FIBEMATE

TECHNICAL WHITEPAPER · POST-QUANTUM MIGRATION

可复核的后量子迁移

从资产盘点，到交付字节

版本	1.0
日期	2026-10-30
适用对象	系统集成商 · 技术决策者 · 迁移项目负责人
文档性质	对外技术白皮书（非认证报告、非安全评估）
复核入口	第 9 节四组命令，均可独立执行

本文所有可验证结论均附复核命令。本文不构成密评意见、FIPS 140 验证或第三方审计结论； 引用第三方数据时随文附其免责声明。

fibemate.net · 仓库 github.com/Lennonhaha/fibemate · GPL-3.0-only

> 阅读方式：本文每个数字都给出出处或复核命令。凡当前无法复核的，明确标注"未验证"，不写成结论。

复核状态（2026-10-30）：交付层数字（266 个归档文件、TSR 清单、`lg-116` 令牌、14 件摘要清单） 于 2026-10-30 在公网重跑，结果为本文件所载；Rust 测试（182 / 0 ignored）与 npm 签名两项来自 2026-10-29 的审计记录，今日未重跑，文中按原样标注日期；第三方数据（NIST IR 8547、PQ Crypta） 为引用，不属于本项目复核范围。

---

0. 一页结论

1. **我们交付的不是"又一个 PQC 库"，而是一条可以被人复算的证据链。** 算法实现、互操作记录、时间戳、线上交付字节，四层各自有独立的检查手段，任何一层都不靠"我们说过"来成立。
2. **迁移的难点不在算法，在"看不见"。** 一个字段被序列化器静默丢弃、一个数字被改动而校验仍然通过、一次部署返回 0 但线上字节已被改写——这三种情况都会让一份"合规材料"看起来完整，实际上载不动结论。
3. **我们的方法只有一句话：把"声明状态"和"评估状态"分开，并强制对账。** 汇总页上的每个数字，必须能由它旁边的计数推出来；推不出来就渲染成红色，而不是渲染成正常。
4. **边界写在最前面：** 我们不承诺"安全"，不承诺密评/认证结论，不替代第三方审计。我们承诺的是：**每个结论都能被买家自己复算。**

---

1. 背景：时间表已经落到日历上

NIST IR 8547 已定稿。 NIST 于 2026-10-27 发布《Transition to Post-Quantum Cryptography Standards》最终版：量子脆弱的公钥算法（RSA、ECDSA、EdDSA、有限域与椭圆曲线 Diffie-Hellman 等）在 **2030 年后进入弃用（deprecated）**，在 **2035 年后从 NIST 标准与指南中 禁用（disallowed）**；对联邦高价值资产与高影响系统，关键建立（key establishment）的期限是 **2030-12-31**，数字签名的期限是 **2031-12-31**（后者来自 IR 8547 所引用的行政令与 OMB 备忘录 时间表）。IR 8547 同时建议以 **ML-KEM-768** 作为默认参数集，并明确：混合方案中只要有一个输入 来自已批准方案，仍可在 FIPS 140 下验证。

为什么密钥建立排在签名前面。 签名可以等到量子计算机出现再伪造，因为它只在连接发生时被 验证；而被记录的密文可以今天抓走、以后解密（harvest now, decrypt later）。对保密期超过十年的 数据，迁移窗口实际上已经关闭了一半。

国内语境需要说清楚。 国内商用密码的管理、检测与密评体系独立于 NIST 体系，国密 PQC 标准 尚在推进中。本文引用的 NIST 时间表是**工程规划依据**，不是国内合规结论；本项目对 SM2/SM3/SM4 的实现用于技术验证与互操作测试，不等同于商用密码产品认证，也不构成密评意见。

