arXiv:2603.07933v1 [cs.CR] 9 Mar 2026

通过休眠授权路径实现条件触发加密资产控制

Jian Sheng
Wang Yeah LLC
jason@yeah.app

2026年3月10日

摘要

传统上，加密数字资产的控制权被视为对私钥的永久持有——这种模式从根本上排除了在密码学层进行监管监督、条件委托和合法合规转让的可能。现有解决方案，包括多重签名方案、门限签名协议、基于智能合约的执行机制以及托管委托，都需要持久密钥暴露、链上状态变更或可信中介，因此无法仅通过密码学手段实现即时、可撤销且依赖于条件的控制。

我们提出条件触发休眠授权路径（CT-DAP），这是一种基于可销毁授权因子的加密资产控制方法。该方法以抽象的根可派生框架为参数，该框架满足三项属性：从单一根实体确定性地派生密钥、生成上下文隔离的控制能力，以及授权绑定撤销。在 CT-DAP 中，资产控制权被建模为休眠授权路径，由用户持有的凭证以及独立保管人持有的一个或多个管理授权因子构成。只有在所有必需的授权因子同时可用时，控制权才会在密码学上激活。

在预定义的外部条件——例如用户明确同意、法律认可继承事件或基于时间的触发器——得到验证后，相应的管理授权因子将被释放，从而激活相关控制路径。通过销毁授权因子即可实现撤销，使相应控制路径永久无法使用，同时无需改变底层密码学根。

我们形式化了威胁模型，定义了用于未授权控制抵抗性、路径隔离和无状态撤销的安全性游戏，并在标准假设下给出证明（AES-GCM-SIV 的带关联数据的认证加密安全性——IND-CPA 和 INT-CTXT——、随机预言机模型中 HKDF 的伪随机函数安全性、Argon2id 的内存困难性，以及 SHA-256 的抗碰撞性）。我们使用原子密码实体生成框架（ACE-GF）实例化该框架，并在商用硬件上评估性能，结果表明，在可配置的安全性与性能权衡下，激活延迟可低于一秒。

关键词： 加密资产控制；休眠授权路径；可销毁的 Au-
授权因子；条件激活；无状态撤销；密钥派生；ACE-GF

1 引言

1.1 背景与动机

加密数字资产的控制是现代密码学系统中的基础性问题。在主流设计中，控制权被默认等同于永久持有密码学秘密（通常是私钥）。虽然这种模式简单且自主性强，但它从根本上限制了现实场景下资产控制的表达能力。

而这些场景需要条件委托、监管监督、继承或具有法律强制效力的权利转让。

具体而言，现有加密资产系统缺乏原生机制来表示暂时不活跃、有条件启用或无需状态迁移即可撤销的控制权。因此，控制转换通常通过临时拼凑的方案实现，例如依赖可信托管人、在加密系统之外签订协议，或使用链上智能合约，而这些做法会引入额外的攻击面和操作复杂性。

随着加密资产日益融入受监管的金融系统、遗产规划和多方组织治理，这些局限性也愈发凸显。在这些场景中，控制权不仅必须安全地行使，还必须满足条件限制并符合规范。

1.2 现有方法的局限性

针对条件控制或共享控制的某些方面，人们提出了几类解决方案，但每一类都存在结构性局限。

永久持有密钥。 依赖单一控制私钥的系统不具备条件访问或委托的内在机制。任何控制权转移都需要直接披露或替换控制秘密，导致排他性不可逆地丧失，并使撤销变得不切实际 [5, 6]。

多重签名与门限方案。 多重签名构造 [26]与门限签名协议（例如 GG18 [11], GG20 [12], FROST [13]）将控制权分散到多个密钥和参与方，但要求所有参与密钥持续可用。控制条件的变化需要重新配置签名策略或迁移资产，从而带来操作开销。此外，这些方案将控制强制执行与参与要求混为一谈，使撤销和条件委托具有状态性且缺乏灵活性 [14]。

秘密共享。 Shamir 秘密共享 [10] 及其主动式变体 [15] 可以按照门限重构策略将主秘密分发给多方。然而，重构该秘密会将其泄露给执行重构的一方，且该方案本身不支持与外部事件绑定的条件激活，也不支持对单个份额的无状态撤销。

基于智能合约的强制执行。 智能合约支持可编程控制逻辑，包括条件执行，但代价是链上状态变更、代码不可变性风险以及对外部数据源的依赖 [17, 18]。通过合约实现的条件触发易受预言机操纵 [8], 、逻辑错误和司法管辖模糊性的影响，且往往无法使密码学强制执行与链下法律授权保持一致。

基于属性和基于策略的加密。 密文策略属性基加密（CP-ABE）[19] 和密钥策略属性基加密（KP-ABE） [20] 支持将细粒度访问控制策略嵌入密文或密钥中。尽管这些方案支持条件解密，但其设计目标是保障数据机密性，而非实现加密资产控制；此外，它们也未解决可撤销控制路径或无需重新加密的无状态撤销问题。

时间锁密码学。时间锁谜题 [22] 和可验证延迟函数（VDF） [23]能够按时间条件释放秘密。然而，它们仅支持时间条件（而非任意外部事件），且谜题生成后无法撤销。

托管模式。托管解决方案将控制权转交给可信中介，牺牲密码学自主性，并引入中心化信任假设。虽然托管人可以执行法律或监管条件，但密码学控制不再直接与资产持有人或受益人 [25] 挂钩。

社交恢复。 社交恢复钱包 [24] 允许指定的监护人协助密钥恢复，但混淆了恢复与持续控制，也不支持休眠状态下按条件激活的控制路径。

总的来说，这些方法将控制 与密钥持有、合约状态或机构信任混为一谈，无法为有条件且可撤销的资产控制提供统一的密码学抽象。

1.3 问题陈述

本研究要解决的核心问题是：

> 如何以一种在密码学上可强制执行、在法律上可审计，且无需持久密钥暴露、资产迁移或修改链上状态的方式，有条件地启用、撤销和转移加密数字资产的控制权？

一个令人满意的解决方案必须同时满足若干要求：

- **密码学可执行性：**无论链下协议如何，未授权方都必须无法行使控制权。
- **条件激活：**在满足预先定义的外部条件之前，控制权应保持未激活状态。
- **即时撤销：**必须能够禁用控制权，而无需轮换密钥或修改资产所有权结构。
- **无状态性：**控制转换不应要求维护可变全局状态。
- **法律兼容性：**密码学控制应符合获得法律认可的授权和委托流程。

现有系统至多只能满足这些要求中的一部分，而且往往需要引入额外的可信组件才能做到。

1.4 核心洞见

本文基于一个关键的概念性洞见：

> 无需转移加密资产的控制权；只需将其激活。

我们不将控制权等同于持有密码学秘密，而是将控制权建模为激活前就已存在的授权路径 。在所有必需的授权因子齐备之前，这些授权路径在密码学层面始终处于休眠状态。控制权的激活并非通过重新分配密钥或修改资产来实现，而是在可验证条件下启用预先建立的授权路径。

撤销通过销毁或使授权因子失效来执行，从而使相应的控制路径永久不可用，且不改变底层密码学根或资产身份。

1.5 提出的方法

我们提出条件触发休眠授权路径 （CT-DAP），一种基于可销毁授权因子的加密资产控制方法。该方法由一个抽象的根可派生框架 （定义 3.1）参数化，并满足三个性质：从单一根实体进行确定性派生、上下文隔离密钥生成，以及通过可销毁凭证组件实现授权绑定撤销。

