电车罗格端到端加密白皮书

ChargeLog End-to-End Encryption White Paper

Crypto v1　2026 年 9 月

摘要：电车罗格采用分层密钥架构，在用户设备上完成业务记录加密。每次加密生成独立记录密钥，以 AES-256-GCM 保护正文，再由用户主密钥封装记录密钥。主密钥通过恢复词、设备信封及可选密码信封获得保护。云端保存密文、封装密钥和必要同步元数据。本文给出数据范围、算法参数、密钥生命周期、协议格式及安全边界。

关键词：端到端加密；认证加密；密钥封装；HPKE；数据恢复

1　保护范围与系统角色

客户端负责密钥生成、正文加密、信封封装、认证解密和本地数据恢复。LoggerCloud 提供账号认证、密文存储、设备授权及同步协调。端到端加密模式下，业务正文在离开设备前已被加密。[7-9]

表 1 端到端加密的业务记录范围

记录类型	加密正文包含的内容
trip_v1	出发地、目的地、日期、里程、规划信息、关联车辆及内嵌充电记录。
charge_v1	金额、币种、电量、时长、续航、SOC、充电模式及关联车辆。
vehicle_private_v1	昵称、默认车辆标记、购车价格、当前里程、电池容量及创建、更新时间。

车辆 UUID、品牌、车型和年份经独立接口同步，服务器可读。服务还可见账号关联、记录 ID、密文长度、版本、云代际、变更序列、服务端时间、删除标记和设备登记信息；行程数量与充电数量另以聚合统计上报。

普通云备份发送服务器可读的 JSON 载荷，端到端加密备份发送密文；HTTPS 用于传输保护。关闭备份只停止相应业务同步，不会自动删除既有云数据。当前年度报告类型未形成完整上传与恢复流程，不计入表 1 的保护范围。

2　密钥层级与存储

用户主密钥 UMK 和记录密钥 RK 均由客户端独立随机生成。UMK 负责封装 RK，RK 负责加密单条记录。恢复词、设备和可选密码分别提供解开同一 UMK 的路径。[7]

表 2 密钥层级与持久化形式

材料	生成及用途	存储形式
UMK	256 位随机对称密钥；封装记录密钥。	云端只保存信封；解锁后可驻留应用内存。
RK	每次加密或重加密生成新的 256 位随机密钥。	云端保存由 UMK 封装的副本。
KR	恢复 seed 经 HKDF-SHA256 派生的 32 字节密钥。	计算使用；云端保存盐和恢复信封。
设备密钥对	P-256 密钥对；HPKE 解封装设备信封。	优先 Secure Enclave；Keychain 保存本机设备材料。
KP（可选）	密码经 Argon2id 派生的 32 字节封装密钥。	云端保存 KDF 参数和密码信封，不保存 KP 明文。

