational coefficients. Mathematische Annalen, 1982, 261: 515-534

[26] Schnorr C P. Factoring integers by CVP algorithms. Number Theory and Cryptography. Berlin, Germany: Springer, 2013, 8260: 73-93

[27] King A D, Suzuki S, Raymond J, et al. Coherent quantum annealing in a programmable 2000 qubit Ising chain. Nature Physics, 2022, 18(11): 1324-1328

[28] King A D, Raymond J, Lanting T, et al. Quantum critical dynamics in a 5000-qubit programmable spin glass. Nature, 2023, 617(7959): 61-66

附 录.

基于 D-Wave Advantage 量子计算机，本文完成了随机生成的 85 比特 RSA 整数 29617538717470095144079807 以及 90 比特 RSA 整数 629367860625666765619139989 的分解。接下来，将展示本文用于 85 比特和 90 比特 RSA 整数分解的值 X 和 Y。最后，我们通过辗转相除法 $P=\gcd(X+Y,N)$ 和 $Q=\gcd(X-Y,N)$ 可以分解出对应的两个质因子。

(1)85 比特 RSA 整数分解对应的 X 和 Y 的值

N=29617538717470095144079807，

X=183150588547162283729247964437424939902529707829
7428186774550485542101304404509178614131571484170531
5018210029186944594925492855960339315560928676305014
4518973311067185358414759328606333503203749240234002
8215698760607223219213548455760620375318461041231379
2673585001705448596757830512063143778927677793677433
0064861108296811610548706712570977101872292223379009
1155154781751513208873530273697100929838151562073356
9330234868823682386593052866104434997811399677508826
6533462473367514249455605994425726427596710012431716
5325959622954205347171942981214536776464234560006013
6303094541991409925060495193065939530063301813406423
2767660647811193710029043820578894465316294343620258
8878784692844256706441145713665576657992237574338609
7375726852849608204300269965727201829545042495009092
4728864403941507784025229055125644639383024848211484
1177100106700306333352299050426516491334674204392375
2590203495185581418751117465690840579885001694317165
9515157351359727103530355363190959603672513439455841
2610654491410646035580613658499360519335372345555180
9088490522199169870402801563279188932426974968806194
8014279530535892846286854416435359652809520303940834
0044806787627979765049316273008649402052211890635315
9847920883574342861986165708220104828098651204270033
0340068899805400014451538314539392776017291064690079
1299609902543217902818559519161716264841981311664437
2199435139065567449480056553540294850053398133486130
3898490479607041098678483678681682192439368790383335
9878497329378316921526082327916382344146611189290869
2288047319841968688827224596554388341672333794457340
8725909677848305250616431143195495500910751262841585
2114712373705640615577859134206967501036442763103239
8181668687369365691637571782560757897985216513561166
5115438163812624409653549243382587283384438256054765
2603213209629354034436997512698862064993633231320698
6908732218435905347103536932632149818875300983274559
6141367574201728469463454021275844478127095091281668
5598233354919418586661444668551810057139985342458770
1171753015576334741325194813096832841215369734590940
4809167963805602419771630917267018986860094769661784
5105187604576210055398613899555983941142203567023031
1762570433101562922339744507399050054030277662058096
9198367934506316606674272511616649555813440832689172
9878627873301224754751552123447183304166484043785269
5172798628784537933353931227645224815274556321465874
2443355452653367578657208738725812762498572798955628
4561450133265266725632609567285380386376440463363296
7247339375785260477453455278951717594690570281710175
3791749668814973360019846183021045221059505221304214
2330039002596299675267108684671222405677664710276012

