Nillion完成2500万美元融资，主打的盲计算是什么？与ZKP、FHE有什么区别？

不少朋友看到Nillion融资$25M的消息，都会好奇 WTF什么是“盲计算”？ MPC、ZKP、FHE、TEE这些生僻概念刚有一些了解，一个崭新的概念又冒出来了。那么，盲计算的工作流程大致怎样？Nillion提供的盲计算解决方案究竟怎样？接下来，谈谈我的理解：

1）什么是Blind Compute（盲计算）？简单而言，盲计算是一种让服务端（节点）对某个加密状态的数据片段执行计算任务，最终达到保护隐私的安全计算方法。

和ZKP、TEE、MPC、FHE等增强加密算法目标都一致，差异在于：ZKP零知识证明生成证明需要巨大的开销，适合链下存储+计算，链上只验证的场景，比如：Rollup Layer2；TEE可信执行环境是一种依赖硬件厂商在隔离环境下进行计算的方法；FHE全同态加密虽然可以直接在加密数据上执行计算，但当下只支持特定运算；

“盲计算”是一种更加General的计算框架，因为ZKP、TEE、FHE等加密技术都可能被作为其技术框架的一部分。

众所周知，ZKP、TEE、FHE等目前都在和Crypto接轨技术落地应用探索和优化阶段。而盲计算则有可能把这些加密核心技术都聚合应用起来，从而为隐私保护探索出一体化的工程实践方案。

2）盲计算的核心逻辑是做分布式节点增强，让单个节点同时具备分段存储+计算的能力，再加上一个可验证的开放治理网络，进而达成节点不知道“完整”数据前提下有效工作的结果。如何理解呢？

常态下保护数据隐私状态需要在A节点存储数据，然后加密后交由B节点计算，再解密后经由C节点验证最终完成数据的存储+计算工作。这个过程中数据传输存在极大的成本损耗，且多次反复Encrypt——>Decrypt的过程数据存在暴露的情况，节点之间的互信成本也高，很难保证隐私不泄漏。

Nillion构建的业务逻辑恰好弥补了这一缺陷，其大致工作流程为（仅供理解）：

Nillion构建了一个分布式节点网络，每个节点都具备存储+计算的增强能力，Nillion网络在收到数据传输处理需求时，先经由Nada特定语言执行编译预处理，让原始数据被拆分成很多片段，并且都处于加密状态。

再经AIVM虚拟机来调度和分配，其分布式节点会随机存储并计算这些数据片段，最终完成聚合和统一验证。整个过程，单一节点并无法知道全部的数据内容，拼凑到一起却能完成整体数据的加密传输和计算。

为啥说盲计算可以聚合应用ZKP、TEE、FHE这些技术，逻辑也很简单，在数据预处理也就是给数据加密阶段完全可以应用FHE同态加密技术，而节点存储计算数据则可以在TEE可信执行环境下进行，在聚合和验证节点工作成果的时候则可以用ZKP提升验证聚合效率。

3）在我看来，ZKP、TEE、FHE、MPC等技术都或多或少存在一些工程化落地缺陷，目前Crypto领域几乎各个赛道都挤满了项目，但大差不差都在做成本和效率优化的工作，且都聚焦于Crypto特定应用场景。

Nillion所提出的盲计算框架，虽然也未实现大规模应用，但其一体化的加密解决方案，很可能在AI可验证计算、机器学习等更广泛的数据保护领域得以通用化采纳。