---

2. 外部实测：PQC 在传输层的代价

以下数据来自第三方 **PQ Crypta** 的 QUIC 实测与公开分析，用于给出量级，不用于背书任何实现。

指标	观察值	说明
握手轮次	QUIC 约 1 RTT；TCP + TLS 1.3 至少 2 RTT	QUIC 把传输与密码握手合并，冷连接省一个 base RTT
QUIC 相对 TCP 的实测优势	约 5–35 ms	跨其测试的住宅与云路径，含协议开销差异
PQC（X25519MLKEM768）计算开销	约 1–4 ms	相对上述网络时延属小量
ML-KEM-768 公钥	1,184 B	对比 X25519 的 32 B
ML-KEM-1024 公钥	1,568 B	更高安全等级
X25519MLKEM768 混合公钥	1,216 B	混合组

免责声明（必须随数据一起引用）：上述数据由 PQ Crypta 的扫描器产生，其测量客户端栈为 Rust 的 `quinn` / `h3` 库，测量结果反映该栈在特定时间点、特定网络路径上实际协商到的行为； 它不是受控基准测试，不覆盖浏览器与移动端栈，也不构成对任何实现性能的承诺。本文未复跑该组 实验，引用时不作二次加工；方法与限制以原始页面为准。

这对采购意味着什么：在多数企业链路上，PQC 的**计算**代价可以忽略，代价主要落在**握手 字节数**与**丢包/时延放大**上。因此迁移方案的重点不是"算法快不快"，而是"握手包能不能过、 证书链能不能被对端处理、降级路径有没有关掉"。

---

3. 真正的问题不是算法，是"看不见"

3.1 字段静默丢失：文档合法，信息变少

在 `cyclonedx-python-lib` 11.12.0 中， `algorithmProperties.algorithmFamily` 与 `ellipticCurve` 在 JSON 与 XML 输出中都会被丢弃，产物**仍然通过 CycloneDX 1.7 schema 校验**。这不是崩溃， 也不是报错——是一份"格式正确、内容更少"的 CBOM。上游把它当作已知的 1.7 支持缺口；真正值得 注意的是它的**静默**。

同族的第二例来自我们自己的语料：在一份 345 份文档的审计语料上，密封产物的 glob 修复前 匹配不到真实标签（真实标签形如 `fibemate/docs/tsa/…`），45 条带时间戳的声明被判为 `unbound` ——一条**永远不会响的零命中规则**。修复后重新对账，计数随之变化。

结论：只报告"合法/非法"的工具，无法区分"检查过且干净"与"根本没检查"。读者默认理解成前者。

3.2 值检查看不见字节改动

同一份声明、同一件产物，四种篡改场景：

#	篡改方式	值级校验（verifact）	字节+令牌校验（verify-evidence）
S0	无	exit 0，3 条 verified	`verified`，exit 0
S1	改一个数字（62 → 61）	`drift`，exit 1	`drift`，exit 1
S2	改一个字节，不动任何数字	**exit 0，仍然绿**	`drift`，exit 1

#	篡改方式	值级校验（verifact）	字节+令牌校验（verify-evidence）
S3	S2 + 重写摘要清单	exit 0	`drift`，exit 1（imprint 不符）
S4	缺 CA 证书	—	`unverifiable`，exit 3

S2 是关键行：值级校验器被问的是"这份文档是否还说了它说过的话"，它答"是"。字节在语句下面 被改掉了，比较值的检查器不会报警——它报成功。S3 说明为什么摘要清单本身必须被令牌封住： 重写清单可以修复第一环，只有令牌对旧清单的 imprint 能抓住它。S4 单独存在的原因也一样： "判断所需的输入不在场"绝不能被渲染成"通过"。

3.3 部署成功 ≠ 交付成功

我们把同一类错误在我们自己的交付链上实测了一次。旧流程把 `git reset --hard` 返回 0 当作部署 成功——它只说明检出目录移动到了被审阅的提交，不说明 nginx 在发什么。替换后的门禁从公网侧 断言三件事：清单文件逐字节一致、时间戳令牌逐字节一致、 `docs/tsa/` 下每个被跟踪文件要么逐字节 一致、要么被 nginx 策略以 403 拒绝。**第一次运行就抓到线上缺陷**：

```
[FAIL] live:archive: 1 of 266 tracked files not delivered as expected
    mismatch docs/tsa/2026-10/verify_cyclonedx_data_loss.py
             (94b28f6d2de183e6 != accd674461443e26)
```