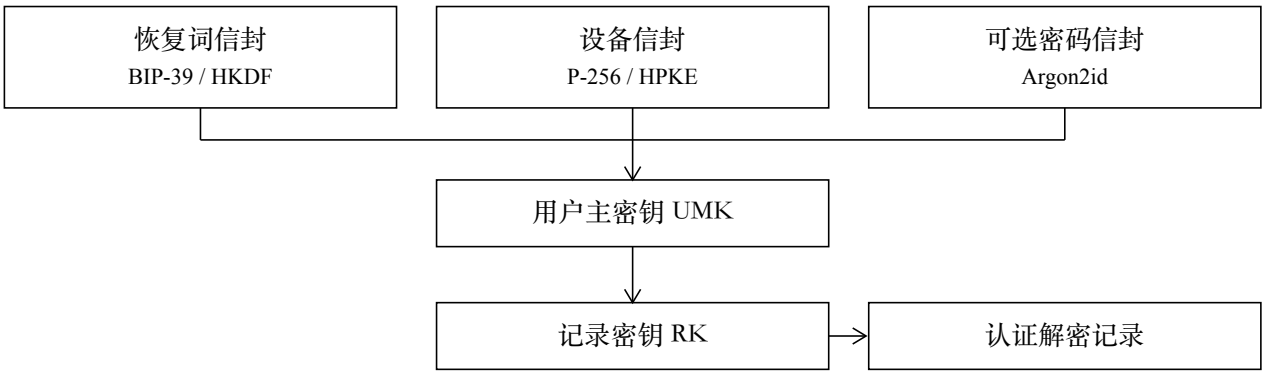

图 1 主密钥恢复与记录解密的分层关系

加密时依次产生正文密文、记录密钥信封和主密钥信封；恢复时先取得 UMK，再解开 RK，最后验证并解密正文。恢复词不直接编码 UMK，而是提供另一组随机恢复材料。

12 个恢复词的初始熵为 128 位；使用 256 位 AES 密钥并不意味着恢复词路径也具有 256 位安全强度。恢复词副本只保存在设备 Keychain 中，不上传到 LoggerCloud。

3　记录认证加密

3.1　规范化正文

记录正文 M 由 schema_version、record_type、record_id、modified_at_ms 和 payload 组成。JSON 使用键排序及不转义斜杠的编码选项；金额、电量、距离与时间按模型转换为整数单位，空值遵循字段编码约定。新写入 UUID 统一小写。这是应用的稳定编码约定，不等同于完整 RFC 8785 实现。[8]

3.2　正文加密与记录密钥封装

令 A 为附加认证数据，N_1 与 N_2 为两次加密独立生成的 nonce，计算关系如下：

$$C = \text{AES-GCM}(RK, N_1, M, A) \tag{1}$$

$$W = \text{AES-GCM}(UMK, N_2, RK, A) \tag{2}$$

C 和 W 各含密文及认证标签。两次 AES.GCM.seal 均使用 256 位密钥，并由 CryptoKit 自动生成 nonce；默认 nonce 为 12 字节，认证标签为 16 字节。[1]

3.3　附加认证数据与验证顺序

正文加密和记录密钥封装使用相同的 UTF-8 附加认证数据：

$$A = \text{UTF8}(\text{"chargelog-record|v1|\{record\_id\}|}\\ \text{\{schema\_version\}|\{key\_version\}"}) \tag{3}$$

大括号表示实际字段插值；公式换行仅用于排版。解密先验证 W 并取出 RK，再验证 C 并恢复正文，任一步认证失败均拒绝记录。读取保留大写 UUID 的历史 AAD 兼容路径，以支持早期数据。

表 3 内容完整性与同步约束的分工

机制	作用与范围
记录 ID、schema 与 key 版本	绑定到两层 AAD，防止直接跨记录替换不匹配的密文组件。
本地 SHA-256 内容摘要	用于变化检测与冲突处理；不上传到记录接口，不是内容签名。
账号、云代际与修订号	未包含在当前 AAD 中，归属和新鲜度需由会话与同步协议约束。

随机 nonce 降低重复概率，但不是确定性的永不重复保证。每次加密生成新 RK；封装多个 RK 时仍共同使用 UMK，因此主密钥层的 nonce 随机性同样重要。

4 恢复词与密码信封

4.1 恢复词派生

客户端使用 WalletCore 生成初始熵为 128 位的 12 个 BIP-39 英文恢复词，并以空 passphrase 取得 seed。BIP-39 定义了助记词校验及 PBKDF2-HMAC-SHA512 的 seed 生成过程。[2] 恢复封装密钥由 HKDF-SHA256 派生：[3]

$$KR = HKDF\text{-}SHA256(seed, salt, info, 32) \tag{4}$$

salt 为随机 16 字节，info 固定为 com.creatisan.chargelog/recovery-wrap/v1。KR 使用 AES-256-GCM 封装 UMK，恢复信封的 AAD 为：

chargelog-master-key|v1|recovery|{key_version}

恢复输入会清理首尾空白并转小写，再检查助记词有效性。仅通过 BIP-39 校验不能证明恢复词属于当前账号；还必须成功认证解开当前恢复信封。完整恢复需要有效恢复材料、相应信封和记录密文。