92720065476376769170070425206970144809546229328499 30
10049500321009752684429282474820139670328347377949 71
88780819973808841236780175709458043547406513268383 91
11163574667094861190496887271414814279461208528334 97
44160030613582640575051585070700028637322876175157 29
13252194732207665722195140745324177685459671258369 00
30965625509149803330943778426344578534165874271089 49
80975395882551579906217159786364093527158398432185 88
94737803171150315451852980992485241850257692622059 40
62159237218927978057442126453994683303715760569848 79
19001075746049942364463698370960809048713022099078 08
25055020547894332736523175816779854079086370852808 11
94176787024065729927952049921214800544764105758015 32
32993059090545901993519616261448680300921867341134 67
28460989524421165773010466306503941799409891516524 56
97576175339882557106778017819945171356933090300435 18
07787490021845399654324874438124546603319927549709 56
57640506689379645946776077471113856457103729800396 96
54294118512052053476359410944033524553151983823187 92
37817786231035538582303139850827433701738407437197 61
23151305692117841980218507953298634593200829794287 71
48204910458906543961238215267616942018378792870712 93
98184375771653653112598876389800036988866819454868 44
78723674092519896981060261116585238147526980515160 01
82207097613824925864235186626992453159917885170934 72
14484085664510919872941374840051715901861744932938 18
06889312599639192174929294131926521712839358817230 90
89228944043490893064525770683306526336036297854332 30
00694430290016432110101512214340344256996231696800 16
73431905099473348693495950004804865182251561371506 75
44444182443236426774088169090165712720421285738073 19
40428422307154173883332191580820022896595296111224 21
06253162514215754209592134076614464473464370657047 16
31787415811217660570373491918615207532665954197388 59
07357949618539490735481979624498432667106438616096 48
51820913786450279381869498163069670666168055780386 58
44082880029213715728707970261603778595872366191744 36
18802462775167677148154903899395774202126412419023 27
91631303985430206134598443513483493152199612809165 16
58835385265886408426380687478270010766096893315617 64
81627317019888149159187094877910671734064619829411 23
71612542875292300073847039719693301329952290507576 66
01981562011949434313053933920787381865176407412370 17
48543245753399225545302201806190669970217091743136 97
23537960602728610544665520962472740271414585875365 35
55252222435986441196531950545245864091587445686384 82
79459127931327182815191427597859782269690905613598 38
46864509363887403148806653574768218832487062957360 33
48502623137571618779520247897006270498528466152545 43
29279750057209332449132665107538943523414799050219 72
62255381942016930126496403881008840206264935367906 12
73631748568088866278706212134520487128789021407769 63
77499586105898493159171264210281013545430859630359 60
30609610180140567798174310900744552001135301984914 55
06959844306913560660090526382513861895432593542999 86
88914482981112211313829296108837862698393927178916 07
06380323736442168451398028509194501214891208421550 71
20110106220665530242791883677794358292814182881259 80
18493593924768958003990904389617248677668842313440 29
03135392568326781549501322876610582472111326356982 10
63741532933573380079239431784447433731596473976741 10
23428118926563565622430096613982187096935835244587 96
11327391789100431928529658719653558707515900497327 90
53390720917903122979149837277531395425891975055534 26
06250560644069361158654580607931558652206808112872 62
74692837074806683595795751449305140609351595525703 35
95033834868477918652008066530947299770394068026726 81
40227409474312274104611367855591308851493159627185 47
14969362757077129106313587769448508669574223613780 62
94088165951476314285616901228375936822966377204717 17
28038562786016344263847189919306091911067538041469 15
78450443385969464745711124611636828546386572225116 66
91190107446906218648859730108193693041612865868333 92
29573551175051277736137239129267647469430136482905 93
87965337059328499923735704382003133370571184269944 66
58479759798553979350602531706540064318485426202103 47
00824983104448706430677118321951363350566250677216 33
47347205207403480363293940629273745054525245693499 79
34981606749501134919392673240974941469751530930327 27
87331476393679000563070624972255003781716576430395 94
66625708122308428693565604515679965481786956512363 36
77547309114734707431406590623215993463698116214022 09
62220535100396130811156893138578882682390173608467 140
22139406931278412283645074914266606594862223839658 96
08913529550396761392170102356500871213604167011393 84
64035324286149682286838693828247598804962118200777 66
81069945492588419478937860555076460945158911100311 31
222557476689871691633015871047973632 8125；

$Y=$ 16678422031445691661348888703471777708884713329 0
12314903977584854277438379020134591201252213571824 30
98889221110116537098549128791023203462930929517499 85
11260345796749045430211605237912289278546918283408 87
14818837768154139645831030344026561053306920238868 53
52920331613793323993848334218142318660627624327728 41
68402968891298089401555564264343568047072635885791 62
45642686696409753048563600840091618466875784750248 34

