量子计算会威胁比特币安全吗？（中）

——谷歌白皮书的结论有验证吗？

文丨徐令予

在本系列的第一篇文章中，我们已经讨论了谷歌白皮书对量子计算资源的估算：在什么条件下，量子计算有可能对公钥密码构成威胁。

但仅有“估算”还不够。一个更关键的问题是：这些结论的可信度究竟如何？换句话说，谷歌白皮书的结论有验证吗？

与通常的研究做法不同，这篇白皮书并没有公开其电路细节，却声称通过一种被称为“零知识证明”的方法，使其结果仍然可以被外界验证。这看起来与科学研究中“公开方法、可重复检验”的传统原则形成了明显的反差。

那么，在不公开细节的前提下，所谓“验证”究竟意味着什么？我们究竟能相信什么？本文将围绕这一问题展开。重点不在于判断白皮书结论的对错，而在于弄清楚：他们所采用的这种验证方式，究竟能够证明什么，又不能证明什么。

仔细阅读白皮书，特别是附录部分之后可以发现，作者并没有构建任何量子硬件，也没有在真实的量子计算机上运行破解相关密码的算法。他们的整个工作，实际上是在经典计算机上完成的。更具体地说，他们做了三件事情：

一、电路设计

在讨论这篇白皮书之前，有一个需要先澄清的基本点：这里所谓的“电路”，并不是实际的硬件装置，而是一种对计算步骤的描述。

更直观地说，它更像是一份“程序”，或者一张“操作说明书”。它规定了：如果未来有一台理想的量子计算机，每一步应该如何操作，先做什么、再做什么，用到多少“计算单元”（这里称为量子比特），以及总共需要多少次基本操作。

因此，这一步的工作，本质上是设计一种计算方案，而不是实现一台真正运行的机器。

需要特别注意的是，这一步仍然完全停留在理论层面。谷歌团队并没有构建任何实际的量子电路，也没有在真实设备上运行量子算法。他们得到的，只是一份满足特定资源条件的“电路描述”，也就是一套关于如何进行计算的方案。

白皮书实际上给出了两种不同的设计思路，在资源使用上各有侧重，但这一点并不是理解本文的关键。

既然已经有了电路设计，接下来就自然产生一个问题：这些设计是否真的能够按预期工作？这就引出了第二步——对电路进行测试。

二、电路测试

有了电路设计之后，接下来的问题就变得非常自然：这些设计是否真的能够按预期工作？

在传统的数学或计算理论中，人们往往会尝试对计算过程给出严格的验证，说明某个算法在实施的各种情况下得到的结果都是正确的。但是谷歌团队采取的是一种不同的思路。他们并没有试图对整个量子算法的实现过程进行严格验证，而是将问题大幅简化，只对其中的一个基础操作进行测试。

这个基础操作，就是椭圆曲线中的“点加法”。可以简单理解为，在既定规则下对两个“点”进行运算，得到另一个点。整个量子算法如果要运行起来，会反复调用这一类基本操作，因此谷歌团队选择从这里入手。

为了使测试成为可能，谷歌团队并没有直接处理完整的量子电路，而是只选取了这一基础操作，并进一步对电路结构加以限制，使其可以在普通计算机上进行模拟。在这样的条件下，这个电路的行为可以被看作一个普通的计算函数，从而可以在经典计算机上运行并进行测试。

从方法上看，这种处理方式与数字电路设计中的软件仿真有一定相似之处：先用程序描述电路，再通过计算机进行测试，而不涉及实际硬件实现。

在具体做法上，谷歌团队并没有对所有可能输入逐一验证，而是采用了一种更接近工程实践的方法：选取一组按照预先约定的方法生成的输入，这些输入在统计意义上具有随机性，并对电路进行多次测试。如果这些测试全部通过，就认为这个电路在绝大多数情况下是正确的。

这种方法类似于软件开发中的“随机测试”。它并不能保证在所有情况下都完全正确，但如果测试数量足够多，那么出错的可能性就会变得非常小。

到这里为止，谷歌团队已经完成了一件重要的事情：他们在不依赖量子硬件的情况下，对电路的一个关键部分进行了可操作的测试，并给出了“高概率正确”的结论。

但新的问题也随之出现：在不对外公开电路细节的前提下，外界如何相信这些测试确实被执行过，而且结果没有被人为修改？这就引出了第三步——如何证明这些测试过程本身是可信的。

三、测试过程的零知识证明

到这里为止，谷歌团队已经完成了电路设计，并对电路的关键部分进行了测试，得出了“高概率正确”的结论。但新的问题也随之出现：在不对外公开电路细节的前提下，外界如何相信这些测试确实被执行过，而且结果没有被人为修改？

该论文附录中的第一小节给出了答案，详见下图。

图中展示的是针对第一种电路测试所生成的“零知识证明”的全部内容。在这份“证明”中，白皮书并没有公开电路本身，而是给出了一组数据。这些数据实际上是在声明一件事情：存在这样一个电路，它满足白皮书中给出的资源限制，并且在所有测试输入上都得到了正确结果。