在 CT-DAP 下，每项控制权被表示为一个由以下部分组成的休眠授权路径：

- 一个或多个用户持有的授权凭证，以及
- 一个或多个由独立保管人持有的管理授权因子。

控制权仅在所有必需的授权因子同时可用时才生效。管理授权因子可在外部事件得到验证后有条件地释放，例如用户明确同意、法律认可的继承条件或基于时间的触发器。

1.6 贡献

本文的核心贡献如下：

1. **形式模型**。 我们定义了具有形式语法的 CT-DAP 模型（第 3 节），规定了路径构造、激活与撤销算法，并以抽象根可推导框架进行参数化。2. **形式化安全分析**。 我们定义了威胁模型（第 3.2 小节）以及三项安全博弈——未授权控制抵抗性、跨路径隔离和无状态撤销——并在标准密码学假设下证明了安全性（第 7 节）。3. **授权绑定撤销**。 我们形式化了一种基于可销毁授权因子的即时、无状态撤销机制，从而解决了可撤销性与操作简便性之间长期存在的权衡。4. **具体实例化与评估**。 我们使用原子密码实体生成框架（ACE-GF） [1] 对 CT-DAP 进行实例化并评估性能，表明其激活延迟低于一秒，且安全性与性能权衡可配置（第 11 节）。5. **应用场景**。 我们展示了该方法在受监管托管、继承规划、委托控制与多方治理中的通用性。

1.7 组织

第 2 节综述相关工作。第 3 节定义系统模型、信任假设和形式语法。第 5 节介绍使用 ACE-GF 的授权路径构建。第 6 节描述条件触发式激活和撤销机制。第 7 节提供形式化安全分析。第 8 节将所提方法与现有方法进行比较。第 9 节介绍应用场景示例。第 10 节讨论实际考量、局限性和未来方向。第 11 节介绍性能基准测试。第 12 节总结全文。

2 相关工作

本节综述加密资产控制领域的现状，并阐明本研究相对于成熟方法和新兴方法的定位。

2.1 单密钥与分层确定性钱包

传统加密货币钱包将控制权等同于持有单个私钥或分层派生的密钥树（BIP-32 [5], BIP-39 [6], BIP-44 [7]）。这些系统简单易用，且用户拥有完全自主权，但不原生支持条件激活、委托或撤销。任何控制权转移都需要直接转移密钥或进行资产迁移，这两种方式都不可逆且操作成本高昂。

2.2 多重签名与门限签名方案

多重签名钱包 [26] 需要 m-of-n 个签名才能授权一笔交易。门限签名方案（TSS）——包括 GG18 [11], GG20 [12], 和 FROST [13]——通过在 n 个参与方之间分布密钥生成与签名，并设置阈值 t，消除了对链上多重签名脚本的需求。

尽管 TSS 减少了链上占用，但这两种方法都要求所有参与方的密钥份额持续可用，缺乏对休眠（有条件地未激活）控制权的原生支持，并且在控制条件发生变化时需要重新共享或迁移。主动秘密共享[15] 可以定期刷新份额，但无法解决与外部事件绑定的条件激活问题。

2.3 秘密共享与社交恢复

Shamir 秘密共享 [10] 支持通过 (t, n) 阈值重构主秘密。社交恢复钱包 [24] 通过指定监护人来应用这一原理，由他们共同恢复访问权限。

这些方案将恢复 与持续控制混为一谈：一旦秘密被重构，参与重构的一方就会获得完全且不可撤销的访问权限。这类方案既没有在保持休眠状态的同时实现条件激活的机制，也无法在不重新分发份额的情况下撤销特定监护人的权限。

2.4 基于属性和基于策略的访问控制

密文策略属性基加密（CP-ABE） [19]和密钥策略属性基加密（KP-ABE） [20] 将细粒度访问策略嵌入加密层。尽管在概念上与我们的条件激活理念相关，属性基加密方案旨在实现数据机密性（谁可以解密密文），而非资产控制（谁可以派生签名密钥或对资产行使密码学权限）。此外，属性基加密中的撤销通常需要重新加密或采用密钥更新机制[21], ，这与我们的无状态性要求相冲突。

2.5 基于智能合约的强制执行

智能合约 [17, 18] 支持控制策略的可编程链上执行，包括多重签名、时间锁和条件转移。然而，基于合约的控制依赖于链上状态变更，容易受到预言机操纵 [8], 的影响，并将控制语义绑定到特定区块链的执行模型。控制转换需要显式的资产迁移（例如，将代币转移到新的合约地址），这既不是无状态的，也不具备链无关性。

表 1：加密资产控制方法比较。 ✓= 原生支持，○ =部分支持或需变通方案， × = 不支持。

方法					
单密钥 / 分层确定性钱包	×	×	×	×	✓
多重签名 / 门限签名方案	○	×	×	○	○
秘密共享 / 恢复	○	×	×	×	✓
密文策略属性基加密 / 密钥策略属性基加密	✓	○	×	○	✓
智能合约	✓	○	×	○	×
时间锁 / 可验证延迟函数	○	✓	×	×	✓
多方计算托管	○	×	×	○	✓
CT-DAP（我们的方案）	✓	✓	✓	✓	✓

2.6 时间锁密码学与可验证延迟函数

时间锁谜题 [22] 和可验证延迟函数（VDF） [23] 提供了受时间条件约束的秘密释放机制。尽管它们对时间条件很有用，但仅限于单一触发类型（经过的时间），无法纳入任意外部事件（例如对继承的法律认可），并且谜题一旦生成便不可撤销。我们的方法支持任意条件类型和即时撤销。

2.7 基于多方计算的托管解决方案

机构托管平台（例如 [27]）使用多方计算（MPC）来分散签名权，而无需重构私钥。虽然多方计算托管能够确保完整私钥不被泄露，但它要求多方计算节点持续在线参与，且策略变更需要重新密钥化。我们的方法通过休眠路径实现条件控制，这些路径可以在授权因子持有者之间无需任何在线协调的情况下被激活或撤销。

2.8 本文定位

表 1 总结了 CT-DAP 相对于现有方法的定位。不同于以往工作，CT-DAP 提供了一种统一抽象，将资产身份与依赖条件的控制解耦，同时支持休眠权利、条件激活和无状态撤销，且无需链上状态或可信中介。

3 系统模型

本节定义 CT-DAP 方法的抽象系统模型、信任假设和形式语法。

3.1 实体与角色

系统涉及以下实体：

- **资产所有者（$\mathcal{O}$）：** 持有密码学根并建立授权路径的实体。$\mathcal{O}$ 持有用户持有的凭证 UserCred 和封装产物 SA。

- **保管人（$\mathcal{C}_i$）：** 一个独立实体，持有并有条件地释放单一管理授权因子 AdminFactor$_i$。托管人无法单方面行使资产控制。

- **受益人（$\mathcal{B}j$）：** 被指定在授权路径 $\mathcal{AP}j$ 激活后获得控制能力的实体。受益人在自我授权场景中也可充当资产所有者。

- **条件验证者（$\mathcal{V}$）：** 验证预定义外部条件是否满足并向保管人提供密码学签名证明的实体（或协议）。验证者可以是可信第三方（例如公证人、监管机构）或去中心化预言机网络 [8]。

3.2 信任模型与假设

我们形式化支撑 CT-DAP 安全性的信任假设。

假设 3.1（诚实但好奇的托管者）。 假定每个保管人 $\mathcal{C}_i$ 会忠实地持有并根据预定义释放条件有条件地释放 AdminFactor$_i$。然而，托管人可能试图从其所掌握的信息中获知其他授权因子或根秘密的相关信息。除非被明确建模，否则托管人不会相互串通，也不会与未授权方串通。

假设 3.2（有界对手）。 对手$\mathcal{A}$ 是一个概率多项式时间（PPT）算法，可能会:

（i）攻破给定路径上最多 $t < n$ 个托管人，并获取其 n 个授权因子（取得其管理员因子$_i$ 值）；（ii）获取密封工件 SA（例如，通过攻破存储）；（iii）观察其他路径$\mathcal{AP}_k$ ($k \neq j$) 中已激活的控制密钥；（iv）与条件验证者交互。

对于目标路径，对手不能同时获得 Cred$^{\text{composite}}$ 的所有组成部分。

假设 3.3（条件验证者完整性）。 条件验证者$\mathcal{V}$ 会正确评估预定义条件，并且仅在条件确实满足时提供证明。对于要求更强保证的部署，可以将$\mathcal{V}$ 实例化为采用多源聚合的去中心化预言机，以降低单点故障风险（参见第 6.2 小节）。

假设 3.4（密码学困难性）。 我们假设:

(i) 密封机制（例如 AES-256-GCM-SIV）同时提供 IND-CPA安全性（密文不可区分性）和 INT-CTXT安全性（密文完整性/认证）：使用错误密钥解密时，返回 $\perp$ 以压倒性概率[3];

(ii) HKDF-SHA256 [4] 在随机预言机模型中是伪随机函数（PRF）；(iii) Argon2id [2] 提供内存困难型密钥派生，并能抵御离线暴力破解攻击；(iv) 用于路径标识符的哈希函数 H（例如 SHA-256）具有抗碰撞性：对于任意概率多项式时间对手 $\mathcal{A}$，Pr[$\mathcal{A}$ 找到 $x \neq x'$ s.t. $H(x) = H(x')$] $\leq$ negl(λ)。

3.3 根可派生框架（抽象接口）

CT-DAP 由满足以下接口的根可派生框架 参数化 ce:

定义 3.1（根可派生框架）。根可派生框架 Π 是一个由概率多项式时间算法（设置，封装，解封，派生）组成的元组，满足：

- 设置(1^λ) $\to$ (REV，参数)：生成一个 256 位的根熵值和公开参数。
- 封装(参数，凭证$_{\text{复合}}$，REV) $\to$ SA：使用复合凭证加密 REV，生成一个密封工件。
- 解封(参数，SA，凭证$_{\text{复合}}$) $\to$ REV $\cup\{\perp\}$：当且仅当提供正确的复合凭证时，重构 REV；否则返回 $\perp$。
- 派生(REV，上下文) $\to$ 密钥$_{\mathcal{AP}}$：从 REV 和上下文元组 Ctx 确定性地派生出一个上下文隔离控制密钥。

定义 3.2（必需安全属性）。 框架 Π 必须满足：

P1 密封不可区分性：对于任意两个等长的根 REV_0,REV_1 ，任何不具备 $\text{Cred}^{\text{composite}}$ 的概率多项式时间对手都无法区分这两个密封制品

$$\text{Seal}(\text{params}, \text{Cred}_{\text{composite}}, \text{REV}_b), \quad b \in \{0,1\}$$

且优势不可忽略。

P2 解封正确性：解封(params，SA，Cred′) $= \perp$ 对所有 Cred′ $\neq \text{Cred}^{\text{composite}}$.

P3 上下文隔离：对于不同上下文 $\text{Ctx}_j \neq \text{Ctx}_k$，值派生(REV，Ctx$_j$)与派生(REV，Ctx$_k$)是计算独立的：任何获知其中一个的概率多项式时间对手，都无法以不可忽略的优势将另一个与均匀分布区分开。

3.4 形式语法

定义 3.3 (CT-DAP 语法)。 CT-DAP 方法由以下算法组成：

- PathSetup($\Pi, 1^\lambda, \text{UserCred}, \{\text{AdminFactor}_i\}_{i=1}^n$) $\to$ (SA$_{\text{根}}$ s, 参数)：建立根实体，生成随机盐 $s \in \{0,1\}_{128}$，并使用复合凭证封装根。返回密封工件、公开盐值和框架参数。
- PathDerive(Π, REV, Ctx$_j$) $\to$ Key$_{\mathcal{AP}_j}$：为路径 j 派生休眠控制密钥。仅在成功解封后调用此算法。
- Activate(Π, SA, UserCred, $\{\text{AdminFactor}_i\}_{i=1}^n$, Ctx$_j$) $\to$ Key$_{\mathcal{AP}_j}$ $\cup\{\perp\}$：通过解封重构 REV，派生 Key$_{\mathcal{AP}_j}$，将内存中的 REV 安全清零，并返回控制密钥。如果任何因子缺失或不正确，则返回$\perp$ 。
- 撤销(i) $\to${已撤销}：销毁管理员因子$_i$ （例如进行安全擦除），使得对于所有依赖管理员因子$_i$ 的路径，Cred$_{\text{复合}}$都永久无法重构。

定义 3.4（授权路径）。 第 j 条授权路径$\mathcal{AP}_j$ 定义为一个 4 元组 le:

$$\mathcal{AP}_j = (\text{SA}_{\text{root}},\ s_j,\ \text{UserCred}_j,\ \mathcal{F}_j)$$

其中：

- SA根：密封工件（REV根的持久化密文；根熵值 REV根本身从不持久化存储——参见算法 1）。

- $s_j \in \{0,\ 1\}^{128}$：Argon2id 的每路径随机盐。
- UserCred$_j$：路径专属的用户持有的凭证。
- $\mathcal{F}_j = \{\text{AdminFactor}_{j,\ 1},\ \ldots\ldots,\ \text{AdminFactor}_{j,\ n_j}\}$：路径 j 的可销毁管理因子集合。

当至少有一个 AdminFactor$_{j,\ i}$ 被扣留时，路径处于休眠状态。当所有组件同时可用时，路径处于激活状态，此时可通过 Unseal 临时重构 REV_{root} ，并随后派生出 $\text{Key}^{\mathcal{AP}_j}$。当至少有一个 AdminFactor$_{j,\ i}$被永久销毁时，路径处于撤销状态。

4 具体实例化：ACE-GF

我们使用原子密码实体生成框架（ACE-GF） [1], 这一无种子存储身份构造，实例化根可派生框架 II 。ACE-GF 满足定义 3.2 的全部三项必需属性（**P1–P3**）。

4.1 ACE-GF 的关键属性

1. **从根确定性派生的控制能力：**ACE-GF 将 REV 设为密码学根实体，它仅临时存在于内存中，绝不进行持久化存储。控制密钥通过 HKDF-SHA256 从 REV 确定性派生，并采用显式上下文编码 [4, 1]。

2. **授权绑定撤销与可销毁组件：**ACE-GF 通过由复合凭证加密的密封工件（SA），将对 REV 的访问绑定起来。销毁任何指定的凭证组件即可实现撤销，从而使 REV 永远无法重建 [1]。 3. **无种子存储操作：**ACE-GF 无需持久存储主秘密（例如 BIP-39 助记词）。静态存储的只有 SA——REV 的加密表示。 4. **抗随机数误用：**封装机制使用 AES-256-GCM-SIV [3], , 提供确定性的抗随机数误用认证加密。 5. **内存困难型凭证派生：**凭证组合通过 Argon2id [2], 处理，确保能够抵抗离线暴力破解攻击。

4.2 与 CT-DAP 的相关性

ACE-GF 将身份 （以 REV 为根）与授权 （由可销毁的凭证管理）分离，为休眠授权路径提供了基础抽象：

- REV 是所有授权路径共同的单一密码学控制源。
- ACE-GF 通过结构化上下文元组 Ctx = (算法标识符，域，索引)实现上下文隔离的密钥派生，从而支持并行且相互隔离的授权路径。
- ACE-GF 的授权绑定撤销机制直接实现了 CT-DAP 的核心机制——可销毁授权因子。

算法 1 PathSetup：建立根实体和休眠路径

要求： 安全参数 1^λ；用户凭证 UserCred；因子 $\{$管理员因子$_i\}_{i=1}^n$**确保：** 密封工件 $\mathrm{SA}_{\mathrm{root}}$；每路径盐 $s \in \{0,1\}^{128}$；公共参数 params**步骤 1：建立根实体**1: (REV$_{\mathrm{root}}$, params) $\leftarrow$ ACE-GF。设置(1^λ)2: $s \xleftarrow{\$} \{0,1\}^{128}$ ▷每路径随机盐

3: Cred$_{\mathrm{composite}} \leftarrow$ Argon2id(UserCred $\|$ 管理员因子$_1 \| \cdots \|$ 管理员因子$_n$, s, m, t, p) ▷内存困难型派生4: $\mathrm{SA}_{\mathrm{root}} \leftarrow$ ACE-GF。封装(params, Cred$^{\mathrm{composite}}$, REV$_{\mathrm{root}}$)

步骤 2：（可选）验证密封工件▷ 根密钥清零前进行自检5: REV$' \leftarrow$ ACE-GF。解封(params, $\mathrm{SA}_{\mathrm{root}}$, Cred$^{\mathrm{composite}}$)6: **断言**REV$' =REV_{\mathrm{root}}$ ▷ 若封装/解封往返失败，则中止7: 清零(REV$'$, REV$_{\mathrm{root}}$) ▷ 根仅临时存在