4927600835309804649381203020635485568075661877845442
4592737911113696832537396569258150762906317167536148
5974343999140702779537771534587716051552558663768012
9387515680559261721673270254320365025005331447971038
9431777428203011018685711341548774816924065805295873
5545582196905181683232065098857175215640321429227119
4276679645392572492733475957616498202631095061246330
2020898089864719305011497415968527621845568324918750
3501509288506223559608140039805855519973126231601692
4555131133522908130361107429181836366523825573190870
4823343092577074556218226481633509614690092352878859
0399446514111451337195279349877779020403702584714590
9846285658561292005028665371893329032469385763496001
8944673589277837619263035217109454394736351510284318
1484658814804619067881293096170813759720503544502922
6397297506111945588045972658820734356499889803291997
0098672155246269854765684083739215120691986544111047
5669126707859831311736978603940221143898559775535124
6689427912599509857682702239339280220664677882725252
5861936149886727512908718790194763033999077842394196
8966068069815598716830820784767384786809091343491639
3177893485498155970421391799977537633102930725932553
5814312207304621474924602242714133090806744700515122
6908907992246036335521769611252049226432845119853046
1357910016627700382443648761190080762429664392669625
3999130240872217729256060847485164681003470677846255
1036175780654498562971101006589839916947608150923073
8708566954181067144832347453291663853756551910620777
3123000788880374792747418511087477611180654954071033
8400289413463331786276092029039481479932750799963882
8764580360539970959514477625034305246031081945624536
5542468740162723867516275576414709506117255813519990
8581038029491197949664890426057875976489798486496664
4849640497384825940002775468341905520073623869916632
3902111785805759189398343020481129578489915019784244
2149056550143551109412566192223139254329542920140592
6627555375051436384990700295546400200555403561756213
0047554120627991892531568844901414426118827669559089
3751461548473883557012916010506749445269989792569421
6658189300208403512842436220836524818964466652779869
4405702652218354887739282332148951744764134775132153
2411623011407253780773894969408469741171537211110189
3408738771105266831491988806805186931201488892230715
3493141852747382783342768705448098673207331258308315
9904512760367098005050146002990424075899743950457646
3210881889417889381326486283705146791554635675502411
6914029480678488799376385618617773166331851238424991
7720213480427635769870315160749429453704531173601330
0058262290703602507382037236989908838231880797836993
6866869478999877378922951343993498035807438690678959
6586751203886179358303119336104298364438821245530940
4667146336268757819755016624935866050246917136990677
6079622087254052861180106394769599638803143788024814
1309870345922668817180430950825850333399167049896600
9433258279023806965683097967789733693124485486973750
5703272509537310328829702994941050104558251339254660
1447564171045260995055315725275167430245705308469193
6279714284775109939648332661752578486882232702926233
2915229529131177785681741416892026806252946940452346
0911696343185271792990449689360348569611023875740163
7070650936362689724577102220920099871135478313119997
2269069800341722070338096945895717337730190295798499
0133510505668566434257577929762510996495402785779342
5128715674644839727461970602670199384138923827814489
1116020648365262019981287809097038852321312644697076
3985629947048648474915444192549998824076609391749056
3566112271351722331540366945629798500507932694776774
8056954485769451891358048644079799296197231505542552
3794312895786491247106710302883979490270403406389175
1543057503945987072231003838321463125087443368951016
0436591818717825959529441353132107185271911976784856
4229133766748708648565460785847827854911628584528405
9655405651984771312006621814346165368511908051350463
3115822049415511313703826421769370709491951723829046
7707562092120976078780021169481288744042770480974752
5466146990006905332207728118760026067632969529075216
9501698192940954150215700988808527705856796041288283
5329157529250087534339162843686774922733489396791199
5478220823290317736540138316224336267462276068208763
6877170209694392753029532603020366848127502228562041
9857397200620558754325972682707299819114220243538659
4812352992331834328915216560512974364515992502102180
3555867107931612906201525612895685435439412966816454
4816348035419399522985054853625700580402787885551900
8248397063480546446555226663185498014247451769666567
5579063574309953953510602296029693537792034785806382
8478788737970575962515420028100052295930585213651104
9385416684759418068448501522687024806496091582950247
6776540192970183245200267705212692086774631596652867
9797217419303169689773647474940456169176305748551383
8193343913391638844448042442747158147110718820306522
2618488807930616259711242600367633949018988474165990
3271998211760488748425412712851733256669427768019611
8775166945766700527012124524752121364480。