4.2 密码信封参数

表 4 可选密码封装参数

参数	实现值
KDF	Argon2id v19
内存 / 迭代 / 并行度	65,536 KiB（64 MiB）/ 3 / 2
盐 / 输出长度	独立随机 16 字节 / 32 字节
信封加密	AES-256-GCM
AAD	chargelog-master-key|v1|password|{key_version}

派生密钥 KP 用于封装同一 UMK。密码信封保存盐和 KDF 参数；取得该信封的人可以离线猜测密码，并通过认证解密结果验证猜测。Argon2id 增加单次尝试成本，密码强度仍决定该路径的实际保护能力。[6]

4.3 当前产品入口

当前启用端到端加密和恢复数据的界面传入 password: nil，以恢复词和设备信封完成流程；恢复时可以保留已有密码信封。底层密码解封装能力不等于当前界面提供“仅凭账号密码恢复”。账号登录凭据与数据解密材料分别管理。[9]

5 设备信封与终端保护

5.1 HPKE 封装

设备生成 P-256 密钥对，采用 CryptoKit 的 P256_SHA256_AES_GCM_256 套件封装 UMK。该套件组合 P-256 密钥封装、HKDF-SHA256 与 AES-256-GCM。[4] 设备上下文同时用于 HPKE info 和附加认证数据：

chargelog-device-key|v1|{device_id}|{key_version}

信封 kind 为 device:{device_id}。解封装必须匹配本机设备 ID；信封包含 encapsulated_key、接收方公钥和包含认证标签的 ciphertext。nonce 和 tag 不作为设备信封的独立字段输出，不能照搬记录加密的拆分格式。

当前 HPKE 调用未提供发送方认证私钥，属于 Base 模式。它保护发给接收设备的秘密，不独立认证发送方身份；设备授权由账号会话和云端密钥配置共同管理。

5.2 Secure Enclave 与 Keychain

当 SecureEnclave.isAvailable 为真时，客户端创建硬件 P-256 私钥，将用于重新打开硬件密钥的 dataRepresentation 保存在 Keychain；该表示不等同于可导出的私钥明文。硬件不可用时生成软件私钥，并保存 rawRepresentation。回退条件是硬件可用性，没有仅限模拟器的编译限制。[7]

表 5 本机密钥材料的访问边界

本机材料	保护方式
设备密钥材料、本机设备信封	按 accountID 分隔；使用 WhenUnlockedThisDeviceOnly 访问等级。
恢复词副本	同样按账号分隔并限定本机解锁访问；另使用 userPresence 访问控制，部分界面入口增加生物识别验证。
已解锁的 UMK	以 SymmetricKey 缓存在应用内存；lock() 和账号切换清除缓存引用。

ThisDeviceOnly 材料不能作为在另一台设备上直接恢复该 Keychain 项的通用备份。[5] 新设备仍需通过有效恢复路径取得主密钥；本机尚未保存恢复词时，身份验证不能生成其副本。

硬件保护设备私钥不代表所有业务解密均在 Secure Enclave 内完成。UMK 和正文在解密过程中可进入应用内存；清除引用也不等同于已验证的物理内存清零。业务数据库和导出文件不自动继承这套云端加密保护。

6 恢复与密钥生命周期

6.1 启用与新设备恢复

首次启用时，客户端生成 UMK、恢复词及设备信封，用户完成恢复词确认后执行备份迁移。目标代际提交成功后，客户端保存对应恢复词副本并更新同步状态。

新设备恢复依次执行：登录目标账号并获取 profile；使用恢复词解封装 UMK；创建本机设备信封并更新云端配置；登记设备、保存已验证的恢复词；逐条解开 RK，验证并恢复业务记录。后续日常使用可以依赖已授权的本机设备信封。[9]