步骤 3：封装路径以供休眠

8: 将 $\mathrm{SA}_{\mathrm{root}}$ 编码为标准 BIP-39 助记词（密文编码）9: 将盐值 s 与 $\mathrm{SA}_{\mathrm{root}}$ 一同以明文存储10: 通过安全信道将管理员因子$_i$ 分发给各自的托管人 $\mathcal{C}_i$ *via secure channels11:* **返回**($\mathrm{SA}_{\mathrm{root}}$, s, params)

注意：上下文隔离控制密钥密钥$^{\mathcal{AP}_j}$ 在设置时不会被派生。它们在激活 （算法*2*）期间按需派生，该过程会临时重构REV$_{\mathrm{root}}$ ，来源为$\mathrm{SA}_{\mathrm{root}}$。

5 授权路径构建

本节形式化休眠授权路径的构造——CT-DAP的核心密码学抽象——并通过ACE-GF加以实例化。

5.1 构造概述

每条授权路径是由用户持有凭证与托管人管理的可销毁授权因子组成的复合结构，并锚定到ACE-GF中的单个REV。该构造分三步进行，并在算法 1中加以形式化。

5.2 凭证组合

路径 j 的复合凭证通过内存困难密钥派生函数处理用户凭证及所有管理因素而构建，该函数使用路径特定随机盐 s_j:

$$\mathsf{Cred}_{\mathrm{composite},j} = \mathsf{Argon2id}\big(\underbrace{\mathsf{UserCred} \,\|\, \mathsf{AdminFactor}_1 \,\|\, \cdots \,\|\, \mathsf{AdminFactor}_n}_{\text{password}},\ s_j,\ m,\ t,\ p\big) \qquad (1)$$

其中 $\|$ 表示使用非标准分隔符（例如空字节 0x00）进行拼接，以确保凭证组件之间的严格域分离，并且:

- $s_j \xleftarrow{\$} \{0,1\}^{128}$ 是每路径随机盐，以明文形式与 SA根 一同存储。盐值可防止跨目标摊销：攻击路径 j 的对手无法重复利用已有工作（例如预计算表或并行 GPU 批次）同时攻击路径 $k \neq j$，即使底层凭证组件完全相同。

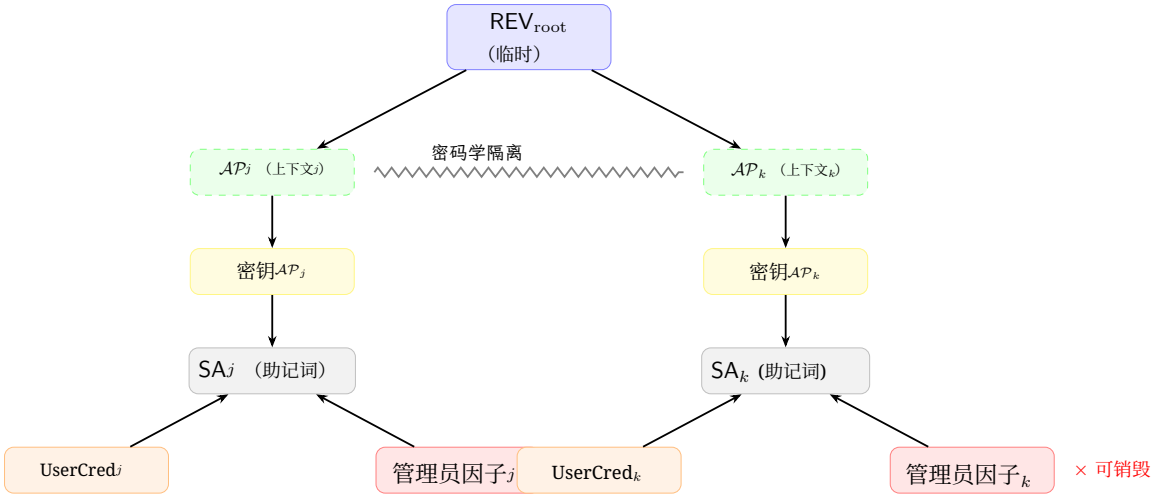


图1：授权路径的隔离与冗余。每条路径均使用唯一的上下文元组和独立的授权因子。销毁某个因子会撤销相应路径，而不影响根或其他路径。

- $(m,\ t,\ p)$ 是 Argon2id 的内存、时间和并行度参数，根据部署策略选择（例如 $m = 2^{19}$ KiB、$t = 3$、$p = 4$，依据 RFC 9106 [16]）。

使用 Argon2id [2] 作为内存困难型密钥派生函数，可确保抵御离线暴力破解攻击；即使对手获得 SA$_{\text{root}}$ 和^{s_j}，只要缺失的凭证组件具有足够的最小熵（≥ 128 位），这一保证依然成立。

5.3 密封工件作为休眠助记词

与助记词直接表示明文主种子的传统系统（例如 BIP-39）不同，**密封工件（SA）以标准 BIP-39 助记词的形式呈现**，其中编码的是 REV$_{\text{root}}$ 的密文，而非明文熵。

若没有所需的复合凭证，SA 在功能上与随机噪声不可区分，无法用于派生任何控制能力。使用 AES-GCM-SIV 进行封装，可确保加密具有确定性且抗随机数误用，从而在凭证与授权路径之间建立稳定、可审计的映射。

5.4 跨路径隔离保证

根据 ACE-GF 的上下文隔离性质（**P3**），从同一个 REV$_{\text{root}}$ 派生但上下文值不同的控制密钥在计算上相互独立：

$$\forall j \neq k, \quad \mathrm{Adv}_{\mathcal{A}}(\mathsf{Key}_{\mathcal{AP}_j} \mid \mathsf{Key}_{\mathcal{AP}_k}) \leq \mathsf{negl}(\lambda) \tag{2}$$

这确保即使受益人的路径 $\mathcal{AP}_k$ 已激活或其凭证遭到泄露，其他路径的安全性仍然不受影响。

6 条件触发与激活

管理授权因子并非持续可用。其释放取决于对预定义外部事件的验证。

6.1 触发条件

支持的条件类型包括但不限于：

- **用户明确同意：** 资产所有者的直接授权（例如，签署的请求）。
- **法律事件：** 经认证实体（例如公证人、遗嘱认证法院）核实的、受法律认可的继承或继任事件。
- **时间条件：** 基于时间的期限或司法管辖期限，可通过可信时间戳或链上时间锁进行验证。
- **监管行动：** 监管机构要求暂停或出于合规需要释放。

6.2 条件验证

条件验证可通过两种互补机制实现：

可信第三方（TTP）验证者。 对于具有法律约束力的条件，由经认证的实体（如公证人、监管机构）担任验证者，向托管人提供经加密签名的条件履行证明。

去中心化预言机。 对于透明且防篡改的条件（如时间锁、跨链事件），去中心化预言机网络 [8] 可取代 TTP，在没有单点故障的情况下提供条件验证。通过多源数据聚合可降低预言机操纵风险。

6.3 激活协议

激活协议以算法2形式化定义。

6.4 撤销协议

撤销在算法 3 中进行了形式化定义。

6.5 激活的不可逆性

激活后，只要 $\mathsf{Cred}^{\mathrm{composite}}$ 保持完整，授权路径就仍然可用。要禁用该路径，保管人会销毁其管理员因子$_i$（例如通过零化进行安全擦除），使 $\mathsf{Cred}^{\mathrm{composite}}$ 永久保持不完整。

7 安全分析

本节对 CT-DAP 进行形式化安全性分析。我们首先说明 Derive 所使用的规范编码，然后陈述三个辅助引理，以刻画各项密码学优势，最后通过组合这些引理并给出明确的优势界，证明主要安全性定理。

算法2 激活：条件触发路径激活

要求： 密封工件 $\mathrm{SA}_{\mathrm{root}}$；用户凭证 UserCred；已释放因子 $\{$管理员因子$_i\}_{i=1}^{n}$；
目标上下文 Ctx$_j$；公共参数 params
确保： 控制密钥 Key$^{\mathcal{AP}_j}$ 或 $\perp$