通过上述 X 和 Y 的值,可以计算出:

$$P=\gcd(X+Y,N)=1899511627931,$$

$$Q=\gcd(X-Y,N)=1559218605559 7。$$

(2)90 比特 RSA 整数分解对应的 X 和 Y 的值

N=629367860625666765619139989，

X=123962475945760283722866800059676074479420375335
8081007569068899434271711475317853212232802548986694
3994163119499998296529765115066870366082842252561102
3827063383622260122334270292320607216456612621798876
5799488130229057573796082282698538160187551754465572
7111643559086901034144359384207909268757901654713945
1559011187354239585670824959068487148888443974788667
7061728261560179675576255345358653371633658267665531
7453415237943909223123254557306144800490674439456776
1149002614045245690321510657266297738497365999073784
9382075119341884351371097757663787692409936371417398
8903694155400932503978338823421229637472069837443847
9150027377001285380213543274117563802771026434460140
4334497874415281685080348674609123228828429119377732
2139533015301604413720135028462351571587816283998970
4077940982986097792099606580713486212083074000411650
5985268274732849955424619868790724913722178901657184
3579184157942549741373611784048194583531843426678527
6268282111152492127477896092534283239948176090639050
8186352495692434760916117116655150631555120603367584
4500058156245153125882236347738378046761001120530146
3462121293648488077235530267731900992498694106029898
4199528232328658777323865301910426003159269981527180
6193794485226688362550924293856221674488170132695794
2962160708220261040796641414533550581505058001515593
7188955403049108650988784678226230216525141166126263
3949322493472856153251541386738141210484752775577243
7750700521284934546015436532156350622289073970336793
2559199276650289745091421214230827022948319810410819
7106266787928661543446743794215119609734920924820137
8563145000584479275998116551673729580627564562823243
1916236564152172014523615133207533606832423406072070
0911966059246940229428359520828809348259237230240868
0126588625798324957862816519012235475201872339365582
3809319945533832845157864743326309299963409401543871
5101997153429879321484435186066107398553595041716185
5645105610913832632702005073435581118543436773521405
6924733680080523347711477540988339367677752303059773
3729476991988073970399791706436171119066890774581105
3962796831446729793661303015811231117005380473157837
3411183859046728001692266870226893588338241534762598
1682274771779149582710746884730764080809270712246023
1296805249592998262063176610316375587075095694660621
5703535068303961998571205317672589177172969908020763
7089942577131717687873289352373419513323460228327699
3917940104125880820537799288299311923028531204051654
8840587138760063634191379160116600453139574955430319
6289381745348520379794565527754855437682239984883745
9088340924037631098882990692739829981147777412001671
2336417077197391872219189451892982440999283369481699
5835711134999994607805119722804952725404894179043578
3693559803892062944614390951101355736435728719835253
76208462442736621518765330292141114950030379796866606
7916841959179930247916945574864352065402991717292205
1841978607978647386076327778662098335142777131228956
0741603934928501321312730160964822358525065290071401
5208221679888278542538665828481876717931685318044261
2118743020819824098051643999081711001497750609833680
0129478341105133728642416518552665413324022619436444
8089828234801819585776963397315259231945445467812332
0126233088189568749207504639521706781103467619065843
1626365023138330133900746629405819670931119493133921
2099857018775617885219947234918043910868663518525155
4987004671399157467059744727927102548714516233210307
0157185093910789016897727055861130436722623767836002
8299121686026026304413996671093686450361248442883092
5103669349194937847178404263027504184796527052555079
6683418776342452343379297056636874576763604331184998
9907355385140400036071312685586189686856667823283256
7539108328845242381884822839985340612841382966297879
2010687148275307248856487652586663942077159721801917
8144271381565769835606740874839770471204939257395017
0029942413607489696120791283934826940855875313551638
9032044070058229219496266880119524385214111467322175
7350943618756872295151759627357006674010581957043890
3279040788037323761741885915873383331683470420649052
6475986200788766291326743380847935114364687547212702
1206953536031109808214378852332403617898113987599849
6050914035740784996744869205682559905944373823003586
4474795602047336944761406697435124697777474104203713
6142289390508876882609495190720228859395912996731295
7401298102805818917014227710322047581948869283075400
3605826835719869716390105995990587699141447689604481
3112252157084380321544032777636064166352723882755994
9973785762964774567681106596360849821999056953994051
7081103415565597114923483165801480745175317971667335
9047354471561789665848899548151558377414846614047236
0980246739287042185895036572037454941711236552112399
9268515397532382514598486608745785542012333053575516
2484636452196055046680547608346536208175299365597063
1321028617228611854030051673299289414621134228315552
9297886990005885042080545995993665847053959548417428
7074163168063379778613883682473953481978941553214338
4597094238350849909591994948537215638886947006781064