6.2 恢复词更换与设备撤销

更换恢复词时生成新的候选恢复材料，保持原 UMK 和 key_version。用户确认后先更新云端恢复信封，再替换本机 Keychain 副本；生成或显示候选词不等于已经生效。

旧恢复词若与旧恢复信封一同泄露，仍可解开未改变的 UMK。处理主密钥泄露需要轮换 UMK，并重新封装记录密钥或重加密数据；仅替换当前恢复信封无法撤销历史副本。当前流程不等同于完整自动主密钥轮换。

撤销设备会移除其云端授权，正常客户端的后续同步检查将拒绝缺失的设备信封。已被保存的明文、主密钥及旧信封不能通过云端撤销远程收回。

6.3 模式切换与删除

表 6 备份迁移与云端删除

操作	数据与状态变化
普通备份 → 端到端加密	拉取原有记录；在客户端加密并上传目标 generation 的记录与 profile；提交迁移后切换。
端到端加密 → 普通备份	解开现有记录，在目标 generation 上传服务器可读业务载荷；提交后切换并清理本地云密钥。
删除云端数据	提交敏感操作验证；成功响应后关闭备份，更新代际，清理本地云密钥及同步状态，保留本机业务记录。

云端提交和本机 Keychain、数据库写入是不同步骤。提交成功后本机仍可能写入失败，不能将整体描述为跨端原子事务；云端已提交的数据状态应予保留。模式切换也不能消除服务器历史备份中可能存在的旧明文。

7　同步协议与数据格式

7.1　当前授权与同步一致性

每次端到端加密同步前，客户端重新获取 profile，要求其 cloud_generation 与同步代际一致，并只接受 key_version 匹配的设备信封。设备授权缺失或解封装失败时，锁定缓存并进入恢复状态；代际不匹配时停止本轮同步。[9]

VaultSyncContext 绑定账号、数据空间、云代际和密钥版本。记录写入携带 If-Match 与 Idempotency-Key，分页检查点绑定备份模式与代际。以上机制协调正常客户端的并发和重试，依赖服务端返回当前状态。

7.2　传输字段

表 7 Crypto v1 的主要传输约定

对象	字段与约定
加密记录 PUT	ciphertext、nonce、tag；wrapped_key_ciphertext、wrapped_key_nonce、wrapped_key_tag；crypto_version、schema_version、key_version、cloud_generation。
记录定位与归属	record_id 位于 URL；请求体不含 account_id、record_id、content_hash、base_revision。归属由认证会话确定；本地 baseRevision 转为 If-Match。
密钥 profile	profile_version、key_version、recovery_wrapped_key、可选 password_wrapped_key，以及按 device_id 映射的 device_wrapped_keys。普通 profile PUT 不发送 cloud_generation。
主密钥信封编码	先将信封 JSON 编码为 UTF-8，再做 Base64；内部二进制字段分别 Base64 编码。设备信封的 nonce/tag 为 nil。
版本与删除	crypto_version = 1，schema_version = 1，新 profile 默认 key_version = 1。记录删除请求携带 generation 与 If-Match，响应为删除标记。

7.3　传输认证与聚合统计

客户端只接受 HTTPS API 配置。密码登录使用 challenge 与 SHA-256 proof；注册和重置使用 Argon2id PHC 材料。这属于账号认证链，不直接恢复 UMK。记录同步和本地持久化完成后上报 trip_count、charge_count，行程内嵌充电记录不重复计数。

跨语言实现须逐字节对齐 UTF-8 AAD、UUID、Base64、整数单位、null 策略与 JSON 编码。记录认证结果、服务端修订号和本地内容摘要各有不同用途，不能互相替代。[8-9]