步骤1：因子释放
对于每个保管人 $\mathcal{C}_i$ **执行**
通过$\mathcal{V}$验证条件
管理员因子$_i \leftarrow$ ACE-GF.解封(params, SA$^{\mathcal{C}_i}$,Cred$^{\mathcal{C}_i}$) ▷ 保管人释放因子
通过安全通道传输管理员因子$_i$ （例如，采用证书固定的 TLS 1.3）
结束 for

步骤 2：复合凭证重建
$\mathrm{Cred}_{\mathrm{composite}} \leftarrow$ Argon2id(UserCred $\|$ 管理员因子$_1 \| \cdots \|$ 管理员因子$_n$； s； m, t, p) ▷ 盐值 s
从 SA 元数据读取

步骤 3：REV 重建与密钥派生
$\mathrm{REV}_{\mathrm{root}} \leftarrow$ ACE-GF.解封(params,$\mathrm{SA}_{\mathrm{root}}$, Cred$^{\mathrm{composite}}$)
如果 $\mathrm{REV}_{\mathrm{root}} = \perp$ **则**
返回 $\perp$ ▷ 凭证不匹配
结束 if
密钥$^{\mathcal{AP}_j} \leftarrow$ ACE-GF。派生($\mathrm{REV}_{\mathrm{root}}$，上下文$_j$)
清零($\mathrm{REV}_{\mathrm{root}}$) ▷ 维持临时根属性
返回密钥$^{\mathcal{AP}_j}$

算法 3 撤销：授权绑定路径撤销

要求： 目标授权因子索引 i；保管人 $\mathcal{C}_i$
确保： 依赖于管理员因子$_i$ 的路径将被永久撤销
$\mathcal{C}_i$ 对管理员因子$_i$ 执行安全擦除（例如，零化存储介质）
▷ 凭据$^{\mathrm{composite}}$ 现在已永久无法重构
▷ $\mathrm{REV}_{\mathrm{root}}$ 无法为需要管理员因子$_i$的路径解封
记录销毁事件以确保可审计性（例如，带时间戳证明）

7.1 用于派生的规范上下文编码

为确保不同实现之间具有明确的域分离，我们要求对传递给 HKDF 的上下文元组采用规范的字节级编码。具体而言，对于上下文 Ctx$_j$ =(AlgID、Domain、Index)，派生函数计算：

$$\mathsf{PRK} = \mathrm{HKDF\text{-}Extract}(\mathsf{salt} = \texttt{"CT-DAP-v1"},\ \mathsf{IKM} = \mathsf{REV}_{\mathrm{root}}) \tag{3}$$

$$\mathsf{Key}_{\mathcal{AP}_j} = \mathrm{HKDF\text{-}Expand}(\mathsf{PRK},\ \mathsf{info} = \mathsf{Encode}(\mathsf{Ctx}_j),\ L) \tag{4}$$

其中，信息字符串是带长度前缀的拼接：

$$\mathsf{Encode}(\mathsf{Ctx}_j) = \langle|\mathsf{AlgID}|\rangle_2 \,\|\, \mathsf{AlgID} \,\|\, \langle|\mathsf{Domain}|\rangle_2 \,\|\, \mathsf{Domain} \,\|\, \langle\mathsf{Index}\rangle_4$$

其中，$\langle\cdot\rangle_k$ 表示将其编码为大端格式的 k 字节。固定宽度的长度前缀确保任意两个不同的元组（AlgID、Domain、Index）$\neq$（AlgID′、Domain′、Index′）不会生成相同的信息字符串，从而防止域分离碰撞。静态盐值 `"CT-DAP-v1"` 将派生绑定到此协议版本；未来版本将递增版本标签。

7.2 辅助引理

我们将主证明中出现的三种密码学优势单独列出。

引理 7.1 (密封不可区分性)。 对于任意两个长度相等的根 REV_0,REV_1 ，令SA_b = *ACE-GF.*封装(参数, 凭证$^{\text{composite}}$,REV_b)，其中 $b \xleftarrow{\$} \{0,1\}$。对于任何不知道 凭证$^{\text{composite}}$ 的概率多项式时间算法 $\mathcal{A}$:

$$\mathrm{Adv}^{\mathrm{SEAL}}_{\mathcal{A}}(\lambda) \;=\; \left|\Pr[\mathcal{A}(\mathsf{SA}_b)=b]-\tfrac{1}{2}\right| \;\le\; \mathrm{Adv}^{\text{IND-CPA}}_{\mathcal{A},\,\text{AES-GCM-SIV}}(\lambda)$$

其中，右侧是针对 *AES-256-GCM-SIV* 的 *IND-CPA* 优势 *[3]* .

证明。 我们将问题归约到 AES-256-GCM-SIV 的 IND-CPA 安全性。

归约。 给定一个用于封装实验的概率多项式时间区分器 $\mathcal{A}$，按如下方式构造一个 IND-CPA 对手$\mathcal{B}$：

(i) $\mathcal{B}$ 接收带关联数据的认证加密公共参数，以及一个使用 IND-CPA 挑战者均匀随机选择的未知密钥 K 的加密预言机 $\mathsf{Enc}_K(\cdot)$。

(ii) $\mathcal{B}$ 设置 $K = \mathsf{Cred}^{\text{composite}}$ （密钥对$\mathcal{B}$未知）。它选择两个等长的根 $\mathsf{REV}_0, \mathsf{REV}_1$，并将它们作为挑战明文 $(m_0, m_1) = (\mathsf{REV}_0, \mathsf{REV}_1)$ 提交。

(iii) IND-CPA 挑战者返回 $c^* = \mathsf{Enc}_K(\mathsf{REV}_b)$，其中随机均匀选取 $b \xleftarrow{\$} \{0,1\}$。

(iv) $\mathcal{B}$ 设置 $\mathsf{SA}^* \leftarrow c^*$ 并将其转发给$\mathcal{A}$。

(v) $\mathcal{A}$ 输出一个猜测 b'。$\mathcal{B}$ 输出 b'。

分析。 根据构造，$\mathcal{A}$ 与$\mathcal{B}$ 交互时的视图与封装实验完全相同：SA^* 是在 $\mathsf{Cred}^{\text{composite}}$下对 REV_b 进行的带关联数据的认证加密。因此，$\Pr[\mathcal{B}$ 获胜 $] = \Pr[\mathcal{A}(\mathsf{SA}_b) = b]$，并且

$$\mathrm{Adv}^{\mathrm{SEAL}}_{\mathcal{A}}(\lambda) = \left|\Pr[\mathcal{A}(\mathsf{SA}_b)=b]-\tfrac{1}{2}\right| = \mathrm{Adv}^{\text{IND-CPA}}_{\mathcal{B},\,\text{AES-GCM-SIV}}(\lambda)$$

根据假设 3.4(i)，该值为 $\mathsf{negl}(\lambda)$。 □

引理 7.2 （凭证抗暴力破解性）。 设复合凭证为

$$\mathsf{Cred}_{\text{composite}} = \mathsf{Argon2id}\big(\mathsf{UserCred} \| \mathsf{AdminFactor}_1 \| \cdots \| \mathsf{AdminFactor}_n;\; s;\; m,t,p\big)$$

其中至少一个组成部分的最小熵为 $\ge \kappa$ 位，且 s 是一个 *128* 位的每路径随机盐。对于任何对手 $\mathcal{A}$，若其向 *Argon2id* 求值预言机发起至多 Q 次查询（每次需消耗 $\ge m$ 字节内存）：

$$\mathrm{Adv}^{\mathrm{MH}}_{\mathcal{A}}(\lambda) \;\le\; \frac{Q}{2^{\kappa}}$$

每路径盐确保，针对目标(s_j,$\mathsf{SA}j$) 所做的工作，不会对满足 $s_k \neq s_j$的另一不同目标(s_k,SA_k) 提供任何非平凡优势：对手无法跨目标分摊查询，而必须针对每个目标以独立的 Q 次求值预算重新开始。

证明。 我们将 Argon2id 建模为一个随机预言机$\mathcal{O}$，它对每个新查询（x, s, m, t, p）返回$\{0,1\}^{256}$中的均匀随机输出。