755479682232937274623140001426788948717173126705160125159663094309139444292005396821819984717074008662358497257128322854284761442406259567339407193768521301370818255066651616409712650183986989351190902847960609182357406239246315812325022153994412168827181664009283907776130000491245677459978671829212654339337758899267835543110327884255838511096335426417826282778462800493781811152571617357743651207271580097668953404276004163529260069783197115942102900201609577028687873000927576127123549140319036154514260602681433915249264375560829520954341543793316694894861424160081511433503225962590941388771273699940741329787948815066058309757324134842352147942151684988233947082778474432689319446197234148935983814530623527794895197890745391712114326229079299997307231741881938094638229243134117340846820238939131259403957275438933332708357066278037359296056900722396968054282107335974274904346810793821290344095770318223250625682196511147599575114786989054336510749681764406083916983664163217515404355102322650002963696564481799412168449900835332723024506268262707253754945480224438710078879878035974001937640414660194320870639222565134815630405339587046654548851373394771496485761031467373068182965904592896007144570124262171345018353143377940970605742118388712896002499624202445123174628138953686450128840364565590777264041006966329193637662947592697948660113753083550976156852835781044786300765149354950251618350771755792845741778660137831684176004347398952713832450254647223474778246378022905965154298098761391309993658003592353695205250315682256487199687242942131568254733328868676186227607860033510889358448062356593110938364457141414691647259078441105060082435648967296253113270492198817987275156460982443257681530392304294964106429528000172539198822520523195050547727658890278060269038653496477331584733789153433554699220706972145795379625290647522710095744694109287047189047772665124281386876806214803403782048408034301694622029944197903901686000916309622791981623739804405345718028387490169510719105369973638756255047113898413996802690205262406394093938032947754269507603464197382945296180905349460819080200488982892221453426500016659204231613525105172430893591788095875920142455777475214265203415160561884239684992688486044141943200481760890040991160052917089591145137559047999950962907212449246358997425118196888818926899372833029313453939986580786257626711433730594424246005265041775923553749609060395565002788215496445294256474036155281351982497061732595868123581347814531880265540249987596471048190434797112809972592522032185261413506009561861201274497341326962212985305700785423014392957059573005637607608308735823712667325837963614560871135289561120451352761078213375762361249957031512894478235675219954971506768866141611124082369925481817469793966378417580874933037118457555041269675311738499482812460870184092055317939653953880706834037281029045543190160648486426356744532751567356894129361864507785873249127982100455519789063721197781222452536721775262271827207721851287164859910221434086608353762858883913883297446974924020650968860057999714045978077018110375442086639321594427910660977975473165035678271651150407450181240235631601916048227528720944610237193598230773012470945478030761323928028019867701126197666990051780838841685902405081660568210707319993752937185719005145795115912821498512434946464010464737621405086546891983174369750962762823629725930876997109161660642580327052928154432859872542360512222495528808646428964430910105485100708414713116141568188518928162861955259437890822870072814824287843299863578336834329559532426786623202649554941921110714662240843461141376933405831929698926194609791463477865733406503575883779728963703722752807126184496431958678236550118250888785631502742035081610083580017089843751;

