VERIFACT · 数字溯源样例报告

文档里的每一个数字都能指回一个产物吗?

对两个真实代码库的只读扫描：liboqs（第三方 PQC 参考实现）与 FIBEMATE（自有全栈仓库）。

工具	@fibemate/verifact 0.2.4（Node.js，本地运行，无网络出站）
扫描日期	2026-10-30
版本说明	本报告替代 2026-10-26 的 0.1.0 版；旧文件已从站点下线，避免两份数字并存。
扫描方式	只读。不修改被扫描仓库，不写入任何属性，不上传任何内容。
判定内容	文档写的值与产物里的值是否一致，以及两者各自的位置。
不判定内容	哪一方“正确”；是否存在漏洞；安全性、合规性或性能优劣。

一、为什么需要这件事

工程文档里的数字——密钥长度、测试通过数、基准耗时、覆盖率——通常以三种状态存在：**能指回产物**、**指不回产物**、**根本不该被当作产物属性。**审阅者看到的是一段文字，无法区分这三者。

结果是两类错误同时发生：真数字因为找不到证据而被当作空话；假数字因为没人核对而长期存活。**问题不在于谁在撒谎，而在于没有账本。**

二、方法

1. **抽取**：从 Markdown 中抽出带数字的断言，给每条一个稳定标识（内容寻址，不是行号）。
2. **建索引**：扫描仓库产物（JSON/CSV/日志/报告/时间戳文件），建立可比较的形状索引。
3. **绑定**：按文件、键名、数值形状为每条断言寻找候选产物，记录相似度。
4. **判定**：一致为 `verified` ，不一致为 `drift` ，找不到为 `unbound` ，多个候选为 `ambiguous` 。

关键设计：**“没有证据”和“不需要证据”必须分开。**作者可以声明某段是叙事、某次测量、或外部快照，但每一种豁免都必须署名并写明理由，没有名字的例外会让报告变成装饰。

三、结果 A：liboqs（第三方，只读）

liboqs 是 Open Quantum Safe 项目的 C 语言 PQC 库，是本领域的公共参考实现之一。扫描范围：仓库内全部 Markdown 文档 44 个、产物 2144 条。

24	16	66.7%	0
带数字的断言	可绑定到产物	绑定率	与产物不一致

该库的算法文档把参数写成表格，而对应的 `*.yml` 里存在同名的结构化定义，因此 24 条断言里有 16 条可以逐条对上，例如：

```
docs/algorithms/sig_stfl/xmss.md:43   xmssmt-sha2 = 20/2   →   docs/algorithms/sig_stfl/xmss.yml
```

另外 8 条找不到可绑定产物，典型是治理文档里的表决比例（2/3）与构建说明里的开关默认值——它们不是代码产物的属性。这正是“不应被计入覆盖率”的一类，而不是“缺失证据”。

四、结果 B：FIBEMATE（自有仓库）

范围：按 `verifact.json` 全仓配置扫描 379 个文档、4742 条产物（含时间戳存证、形式化验证产物、基准输出）。快照时间：2026-10-30 12:31；文档根覆盖 `fibemate`、`fibemate-tools`、`fibemate-core`，排除 `docs/pr-archive`、`CHANGELOG.md`、`scripts/` 与 `tools/`。

指标	数值	说明
抽取到的断言总数	2347	含被分类、被忽略、被拒绑的条目
进入判定的断言	2137	真正应当有产物的部分
已验证	137	文档值与产物值逐条一致
未绑定	1768	找不到产物——诚实记录，不隐藏
候选冲突	214	多个候选，需人工判定
与产物不一致	0	本次为 0，门禁通过
引用已定位（lg-XXX 等实体）	18	声明引用的存证实体在扫描范围内找到
已分类（非产物属性）	72	叙事/进度/一次性测量，逐条署名
哈希存证（不介入判定）	84	TSA 盖戳文件，改一个字节即破坏存证

标准编号（非度量）	25	FIPS 203/205 之类，避免被当成比值
显式豁免	21	每条必须写明为什么不该核

核验率按两个分母同时给出：6.4%（137/2137，只算应当有产物的）与 6.2%（137/2217，含分类与拒绑）。只报一个数就是选择性地好看。

口径说明：`claimsTotal` 含被分类、被忽略与被拒绑的条目，所以它与 `codeArtifactClaims` 不等；两个百分比的分母不同，不可互相引用。本节即全仓口径，与 2026-10-26 那份 `docs/` 子集快照（314 条断言）不是同一分母，不构成趋势比较。

五、门禁有没有牙：一次真实的红

下面不是构造的演示，而是移除配置里全部 19 条豁免后，工具在同一仓库上给出的真实输出（退出码 1）：

> **门禁 FAILED**：6 条声明与产物不一致。四条是把 Node.js 基准的数字绑到了硬件基线（262µs），另外两条是绑定对象本身找错了（em-dash 计数、BKZ 成本）。

#	位置	键	文档值	绑定到的产物	相似度
1	`fibemate/README.md:269`	keygen-s	60ms	pqc-hw-bench/README.md 262µs	0.57
2	`fibemate/README.md:469`	cost	2	pqc-hw-bench/docs/lattice-estimator-run-2026-09-28.md 30bits	0.75
3	`fibemate/README.md:515`	kem-stress	8.9s	pqc-hw-bench/README.md 262µs	0.53
4	`fibemate/docs/audit-log/gbk-fix-verify-cleanup_2026-08-14.md:12`	double-ratchet-pq-js	13,	tools/cars-scorecard.json 100	0.59
5	`fibemate/www/docs/performance-benchmarks-2026-07-18.md:35`	rsa-keygen	60ms	pqc-hw-bench/README.md 262µs	0.57
6	`fibemate/www/docs/performance-benchmarks-2026-07-18.md:153`	kem-完整	9.4ms	pqc-hw-bench/README.md 262µs	0.60

这 6 条此前被豁免静默排除。**移除豁免并不会制造问题，它只是让问题可见。**恢复豁免后门禁重新变绿，但每一条豁免都必须写明：为什么不该核、指向哪里、何时应当删除。

口径：共 19 条豁免中，6 条在移除后表现为 drift；其余 13 条仍属“口径不同”或“非产物属性”，保留豁免。上表不是“19 条全有问题”。

> **结论**：这个工具的价值不在于“报告很绿”，而在于它能证明自己会红，并且红得有据可查。

六、这份报告不主张什么

- 不判定数字背后的实现是否正确，也不构成安全评估或合规结论。
- “已验证”仅表示文档值与某个产物值一致；如果产物本身是伪造的，它同样一致。
- “未绑定”不代表有错，只代表账本上没有对应条目。
- 扫描只读、本地运行；本报告不包含任何被扫描仓库的源代码。

工具与规则均为开源实现，判定逻辑、豁免理由、门禁策略都在仓库内可复核。