线上 5,025 B，提交 5,170 B。 `docs/tsa/**` 被标记 `-text`，正是为了让封存产物保留 CRLF 字节； 而发布到 `www/` 的镜像走了默认的 `* text=auto eol=lf`，145 个回车被规范化掉。**同样的文本， 不同的字节。** 封存摘要从此匹配不上任何可下载的文件——证据包只差一个换行符就不能被引用， 而当时所有既有检查（lint、编码、BOM、SPDX）都是绿的。

这张表是本文的骨架：

检查类型	能回答	不能回答
文本级（lint / 编码 / schema）	文本是否良构、是否符合格式	字节是否仍是签发时的字节
值级（数字对账）	文档是否还声称同一件事	文档本身有没有被改字节
字节级（摘要 + 时间戳）	这份字节是否就是被封存的那份	内容是否正确、结论是否成立
交付级（公网探测）	线上发的字节是否等于提交的字节	内容是否正确、第三方是否认可

4. 方法：把"声明"与"评估"分开

`verifact` 给每条声明两个互不折叠的状态：

- **声明状态**：产物是否存在、是否被哈希、是否带时间戳；
- **评估状态**：代码是否真的做了声明所说的事。

具体规则：

1. **内容寻址绑定**：声明绑定到文件摘要，文件一变，声明不会"跟着走"；
2. **严格不等**：只有每个必需字段都匹配才算 `verified`；缺字段是 `unbound`，永远不算通过；
3. **显式未知**：`ambiguous` 与 `drift` 必须出现在报告里，并且必须参与对账；
4. **对账规则**：页面上任何汇总数字，必须能由旁边的计数推出（例如 `137+0+213+1768=2118`、`2118+17+1=2136`、`2136+126=2262`）；推不出来就渲染成红色。

这条"对账规则"是整套方法里最便宜、也最少见的防御：它不要求工具更聪明，只要求它**不许在算术上撒谎**。

证据分级（采购时可直接按级验收）

级别	证据形态	能支持的说法	典型成本
L0	自测 / 单元测试	"代码按我们的理解运行"	低
L1	官方向量（ACVP KAT）	"在标准给定输入上符合标准"	中
L2	独立 oracle 互操作	"两个独立实现对这些字节达成一致"	中高
L3	字节级记录 + RFC 3161 时间戳 + 离线校验器	"这份记录在某个时刻存在，且未被改动"	高
L4	第三方审计 / 认证	"独立机构出具了意见"	最高

我们没有 L4。 本项目当前最强证据是 L3：记录可离线校验，但出具方是我们自己。

---

5. 能力与证据（可复核）

能力	证据	复核方式	不覆盖
ML-KEM / ML-DSA 标准符合性	Rust 侧 182 个测试通过、0 ignored；ACVP KAT 行逐条可见	`cd fibemate-tauri && cargo test --all-features`	不证明实现无侧信道
三端字节一致	3 个固定种子，每种子 12/12 项字节相等（pk/sk/ct/ss + 双向 round trip）	公开种子与矩阵规格，重跑比对	Rust 与 WASM 同源于 `libcrux`，不是两个独立实现
互操作矩阵	62/62 检查、12 行、6 个基线、0 失败；`ss =P` 与 `ct =P` 分列，不合并	矩阵 JSON + 12 份十六进制日志	仅 ML-KEM-768、仅软件端
记录封存	14 件产物摘要清单 + FreeTSA `lg-116`（serial `0x09823CA3`，2026-10-28 11:58:04 GMT）	离线 `verify-evidence`：逐文件 SHA-256 → 清单 → imprint → `openssl ts -verify` → nonce	时间戳证明存在，不证明正确
篡改可见性	S2：改一个字节，值级校验仍绿，字节级校验 exit 1	同一份记录上重放 S0–S4	不是形式化证明
供应链来源	npm：`npm audit signatures` 通过；PyPI：wheel 与 sdist 均具备 PEP 740 证明	`npm audit signatures`；`/integrity/` 端点逐文件查	证明构建链路，不证明输入可信