$Y=3619214549733950444589819375143980768243644170119011419687140761095278760530130922531493756393140745381712054645010899408838007705823050517862750934666325903233159223007345611480812517722066525284116492343461765496739206348856556741543745486563539927311641726812521602677492958312474623863338441518021812123957597146755618028691802012225540138638695408332862170633840989198619037347121217664496459280176527302872039963383586166333486854321915193090353751009674173859529109000020617402437643724362111044026898154362555645502372053741171011663737936863060640849835431229100394881688283169332994648282269460047783405230963215556794186458494904997759218524177571234861209050408410380230337306706741366679814004534666037913242064539518553292640850176857575035284897424676308713633580325044714340447567373159433948108969694172521782106431254112306103736896992737202456042661137813247120905210466469472015640667547955025527418922984677839924281978837614914199073901080182091112000952562681511728525608223299556976158793174570237685426564828884252575747578790889450865535911033106247854208320176581802962866902299643731301532030366690496805167860136550653786343540641213476068465383060699611074103561357635574952600380466844568425371862448278717739051 2$

0758392108089347635111130441461960125272491792149575
7326575590117650932339922374536173683967110376942539
8247170691187148615598340970070955881761918608085205
6774698348537940658537731914154530921128619919532014
9099535885166530323514066161624289562535739278418748
0169453381736839680893765838363121947423539333627022
0463522689124348073536735485336253453751157551602696
8062274842545371865741798342356039211765838053004102
9461449844635916370751994834015048907007504298553616
0205586292803077658687267860165643136223984035256213
8570959370531251923089994797859324582611533031455263
1995546085275864857585716889809468565410031322956735
5270740889313420441497989960329911128657528930283214
1751238778166755264048005163734674571844051540719136
7809435846471016492071068978256073961984821783157603
1880745383277565972699798054641792453772081242549234
0536199080455793974333710173110547907847277762504208
3096882143587707053016662629299816715866345932680978
3199536409371114970630344573969604534621748262183581
7692187602939086382817117949591079104384715324846534
3770477829179134508649621446372827684359435072029548
0689140313030652573664129415986787543988817040750331
7381043347320001115114062500472347043959340504800844
9225145560255223669701075929098850792014155125358276
0530798498889422559926313545989556790931942784841583
0708401481681765725676038550097867861148891873568876
7638071433436245531489777023667397359185327865408709
1426583780257302639647897847444482469050276939537779
8281698963607025844173533086964243330344956158346195
3670115950191546695568036478410317290690879336126364
4155644257220838947921117785794522781808607506056790
4495055867695897633766954184692720741202518060364945
8975063796253173177305212212885521567514702950445197
2827694120214514738628575687134483799512216647471322
1836518128009368286165644783558846265074611305741522
9285255428220131237791404146312155847139672092075056
6680194393730236771418339756343216871648810895944689
7883842855004653853298387999053147050539488518486053
5222561909619179492806664728326036099749280636922550
6729857152220310293405693518941817725134810615341170
5895878301363115060514491122317428070000590797243223
6401844396298016189742600537938215328522045544831803
3180486923380962083281669166302071447259406511563642
4082814137863204213677453943475781919051253958936859
8224972471214262405208395793221993563785738874850316
3204323733344979460069295802284998297388500443400635
7702460168133481382647416803997595836404325856960787
5232757917231554449368649112799098278473342947830322
7835581558210691554054603045083007392370061473136686
2337194631333272343496945550601863156293392555250481
6147881951626949661211262286498973675976367176324321
3294578396438629495856429656000135889978018136955788
0042259975211190883176592053929628710478044242478165
2500555731231274286259118812799276861115175875079488
3340043918965155589264314923349308420453289630962982
1872871426683648270092948409425421961613797089256007
8617776838077146914482986426156902914077499165791666
3797727814610745085239114937764147840789563398924317
0119956856601052365460642016075030204300729174040493
0886037230185599385405817717049781512433245439479167
5692879407402548811751701960103578023883084775004409
0656572816893710971382177048565140891921270475853940
7433599439281466192877272653083094127688743586578278
6159927756370242341067310766028561035731830607840665
4418054894971311946877718751435374316383780806997710
3445357218347318997193902715761926936295573832294499
2664166287715625866462457425556753680201578642813538
9588477793188449093271725140098650527634638602970025
1652338969935018376132802364210189045592041058013257
3642032744200456225999010724244358549579019417494877
5827871867857757525785030203279666095752497689876131
2832744687996446577549876614601679843620175553990778
7254120984514562479123876323631944447620741948466849
36222020071346526076577401215231669168278308111227492
9481324754979921912665053695882897800008999926103337
9412646720935801857573963606613769451599514711146521
7238622612442914683108047081362447808758677058179296
5010774214545805261379967236805444693921870802345121
0795612022518802558406007046654501177581216288667680
1852039114408093445300453708136455409687446220988257
4339207732888757319556354594786696355403601798624897
6918710434537173874538941415252781062631965278659315
2048246248074241856402450454784702613822630063509351
6329229517533352396890217780186153070905324081401663
1535930692312622326125206427940254569854549410568996
0746660538242373443080796170574467632933732095325751
4848917882560261418580692012851274526263350695406913
2609830303315571598092192469813060126782728132685339
9624345857798328870780404004520356412348124121145943
3959409478637643148143480448691452821783901974261112
4497767114965031954480961221173144931516710173224857
3692803535075856845002756387701610989871631249898894
5481329626509035866316498492265544330211373381222690
9584517296856999945281610963850934798250021022145045
4055386764230087727575596482635061633955358764361040
5297405922717153711109981741073244783038563113199743

