量子计算能否破解公钥密码？

从 Shor's 算法到逻辑比特：一条难以逾越的鸿沟

文丨徐令予

在近三十年的科技叙事中，量子计算的能力被赋予了某种“神话色彩”。其中流传最广的断言莫过于：量子计算机一旦问世，现有的 RSA 等公钥密码体系将瞬间土崩瓦解。

这一判断并非空穴来风。Shor's 算法在理论上确实表明，整数的质因数分解可以在量子计算框架下高效完成，严重威胁公钥密码 RSA 的安全。然而，从“理论上可行”到“现实中可为”，中间横亘着一条极其深邃的鸿沟。

需要明确的是，本文的目的并非全盘否定量子计算。事实上，在模拟量子系统、寻找新型材料或优化特定算法等特殊领域，量子计算已经并正在展现其独特的潜力。然而，一个耐人寻味的现实是：作为量子计算数十年来最引人注目的“卖点”以及争取研发经费的最大旗帜——破解公钥密码——在实验层面却几乎交出了一张白卷。这种理论上的“降维打击”与现实中的“寸步难行”之间形成的巨大反差，正是我们审视这一领域时最需要保持冷静的地方。

本文试图把这一问题拆解清楚。首先说明 RSA 的安全性究竟建立在什么基础之上；其次解释 Shor's 算法到底改变了什么；然后回到实验现实，看看三十年来量子计算在整数质因数分解上有何进展；最后将问题归结到一个更根本的层面：是否能够构建有足够数量的逻辑量子比特，去支撑真正可扩展的量子计算。

一、公钥密码 RSA 安全性的基础

RSA 公钥密码的安全性，根植于一个极简的数学事实：将两个大素数相乘很容易，但要把它们的乘积重新分解开，却极其困难。这种“正向容易、反向很难”的计算不对称性是 RSA 算法的核心。

RSA 的基本思路其实并不复杂。先选两个很大的素数 (p) 和 (q)，把它们相乘得到一个大整数 (N=pq)。这个乘法很容易做，但反过来，如果只把 (N) 交给别人，让别人重新找出当初那两个素数 (p) 和 (q)，事情就会变得极其困难。

在 RSA 中，公钥里最核心的参数之一，就是这个大整数 (N)。公钥可以公开，任何人都能看到它；但私钥的生成却离不开 (p) 和 (q) 这两个素数。也就是说，谁如果能把 (N) 分解开，谁就有可能进一步推出私钥；谁分解不开，谁就只能看到公钥，却推不出私钥。

与此同时，RSA 的设计保证：用公钥加密的信息，只能由对应的私钥才能解密。因此，只要无法从公钥推出私钥，加密后的信息就无法被解读，其安全性也就得到了保障。

因此，“质因数分解困难”并不只是一个数学问题，它就是 RSA 密码安全性的根基。

可以先看一个简单的例子。如果 (N=21)，几乎任何人都能一眼看出：$21 = 3 \times 7$ 。也就是说，这样的“公钥”根本藏不住任何东西，因为它太容易被拆开了。

但 RSA 真正使用的并不是 21 这种小数，而是长达 1024 bit、2048 bit，甚至更长的大整数。随着 (N) 的位数增加，把它分解成两个素数的难度会上升得非常快。

到目前为止，人类公开完成的最大整数分解，大约是 829 bit 的 RSA-250。那已经需要国际团队动用大量计算资源；如果把这些计算量折算成一台单个处理器来做，大约要连续运行几千年[1]。更重要的是，这类计算并不能靠简单地堆处理器来获得理想的并行加速。

而这还不到 1024 bit，对于今天广泛使用的 2048 bit，难度会再跃升好几个数量级，这是一个在现实世界难以完成的计算任务。所以，RSA 这类公钥密码的安全性，本质上是一种基于计算复杂性的现实安全。

二、Shor's 算法到底改变了什么

1995 年，Shor's 算法的提出，第一次从根本上动摇了 RSA 公钥密码安全的基础。

它的意义，并不是“破解了 RSA”，而在于改变了人们对这个问题的认知：质因数分解之所以在现实中几乎不可行，并不完全是问题本身的原因，而是与我们所使用的计算方式有关。随着数字位数增大，质因数分解在经典计算机上计算量迅速增长而变得不可行；而在量子计算的框架下，它在原理上可以在多项式时间内完成。大致估算，原本需要数千年才能破解的 RSA 密码，有可能在数天内被量子计算机破解。

从实现机制上看，Shor's 算法并不是直接去尝试分解整数，而是把问题转化为一个“寻找周期”的问题。一旦这个周期被找到，原来的分解问题就可以被还原出来。关键在于，这个周期隐藏在一个巨大的数值结构之中，在经典计算机上难以高效提取；而在量子计算中，可以利用叠加态同时处理大量可能性，并通过 Quantum Fourier Transform 将其中的周期结构迅速显现出来。

需要强调的是，傅里叶变换本身并不是量子独有的，经典计算机同样可以高效完成类似计算；真正的区别在于，量子计算能够在叠加态中同时处理指数规模的数据，并在一次计算过程中提取其中的整体结构。这一点，是经典计算无法做到的。

换言之，Shor's 算法并不是找到了一种更高效的经典算法来破解 RSA，而是彻底改变了思路：如果换一种计算机器，原本“几乎算不动”的问题，其计算难度可以发生本质改变。如果存在一台足够强大的量子计算机，那么 RSA 在原理上将不再安全。

但这个结论隐藏着一个极其关键、同时也最容易被忽略的前提：这台计算机必须是可以运行 Shor's 算法的大规模、稳定、可扩展的量子计算系统。离开了这样的前提，Shor's 算法就只是一种优美的数学理论，它不会有任何实用价值。