单目标界。 对手$\mathcal{A}$ 至多知道 $n+1$ 个凭证组件中的 $n-1$ 个{UserCred, 管理员因子$_1$,…, 管理员因子$_n$}。缺失组件的最小熵为 $\ge \kappa$，因此每次以猜测值查询$\mathcal{O}$ 时，与真实 $\mathsf{Cred}^{\text{composite}}$ 匹配的概率至多为$2^{-\kappa}$。在 Q 次查询上，并合界给出

$$\Pr[\exists\, i \le Q : \mathcal{O}(x_i, s, m, t, p) = \mathsf{Cred}_{\text{composite}}] \;\le\; \frac{Q}{2^{\kappa}}$$

内存困难性成本。 每次 Argon2id 计算都需要 $\ge m$ 字节的工作内存 [2,16]。总内存为 M 的对手，每个时间步最多只能执行 $\lfloor M/m \rfloor$ 次并行计算，从而限制可实现的查询速率，并使大规模 Q 在资源上难以承受。

跨目标独立性。每条路径都独立抽取随机盐 s_j。由于$\mathcal{O}$ 被建模为随机预言机，使用盐值 s_j 和盐值 $s_k \neq s_j$ 的查询会产生相互独立的输出。因此，针对目标 (s_j,SA$_j$) 所做的工作不会给针对 (s_k,SA$_k$) 带来任何非平凡优势：对手必须为每个目标分别投入 Q 次计算的预算。 □

注记 7.1（关于 Argon2id 的随机预言机模型）。上述证明将 Argon2id 建模为随机预言机，这是 Argon2 规范本身 [2] 以及 RFC 9106 [16]所共同采用的标准理想化。实践中，Argon2id 并非随机预言机；其安全性依赖于内存困难性——即任何比诚实计算方更快计算该函数的算法都必须使用相当的内存。在标准模型中形式化内存困难性需要基于图 pebbling 的框架 [29], ，这些框架给出渐近意义上更弱但更为现实的界，其形式为每次求值的代价 cost($\mathcal{A}$) $\geq \Omega(m \cdot t)$。

因此，我们基于随机预言机模型的界 $Q/2^{\kappa}$ 应被解释为在线查询优势的上界，并辅以每次查询耗费 $\geq m$ 字节内存这一经济性论证。对 Argon2id 内存困难性的完整标准模型分析是一项独立的研究问题，超出本工作的范围；关于目前最先进的论述，请读者参阅 Alwen 和 Blocki [29]。

引理 7.3 (派生的伪随机函数安全性。设 派生 按第 *7.1* 小节实例化为 *HKDF-SHA256*。对于最多发出 q 次查询且查询使用互不相同的规范信息字符串的任意概率多项式时间算法 A:

$$\mathrm{Adv}^{\mathrm{PRF}}_{\mathcal{A},\mathrm{HKDF}}(\lambda) \;\leq\; q \cdot \mathrm{Adv}^{\mathrm{PRF}}_{\mathrm{HMAC\text{-}SHA256}}(\lambda)$$

在 *SHA-256* 的随机预言机模型中，右侧为 可忽略函数(λ) *[4]*。

证明。我们将问题归约到 HMAC-SHA256 的伪随机函数安全性。

HKDF 结构。HKDF 包含两个阶段 [4]: PRK = HKDF-Extract(盐值,输入密钥材料) 和 OKM = HKDF-Expand(PRK, 信息字符串 L)。对于固定的 PRK，HKDF-Expand 是以 PRK 为密钥对 HMAC-SHA256 的迭代应用。因此，它是从信息空间到$\{0,1\}^L$的伪随机函数，其优势由 HMAC-SHA256 的伪随机函数优势所界定。

混合论证。考虑一系列 $q+1$ 混合游戏 H_0，H_1、.....、H_q。在 H_0 中，所有q输出 Key$^{\mathcal{AP}_1}$、.....、Key$^{\mathcal{AP}_q}$均通过 HKDF-Expand 按诚实方式计算。在 H_i 中，前 i 个 $q-i$ 输出被替换为相互独立的均匀随机字符串，其余部分仍按诚实方式计算。在 H_q 中，所有输出均为均匀随机字符串。

规范编码（第 7.1 小节）确保不同上下文 Ctx$_j$ ≠Ctx$_k$ 会产生不同的信息字符串 Encode(Ctx$_j$) $\neq$ Encode(Ctx$_k$)。因此，每次转换都会 $H_{i-1} \to H_i$ 在一个新输入上，将一个伪随机函数输出替换为一个随机字符串，Adv$^{\mathrm{PRF}}_{\mathrm{HMAC\text{-}SHA256}}$无法区分两者。对 q 所有转换求和:

$$\mathrm{Adv}^{\mathrm{PRF}}_{\mathcal{A},\mathrm{HKDF}}(\lambda) \;\leq\; q \cdot \mathrm{Adv}^{\mathrm{PRF}}_{\mathrm{HMAC\text{-}SHA256}}(\lambda)$$

In t在 SHA-256 的随机预言机模型下，右侧对于多项式 q. □

7.3 未授权控制抵抗性

定义 7.1 （UCR游戏）。 未授权控制抵抗性游戏 Game$^{\mathcal{A}}_{\mathrm{UCR}}$(λ) 以从有效路径上下文集合 {Ctx$_1$,. . . , Ctx$_m$}中均匀选取的挑战上下文Ctx* 作为参数，流程如下:

1. **设置**。 挑战者运行 PathSetup，生成 (SA$_{\mathrm{root}}$, s, params)，其中凭证组件为 (UserCred, 管理员因子$_1$,. . . , 管理员因子$_n$)，路径上下文为{Ctx$_1$,. . . , Ctx$_m$*}*。挑战者计算 *KeyDerive(REV,* Ctx*)。

2. **腐化**。 $\mathcal{A}$ 选择一个适当的子集 $S \subsetneq$ {UserCred、管理员因子$_1$、.....、管理员因子$_n$}（即至少有一个组件被保留）。挑战者公开 S 中的所有组件。

3. **公开信息**。$\mathcal{A}$ 还会接收 $\mathrm{SA}_{\mathrm{root}}$、params、 s、上下文标签集合 $\{\mathsf{Ctx}_j\}_{j=1}^{m}$，以及对 $\mathsf{Derive}(\mathrm{REV}_{\mathrm{root}}, \cdot)$ 的预言机访问，适用于非挑战上下文 $\mathsf{Ctx}_j \neq \mathsf{Ctx}^*$ （用于建模在其他路径上的合法密钥使用）。

4. **输出**。$\mathcal{A}$ 输出一个候选密钥 K^*。

5. **获胜条件**。$\mathcal{A}$ 获胜，当且仅当 $K^* = \text{密钥}_{\mathcal{AP}_*}$。

定理 7.4 （未授权控制抵抗性）。 在假设 *3.4* 下，对于任意概率多项式时间对手 A：

$$\Pr[\mathsf{Game}^{\mathcal{A}}_{\mathrm{UCR}}(\lambda) = 1] \;\leq\; \mathrm{Adv}^{\mathrm{SEAL}}(\lambda) \;+\; \mathrm{Adv}^{\mathrm{MH}}(\lambda) \;+\; \mathrm{Adv}^{\mathrm{PRF}}(\lambda) \;\leq\; \mathsf{negl}(\lambda)$$

其中这　三个项分别对应定理 *7.1*、定理 *7.2* 和定理 *7.3*　.