1672638356783137538742847767522202763832872216086946961842030157799189795241045274265253968793109253271876975867781694120506753725349583177934826203263071464456192。

通过上述 X 和 Y 的值，可以计算出：

$$P=\gcd(X+Y, N)=6398047085669,$$

$$Q=\gcd(X-Y, N)=98368744743281。$$

HONG Chun-Lei, Ph. D. candidate. His main research interests include cyberspace security and quantum computing cryptography.

PEI Zhi, Ph. D. candidate. Her main research interests include cyberspace security and quantum computing cryptography.

WANG Chao, Ph. D., professor. His main research interests include artificial intelligence, cyberspace security and quantum computing cryptography.

Background

This study belongs to the intersection of cryptography and quantum computing, with a focus on the attack of RSA public key cryptography by quantum computing. RSA is widely used to protect the security of internet data transmission, especially in important fields such as government communication et al. In addition, RSA is also used to implement digital signatures to ensure the integrity and authentication of data transmission. Theoretically, quantum computing can break RSA in polynomial time, but due to the limitations of current quantum computer hardware and error correction technology, the progress of quantum computing in actual attacks on RSA is still slow. Origin Quantum company has pointed out that in the context of the overall slow development of quantum attack cryptography technology, even 1-bit improvement in RSA integer factorization capability is also highly challenging.

In July 2023, Google researchers stated that it is difficult to factorize RSA integers greater than 80 bits using sublinear quantum resources, even with a perfect quantum optimizer applied to Babai's algorithm. In August 2023, IonQ researchers stated that completing the factorization of RSA integers greater than 80 bits using the QAOA+Babai algorithm on current quantum computer hardware is challenging. Currently, the progress of attacking RSA based on real quantum computers, both domestically and internationally, has not yet achieved an attack of more than 80 bits.

Quantum annealing, with its unique quantum tunneling effect, can escape from the local minima that traditional intelligent optimization algorithms are prone to easily, and quickly approach the global optimal. Considering that the quantum resources of D-Wave Advantage have reached more than 5000 qubits, this study uses more than sublinear quantum resources, and employs a hybrid architecture that integrates quantum tunneling driven artificial intelligence optimization algorithm with traditional cryptographic algorithms. Efficiently factoring RSA integers up to 90-bit by beyond the scale of sublinear quantum resources on a D-Wave Advantage: $629367860625666765619139989=6398047085669\times 98368744743281$. The experimental indicators significantly exceed those of Fujitsu, Lockheed Martin, and Purdue University, which factorize 8-bit, 13-bit, and 19-bit RSA integers.

The experimental results show that it is meaningful to study the quantum intelligent algorithms and quantum computers equipped with more qubits to attack cryptographic algorithms. In the future, it is essential to focus on the potential of quantum tunneling to facilitate the resolution of NP-hard problems such as the Closest Vector Problem (CVP). Its capability for global optimization could prove crucial in advancing cryptographic attacks.