8 威胁模型与安全边界

以下分析以客户端可信、随机数与密码学实现正确、有效解密材料未泄露为前提。保护目标是云端受保护业务内容的机密性及认证完整性，不包括隐藏全部服务元数据。

表 8 主要威胁与保护边界

威胁	保护效果与限制
云端密文泄露	仅持有密文和封装密钥不足以直接恢复正文；账号关联、记录长度及同步元数据仍可见。密码信封存在时须考虑离线猜测。
网络监听与修改	HTTPS 保护传输；记录 AES-GCM 检测正文及绑定字段被修改。终端和服务端仍是传输链的端点。
记录组件替换	两层 AAD 绑定记录 ID 和版本；直接混用不匹配的正文、记录密钥信封或绑定字段应导致认证失败。
恶意服务器回滚或删除	旧有效密文仍可能通过认证；代际、修订号、变更序列及删除标记未由记录 AAD 认证，不提供独立的全局防回滚和完整历史证明。
终端失陷	恶意进程、系统漏洞或受损客户端可接触解锁后的密钥和明文。硬件设备私钥保护不能代替完整终端安全。
密钥材料泄露	UMK 泄露影响其封装的记录；恢复词与相应信封泄露可恢复 UMK。撤销设备或更换恢复词不能收回历史材料。

8.1 恢复能力与历史数据

若所有可用设备解密材料、有效恢复词和任何可用的密码兼容恢复路径均不可用，账号找回本身不能恢复原有密文。仅有恢复词但缺少对应信封或密文，也不能重建已删除的数据。

8.2 保证范围

当前架构不提供完整前向保密、后量子安全、对受损客户端的保护或云端可用性保证。普通备份、业务数据库、导出文件以及服务器备份保留策略具有独立的安全边界。

本文以 2026-09-11 客户端工作区为实现基线，不构成独立安全审计或生产环境认证。项目包含算法向量、错误密码、AAD 篡改、账号隔离及授权状态等测试定义；本次文档修订未执行这些测试。

附录 A　信封算法标识

表 A.1 主密钥信封的算法标识

信封	algorithm 值
恢复词	bip39-hkdf-sha256-aes-256-gcm
密码	argon2id-aes-256-gcm
设备	hpke-p256-hkdf-sha256-aes-256-gcm

恢复词信封的 KDF 元数据为 hkdf-sha256，版本 1，输出 32 字节；salt 与信封一同保存。设备信封使用 kind = device:{device_id}，恢复词与密码信封分别使用 recovery 和 password。

参考文献

[1] Apple. CryptoKit: AES.GCM; SealedBox.tag; SealedBox.combined [EB/OL].

[2] Bitcoin BIPs. BIP-39: Mnemonic code for generating deterministic keys [S]. 2013.

[3] Krawczyk H, Eronen P. HMAC-based Extract-and-Expand Key Derivation Function (HKDF): RFC 5869 [S]. 2010.

[4] Barnes R, et al. Hybrid Public Key Encryption: RFC 9180 [S]. 2022.

[5] Apple. Keychain data protection [EB/OL].

[6] Biryukov A, et al. Argon2 Memory-Hard Function for Password Hashing and Proof-of-Work Applications: RFC 9106 [S]. 2021.

[7] 电车罗格. SecureCloudCrypto.swift：随机密钥、记录认证加密、恢复信封、设备封装与 Keychain 存储. 客户端源码.

[8] 电车罗格. EncryptedCloudModels.swift：载荷、编码与 UUID 规范化；VehicleCloudSyncService.swift：车辆公开资料；VaultSessionCoordinator.swift：账号数据空间. 客户端源码.

[9] 电车罗格. EncryptedCloudBackupService.swift 及相关设置、恢复入口：同步、恢复和迁移；LoggerCloudAuthService.swift：账号认证；docs/loggercloud-chargelog-v1.md：客户端协议.

实现与验证索引

算法与授权测试定义位于 ChargeLogTests/EncryptedCloudCryptoTests.swift，黄金向量位于 docs/crypto-v1-golden-vectors.json。主要覆盖 AES-GCM/HKDF、恢复词与 Argon2 参数、错误密码、AAD 替换、UUID 兼容、账号隔离、云端设备授权和传输字段。

文档的源码基线与摘要记录见同目录 provenance.json；实现日期为 2026-09-11。参考文献链接用于查询算法及平台规范。