因此，从 Shor's 算法提出后，构建相应的量子计算系统就成了破解 RSA 密码的关键。

三、破解 RSA 的实验结果令人十分失望

自 1995 年起，世界多国的大学和科研机构在量子计算机的研发中投入了大量的精力和时间，为了破解 RSA 可谓不惜一切代价。但是理想很丰满，现实很骨感。

近三十年来，量子计算在“分解质因数”上始终无法突破 21=3*7 。相关论文和实验报告中常被提到成功“分解 15”和“分解 21”，虽然这些结果显得非常寒酸，但背后的真相更为不堪。因为这类实验通常依赖针对特定数字的高度定制化电路、问题特殊化、编译优化，离“输入任意一个整数，通用地执行 Shor's 算法并得到质因子”还有本质距离。

破解 21 都做不到，那么破解 2048 位 RSA 要分解一个多大的数字呢？看看下面的图片就能明白什么叫异想天开了。

RSA-2048具有617个十进制数字，共2048bits。是目前最大的RSA数字，有20万美金的悬赏用于对RSA-2048的因式分解。

```
RSA-2048 = 25195908475657893494027183240048398571429282126204032027777137836043662020707595556264018525880784406918290641249515082189298559149176184502808489120072844992687392807287776735971418347270261896375014971824691165077613379859095700097330459748808428401797429100642458691817195118746121515172654632282216869987549182422433637259085141865462043576798423387184774447920739934236584823824281198163815010674810451660377306056201619676256133844143603833904414952634432190114657544454178424020924616515723350778707749817125772467962926386356373289912154831438167899885040445364023527381951378636564391212010397122822120720357
```

为什么连破解 N = 21 都如此困难？直觉误区是 21 只有 5bit，应该很容易被破解。但实际难点在于：Shor's 算法需要实施可逆模运算、量子傅里叶变换和一整套受控操作，所以必须有一组可以长期保持状态、能够反复操作的量子比特。

量子比特有点像经典计算机 CPU 中的寄存器，是所有运算绕不开的地方。没有一组寄存器，再复杂的程序也无法运行；没有一组稳定可靠的量子比特，Shor's 量子算法寸步难行。即使破解 N=21，理论上至少需要约 10 - 15 个稳定可靠的量子比特，但实际上能有个位数的稳定可靠量子比特都是难于上青天！

尽管媒体经常报道，“某某机构已经做出几十个、上百个量子比特”，但它们都不是稳定可靠的量子比特，它们离运行 Shor's 算法的要求差之甚远。破解 RSA 密码困难重重，缺乏足够数量合格的量子比特始终是问题的重中之重！

四、噪声是逻辑量子比特的天敌

现实中的量子比特非常不稳定，它们会不断受到环境干扰，原有的量子状态会迅速退相干，因而就无法像经典寄存器那样直接作为可靠的运算单元使用。为此，人们引入了 Quantum Error Correction 的概念：通过把大量物理量子比特组合在一起，构造出一个在计算过程中能够维持其状态的“有效比特”，这就是所谓的逻辑量子比特 （logical qubits）。

这正是量子误差纠正的基本思想：用大量物理比特组织成一个逻辑比特，并在整个计算过程中持续进行纠错操作，使这个逻辑比特在统计意义上保持稳定。

从理论上看，这条路是可行的。所谓的“阈值定理”表明，只要单次操作的错误率低于某个阈值，就可以通过不断增加冗余，把逻辑错误率压低到任意小，从而支持任意规模的计算。也正因为如此，很多人认为，量子计算的问题只是工程问题，而不是原理问题。

但关键在于，这一结论依赖一组非常强的前提：噪声必须近似局部、相互独立，误差率不能随着系统规模扩大而显著上升，纠错过程本身也必须足够可靠。这些条件在数学模型中成立，但在真实物理系统中，却很难完全满足。

实验结果表明：一方面，随着纠错规模的增加，确实可以在一定范围内降低逻辑错误率，这说明纠错机制本身是有效的；但另一方面，一些更难处理的噪声开始显现出来，例如跨多个比特的相关错误、偶发但一次性影响大量比特的突发错误，以及测量与控制过程中引入的系统性误差。

于是，一个根本性的矛盾浮现出来：为了得到更稳定的逻辑比特，必须引入更多的物理比特和更多的操作；但系统越复杂，新的噪声来源也越多，纠错本身就变得越来越困难。从这个角度看，逻辑量子比特之所以难以实现，并不仅仅是因为“技术还不够成熟”，而是因为它陷进了需要不断对抗变幻莫测的噪声的困境之中。

结语

公钥密码 RSA 的安全基础是大整数质因数分解。近三十年来，量子计算在分解质因数上始终无法突破 21=3*7 ，试图破解 2048 位的 RSA 密码根本无从谈起。

量子计算破解 RSA 密码，在数学上已然成真，在实验中步履维艰，而在工程上则仍是空中楼阁。这场争论的核心命题，早已超越了 Shor's 算法本身的有效性，而聚焦于一个更为本质的拷问：大规模容错量子计算，究竟能否在真实物理世界中实现。

注释

[1] 2020 年，国际团队完成了 RSA-250（829 bit）的分解。公开资料通常将其总计算量表示为约 2700 CPU 核心年。正文中将其换算为“单个处理器运行几千年”，只是为了帮助读者直观理解规模。整数分解所用的数域筛法虽然可以有某种程度的并行计算，但并不能实现理想的线性加速，关键步骤还受到内存与通信开销的限制，因此不可能做到“处理器多一万倍，时间就缩短一万倍”。

徐令予 作于美国南加州 （2024 年 4 月 21 日）