能力	证据	复核方式	不覆盖
交付字节	公网门禁：清单、令牌、266 个归档文件逐字节比对（或 403）	`node scripts/check-live.cjs`	只证线上字节=提交字节

6. 与系统集成商的分工

国内 PQC 项目的现实是：**投标资格在集成商手里，底层工程能力往往不在。** 我们不与有资质的 集成商抢标，也不做资质挂靠；我们做的是他们投标文件背后最难补的那一段。

角色	承担
集成商	投标主体、资质与商务、客户关系、密评/测评流程对接
FIBEMATE	PQC 迁移方案、算法与协议层实现复核、CBOM 盘点、证据链与复现脚本、技术答辩支撑

我们不做的三件事：不代替密评机构出结论；不承诺测评通过率；不以任何形式出借资质或代签。

对集成商的实际价值：投标文件里的"我们支持 ML-KEM"可以升级为"这是三端字节一致的记录、 这是可离线校验的时间戳、这是公网交付字节的校验结果，评委可以自己跑"。在技术评分环节， 这比多写十页方案有效。

7. 交付物

一次典型交付包含五件东西，全部可移交、可复算：

1. **仓库快照**：固定提交号，含全部实现与测试；
2. **审计报告**：结论 + 每个结论的复核命令 + 明确的不覆盖项；
3. **证据包**：摘要清单（`SHA256SUMS`） + RFC 3161 时间戳（`.tsq` / `.tsr`） + CA 证书；
4. **复现脚本**：离线校验器、互操作矩阵、交付门禁；
5. **已知缺口清单**：当前不能证明什么，以及补上它需要什么。

合作形态可以是：技术分包（按项目）、年度技术顾问、或源码授权 + 迁移支持。具体商务与责任 条款以合同为准。

8. 边界：先说不承诺什么

1. **不承诺安全。** 字节一致与官方向量符合，只说明"符合标准、两端理解一致"，不说明标准不可破、不说明实现抗侧信道。
2. **不承诺合规。** 本文不构成密评意见、不等同 FIPS 140 验证、不替代任何第三方认证。
3. **不承诺协议安全。** 互操作矩阵验证的是序列化与 KEM 行为，不是 TLS 握手、ratchet 或密钥存储的安全性。
4. **不承诺生产环境性能。** 我们没有真实 TLS 网关的并发压测数据；库级微基准不能外推成端到端指标。
5. **不承诺第三方认可。** 记录可离线校验，但出具方是我们；L4 证据需要独立机构。

6. **不承诺覆盖全部算法。** 当前矩阵为 ML-KEM-768 单参数集、软件端；签名方案尚未进入矩阵。
7. **时间戳只证存在。** 一个错误答案上的时间戳，仍然只是一个错误答案上的时间戳。
8. **我们是所评估工具的作者。** 变异测试与外部 schema 验证是为了让这种偏向代价更高，不是消除它。

---

9. 复核入口

```
# 标准符合性（Rust / ACVP KAT）
cd fibemate-tauri && cargo test --all-features

# 公网交付字节（清单 + 令牌 + 归档逐字节）
cd fibemate && node scripts/check-live.cjs

# 声明 vs 字节（S0–S4）
cd fibemate-tools && node scripts/claim-evidence/run-e2e.cjs --fibemate ../fibemate

# npm 供应链签名
cd fibemate-tools && npm audit signatures
```

审计口径与全部原始结论见仓库根目录 `AUDIT.md`；证据链的方法叙述见 `docs/tsa/2026-10/paper-story.md`；漂移事件（含为什么没有重签）见 `docs/tsa/DRIFT-20261029.md`。

本文为版本 1.0（2026-10-30）。新增或修改任何数字前，必须重跑第 9 节对应的复核命令并更新 版本号；引用第三方数据时必须同时引用其免责声明。