证明。 我们通过一系列游戏展开分析，跟踪每次转换所引入的优势。**游戏 0** （真实游戏）。即上文定义的真实 UCR游戏。

游戏 1 （AEAD 混合步骤）。将 $\mathrm{SA}_{\mathrm{root}}$ 替换为对相同长度的均匀随机值 REV^* 的加密。根据定理 7.1：

$$\big|\Pr[\mathrm{G1}] - \Pr[\mathrm{G0}]\big| \;\leq\; \mathrm{Adv}^{\mathrm{SEAL}}(\lambda) \;\leq\; \mathrm{Adv}^{\text{IND-CPA}}_{\text{AES-GCM-SIV}}(\lambda)$$

游戏 2 （内存困难性混合步骤）。在游戏 1 中，$\mathcal{A}$ 必须恢复正确的 $\mathsf{Cred}^{\mathrm{composite}}$ 才能解封。由于$\mathcal{A}$ 最多持有 $n-1$ 个组件，且缺失组件具有最小熵$\geq \kappa$，根据定理 7.2：

$$\big|\Pr[\mathrm{G2}] - \Pr[\mathrm{G1}]\big| \;\leq\; \mathrm{Adv}^{\mathrm{MH}}(\lambda) \;\leq\; \frac{Q}{2^{\kappa}}$$

游戏 3 （伪随机函数混合步骤）。以$\mathcal{A}$ 未 恢复 $\mathrm{REV}_{\mathrm{root}}$ 的事件为条件（互补事件由游戏 1–2 中的 $\mathrm{Adv}^{\mathrm{SEAL}}$ +$\mathrm{Adv}^{\mathrm{MH}}$ 所界定）。由于$\mathcal{A}$ 不持有 $\mathrm{REV}_{\mathrm{root}}$，挑战密钥 $\mathrm{Key}^{\mathcal{AP}_*}$ = $\mathsf{Derive}(\mathrm{REV}_{\mathrm{root}},\mathsf{Ctx}^*)$ 是以未知的 $\mathrm{REV}_{\mathrm{root}}$为密钥的伪随机函数的输出。根据定理 7.3：

$$\Pr[\mathrm{G3}] \;\leq\; \mathrm{Adv}^{\mathrm{PRF}}(\lambda)$$

将所有转换相加：$\Pr[\mathsf{Game}^{\mathcal{A}}_{\mathrm{UCR}} = 1] \leq \mathrm{Adv}^{\mathrm{SEAL}}$ +$\mathrm{Adv}^{\mathrm{MH}}$ +$\mathrm{Adv}^{\mathrm{PRF}} \leq \mathsf{negl}(\lambda)$。 □

7.4 跨路径隔离

定义　第 7.2 节 (CPI 博弈)。 跨路径隔离 博弈 $\mathsf{Game}^{\mathcal{A}}_{\mathrm{CPI}}(\lambda)$按如下方式进行　s:

1. 挑战者生成 $\mathrm{REV}_{\mathrm{root}}$ ，并使用第 7.1 小节的规范编码，为不同上下文 $\mathsf{Ctx}_j \neq$ 和 Ctx_k 派生密钥 $\mathrm{Key}^{\mathcal{AP}_j}$ 和 $\mathrm{Key}^{\mathcal{AP}_k}$ 。

2. 对手收到 $\mathrm{Key}^{\mathcal{AP}_j}$ （即，路径 j 已泄露或已激活）。

3. 挑战者采样 $b \xleftarrow{\$} \{0,\ 1\}$。若 $b = 0$，则发送 $\mathrm{Key}^{\mathcal{AP}_k}$；若 $b = 1$，则发送一个与其长度相同的均匀随机密钥 K^* 。

4. $\mathcal{A}$ 输出猜测 b'。

5. $\mathcal{A}$ 若满足 $b' = b$，则获胜。

定理 7.5 (跨路径隔离)。 在假设 *3.4(ii)* 下，对于任意概率多项式时间对手 A：

$$\big|\Pr[b' = b] - \tfrac{1}{2}\big| \;\leq\; \mathrm{Adv}^{\mathrm{PRF}}_{\mathrm{HKDF}}(\lambda) \;\leq\; \mathsf{negl}(\lambda)$$

证明。 令 $\mathsf{PRK} = \text{HKDF-Extract}(\texttt{"CT-DAP-v1"}, \mathsf{REV}_{\text{root}})$。则：

$$\mathsf{Key}_{\mathcal{AP}_j} = \text{HKDF-Expand}(\mathsf{PRK}, \mathsf{Encode}(\mathsf{Ctx}_j), L)$$
$$\mathsf{Key}_{\mathcal{AP}_k} = \text{HKDF-Expand}(\mathsf{PRK}, \mathsf{Encode}(\mathsf{Ctx}_k), L)$$

由于 $\mathsf{Ctx}_j \neq \mathsf{Ctx}_k$ 且 Encode 是单射（第 7.1 小节），这两个信息字符串是不同的。

假设 $\mathcal{A}$ 在给定 $\mathsf{Key}^{\mathcal{AP}_j}$ 的情况下，以优势 ϵ 区分 $\mathsf{Key}^{\mathcal{AP}_k}$ 与随机值。我们构造一个能够访问预言机的伪随机函数区分器 $\mathcal{D}$，其预言机要么是 $F_{\mathsf{PRK}}(\cdot)$，要么是真正的随机函数 $R(\cdot)$。$\mathcal{D}$ 查询 $\mathsf{Encode}(\mathsf{Ctx}_j)$ 和 $\mathsf{Encode}(\mathsf{Ctx}_k)$，将第一个响应转发给 $\mathcal{A}$ 作为 $\mathsf{Key}^{\mathcal{AP}_j}$，并使用 $\mathcal{A}$ 对第二个响应给出的区分结果。由此得到 $\mathsf{Adv}^{\text{PRF}}_{\mathcal{D}} \geq \epsilon$，这与定理 7.3 矛盾。 □

7.5 触发限制性与释放记录绑定

一个关键问题是，对手可能重放或混淆释放记录，诱使保管人在条件实际上并未满足的情况下释放管理员因子$_i$ 。我们将验证者的证明与激活输入之间所需的绑定关系形式化。

定义 7.3（释放记录）。 对于每次因子释放，条件验证者 $\mathcal{V}$ 都会生成一份签名证明：

$$\mathsf{Rel}_{j,i} = \mathsf{Sign}\big(\mathsf{sk}_{\mathcal{V}},\ (\mathsf{PathID}_j \,\|\, \mathsf{ConditionHash} \,\|\, \mathsf{Timestamp} \,\|\, \mathsf{Nonce})\big)$$

其中，$\mathsf{PathID}_j = H(\mathsf{SA}_{\text{root}} \| \mathsf{Ctx}_j)$将证明绑定到特定的密封工件和路径上下文，而 $\mathsf{Nonce} \xleftarrow{\$} \{0,1\}^{128}$ 是一次性使用的值。

定义 7.4（触发限制性博弈）**：** 如果对手 $\mathcal{A}$ 能够生成有效的释放记录 $\mathsf{Rel}^*_{j,i}$，使保管人 $\mathcal{C}_i$ 在路径 $_j$的条件尚未满足时释放该路径的管理员因子 $\mathsf{AdminFactor}_i$，则其赢得博弈 $\mathsf{Game}^{\mathcal{A}}_{\text{TR}}(\lambda)$。

定理 7.6（触发限制性）**：** 在验证者签名方案具备 *EUF-CMA* 安全性、H 具备抗碰撞性（假设 *3.4(iv)*），且满足假设 *3.3*的条件下，对于针对大小为 $N = 2^\lambda$的随机数空间、最多发出 $q_{\text{rel}} = \text{poly}(\lambda)$ 次释放查询的任意概率多项式时间对手 A：

$$\Pr\big[\mathsf{Game}^{\mathcal{A}}_{\text{TR}}(\lambda) = 1\big] \;\leq\; \underbrace{\mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Sig}}(\lambda)}_{forgery} + \underbrace{\mathsf{Adv}^{\text{CR}}_{H}(\lambda)}_{cross\text{-}path\ collision} + \underbrace{\frac{q^2_{\text{rel}}}{2^{\lambda+1}}}_{nonce\ collision}$$

The 前两项根据假设为 $\mathsf{negl}(\lambda)$ 。对于第三项，由于 $q_{\text{rel}} = \text{poly}(\lambda)$：

$$\frac{q^2_{\text{rel}}}{2^{\lambda+1}} \;\leq\; \frac{\text{poly}(\lambda)^2}{2^{\lambda+1}} \;=\; \mathsf{negl}(\lambda)$$