这组数据大致可以分为四部分。首先是对电路的功能和资源要求的描述，包括其计算任务、所需资源以及测试范围等；其次是一段“哈希值”（hash），可以理解为电路的“指纹”，用于唯一标识这一电路，防止作者在不同场合使用不同的电路；接下来是一段很长的“证明数据”（proof），它是对整个测试过程的一种压缩表示；最后是一段“验证密钥”（verification key），用于检查这份证明是否成立。

1. Low-Qubit Circuit Variant

Zero Knowledge Proof Statement 1 (Low-Qubit Variant) *We possess a quantum kickmix circuit* $C_{low\text{-}qubit}$ *(uniquely committed to via its cryptographic hash) with resource counts of at most:*

- $2,700,000$ *non-Clifford gates (CCX + CCZ)*
- $1,175$ *logical qubits*
- $17,000,000$ *total operations*

that correctly computes point addition on the elliptic curve `secp256k1` *across all 9,024 pseudo-random inputs deterministically derived from the circuit's own hash.*

Circuit SHA-256 Hash:
0xcc8f532ffea1583ceed3c9af75de3263ebaddd5fdf3cddfb3dea848b94d0396a

Groth16 Proof Bytes:
0x0e78f4db0000000000000000000000000000000000000000000000000000000000000000000000000000000008cd56e10c2fe24795cff1e1d1f40d3a
324528d315674da45d26afb376e867000000000000000000000000000000000000000000000000000000000000000000000000000000000215c7fe4fc59
7b861d82370ab556684ae36e98cf073e7f754f2788ad58721dbd012927516f316e7b4f3effb1dbd567732611cb0334f2d75e529c5e3
becd0629c17605c7ff87c6f23324328744454bdec0df425a4a63e3358c10079c85ef757412ae86ae1f85bf47ef6980852d6f65423be
2d90adb5b29896493324128b1cda0a0042f7138c850a1ca441210ba770a2eee39d56f6f90bf68b7a346e1658c6529715334621b6e1a
63b85875b8c8a610e0d885662879755803027dad57d97140afb2498bbb63215b236575f95b0019f2b9713bc810e1e044d47ab360e92
b899c46512fc97460609186bf1fe01c892a8015fb00e7fdea11b08f88c6adb79b1243518

Verification Key:
0x00ca4af6cb15dbd83ec3eaab3a0664023828d90a98e650d2d340712f5f3eb0d4

需要特别指出的是，这里的“声明”并不是单独存在的一段文字，而是已经被写入到测试程序本身。程序在运行过程中，会自动检查这些条件是否成立，例如资源是否超过上限、测试是否全部通过，以及电路是否与所声明的对象一致。如果其中任何一项不满足，验证就无法通过。因此，外界所验证的，并不是作者的描述，而是程序对这些条件的实际检查结果。

这些数据本身并不用于阅读，而是用于验证。任何人都可以利用公开的验证程序，将证明数据和验证密钥作为输入进行检查。如果验证通过，就可以确认：确实存在这样一个电路，它满足资源约束，并且已经通过了所有测试。

这种验证方式的关键在于，它不需要公开电路本身。外界无法看到电路的具体结构，也无法复现测试过程的细节，但仍然可以确认测试确实被执行过，而且结果没有被篡改。

从这个角度看，这一步所完成的，并不是对电路本身的证明，而是对“测试过程”的证明。它所保证的，是“这些测试确实发生过”，以及测试结果在统计意义上的可靠性，而并不直接涉及电路整体行为的全面验证。

结语

综合来看，该白皮书可以用一句话来概括：它并没有实现量子计算，也没有直接动摇现有密码体系，而是提出了一种新的方式来描述、测试并验证量子电路的资源开销。

从设计到测试，再到对测试过程本身的证明，这一整套方法，使原本停留在“理论估算”的结果，转变为一种可以被独立检验的技术声明。这在某种意义上，对传统科学研究中“可重复性”的原则形成了一种补充：当计算过程无法公开或难以复现时，可以通过密码学方法，使“计算确实发生过”这一事实本身变得可验证。

在信息安全领域，这种方法也具有现实意义。对于潜在的攻击技术，可以在不公开具体细节的前提下，提供一种可验证的“存在性证明”，从而在不泄露敏感信息的情况下发出预警，为防御方争取时间。

但与此同时，也需要清醒地看到，这种验证所保证的，是“测试确实发生过”，以及测试结果在统计意义上的可靠性，而并不直接等同于对电路整体行为的全面确认，更不意味着量子计算机在现实中已经具备了相应能力。

因此，与其说白皮书展示了量子计算的现实突破，不如说它展示了一种新的“证明方式”。至于这种方式在未来会产生多大的影响，还有待时间的检验。