因此，整个界限都是可忽略的。这三项分别对应不同的攻击策略：

- 伪造： 在验证者不配合的情况下生成有效签名。
- 跨路径重放： 找到哈希碰撞 $H(\mathsf{SA}_{\text{根}} \| \mathsf{Ctx}_j) = H(\mathsf{SA}_{\text{根}} \| \mathsf{Ctx}_k)$，以便 $j \neq k$跨路径重放证明。
- 时间重放： 重用被保管人的本地去重状态接受过的随机数，其概率受随机数空间上的生日界约束。

证明。 保管人 $\mathcal{C}_i$ 通过检查以下内容来验证 $\mathsf{Rel}_{j,\ i}$ ： (a) 使用 $\mathsf{pk}_{\mathcal{V}}$对应的密钥生成的签名，(b) PathID_j与本地存储的 $H(\mathsf{SA}_{\mathrm{root}}\|\mathsf{Ctx}_j)$相匹配，以及 (c) 该随机数此前未被接受。我们对 $\mathcal{A}$ 绕过每项检查的概率进行界定。

抗伪造性（检查 *a*）： 若要在没有验证者配合的情况下生成有效的 $\mathsf{Rel}^{*}_{j,\ i}$ 就必须伪造由 $\mathsf{sk}_{\mathcal{V}}$生成的签名，贡献 $\mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{Sig}}$。

跨路径重放（检查 *b*）： 路径 k 的有效证明 $\mathsf{Rel}_{k,\ i}$ 包含 $\mathsf{PathID}_k = H(\mathsf{SA}_{\mathrm{root}}\|\mathsf{Ctx}_k)$。由于 $\mathsf{Ctx}_j \neq \mathsf{Ctx}_k$ 意味着 $\mathsf{PathID}_j \neq \mathsf{PathID}_k$，除非发生概率为 $\mathsf{Adv}^{\mathrm{CR}}_{H}$的情况（依据假设 3.4(iv)），保管人的检查 (b) 会拒绝重放的协议记录。

时间重放（检查 *c*）： 保管人维护一个本地已接受随机数集合。在 q_{rel} 条释放记录中发生随机数碰撞的概率至多为 $q_{\mathrm{rel}}^2/2N$，依据生日界。若未发生碰撞，则每个合法随机数至多被接受一次，因此先前已使用证明的重放会被确定性地拒绝。

对三个失败事件应用并合界，即可得到所述不等式。 □

7.6 无状态撤销

定义 7.5 （无状态撤销属性）。 我们称 CT-DAP 提供无状态撤销，如果对某个管理员因子$_i$执行 Revoke(i) 后：

（i）任何概率多项式时间对手都无法重构任何依赖于管理员因子$_i$的路径所对应的 $\mathsf{Cred}_{\mathrm{composite}}$（因此也无法重构 $\mathsf{REV}_{\mathrm{root}}$ 或 $\mathsf{Key}^{\mathcal{AP}_j}$）。（ii）除保管人执行销毁操作外，撤销不需要任何状态更新、广播或协调。（iii）不依赖于管理员因子$_i$ 的路径不受影响。

定理 7.7 （无状态撤销）。 在假设 *3.4* 下，销毁管理员因子$_i$ 可确保：

$$\mathit{ACE\text{-}GF}.\mathsf{Unseal}(\mathsf{params}, \mathsf{SA}_{\mathrm{root}}, \mathsf{Cred}_{\mathrm{partial}}) = \perp \quad \forall\, \mathsf{Cred}_{\mathrm{partial}} \neq \mathsf{Cred}_{\mathrm{composite}}$$

证明。 销毁管理员因子$_i$后，对手至多能够构造

$$\mathsf{Cred}' = \mathsf{Argon2id}(\mathsf{UserCred}\|\text{管理员因子}_1\|\cdots\|\text{管理员因子}'_i\|\cdots\|\text{管理员因子}_n;\ \ s;\ \ m,t,p)$$

其中 $\mathsf{AdminFactor}'_i \neq \mathsf{AdminFactor}_i$。

根据 AES-256-GCM-SIV 认证加密的正确性，使用错误密钥进行解密会得到 $\perp$（认证标签验证确定性地失败）。由于 $\mathsf{Cred}' \neq \mathsf{Cred}_{\mathrm{composite}}$，派生的解密密钥不同，因此 Unseal 返回 $\perp$。

该撤销机制是无状态的，因为它只需要在本地销毁 $\mathsf{AdminFactor}_i$ by$\mathcal{C}_i$——无需撤销列表、状态广播或重新密钥化。其他路径的独立性源于上下文隔离（定理 7.5）。

□

7.7 激活正确性

命题 7.8 （激活正确性）。 如果正确提供了所有凭证组件，则 激活 将以压倒性概率返回正确的控制密钥：

$$\Pr\big[\mathsf{Activate}(\Pi, \mathsf{SA}, \mathsf{UserCred}, \{\mathsf{AdminFactor}_i\}, \mathsf{Ctx}_j) \neq \mathsf{Key}_{\mathcal{AP}_j}\big] \ \leq\ p_{\mathrm{unseal}} + p_{\mathrm{derive}}$$

其中， p_{unseal} 和 p_{derive} 是底层原语的失败概率。

表2：安全属性和基本假设概述。

属性	陈述	假设
未授权控制	定理 7.4	带关联数据的认证加密（IND-CPA + INT-CTXT） + Argon2id 内存困难 + HKDF 伪随机函数（随机预言机模型）
跨路径隔离	定理 7.5	HKDF 伪随机函数（随机预言机模型） + 规范编码单射性
触发限制性	定理 7.6	存在性不可伪造性（选择消息攻击下）（验证者签名） + 抗碰撞性 H ((iv))
无状态撤销	定理 7.7	带关联数据的认证加密正确性（AES-GCM-SIV 标签验证）
激活正确性	定理 7.8	带关联数据的认证加密正确性 + HKDF 确定性
后量子就绪	第 7.8 小节	上下文编码敏捷性

证明。我们将失败事件分解为 $E = \{\text{激活} \neq \text{密钥}_{\mathcal{A}\quad\quad\quad\mathcal{P}_j}\}$ 成两个不相交的子-事件 s:E_1：**凭证正确但解封失败**。给定正确的 $\text{Cred}^{\text{composite}}$，Unseal 会使用 AES-256-GCM-SIV 解密 SA_{root}。根据 AEAD 方案的正确性，使用正确的密钥进行解密会确定性地成功（对于格式正确的密文，AES-GCM-SIV 是确定性 AEAD，不会发生解密错误 [3]）。因此 $p_{\text{unseal}} = 0$，前提是 SA_{root} 未在存储过程中损坏。如果不假设存储完整性，p_{unseal} 则受未检测到位翻转的概率限制（GCM-SIV认证标签以 $\leq 2^{-128}$ 的概率检测到位翻转）。

E_2：**REV 正确但派生失败**。派生是一个确定性函数（HKDF-SHA256）——给定 REV_{root} 和 Ctx$_j$，它总会生成相同的 $\text{Key}^{\mathcal{AP}_j}$。因此 $p_{\text{derive}} = 0$。

合并可得：在存储完好的情况下，$\Pr[E] \leq 0 + 0 = 0$；在存储位翻转模型下，$\Pr[E] \leq 2^{-128}$。□

7.8 后量子就绪性

继承自 ACE-GF 的上下文编码模型，CT-DAP 支持平滑迁移至后量子密码（PQC）域。采用后量子密码算法（例如 ML-DSA/Dilithium [28]）时，只需使用新的算法标识符更新上下文元组 Ctx，并从 REV_{root} 派生新的控制密钥，无需更改授权路径、密封工件或授权因子。

7.9 安全属性总结

表2总结了安全属性、其形式化陈述，以及各自所依赖的具体密码学假设。

8 与现有方法的比较

8.1 单密钥控制方法

传统系统将控制权等同于持有单个私钥 [5, 6]。这种模型虽然简单，但不支持条件激活、委托或撤销。CT-DAP 通过将控制权表示为休眠授权路径，将控制能力与永久持有秘密相分离。

表3：技术属性的结构比较。

属性	传统 BIP-39	CT-DAP（ACE-GF）
助记词语义	等同于明文主种子或私钥	根熵值的加密密文熵值
凭证修改	需要将资产迁移至新的密钥/地址	原地修改；REV保持不变
路径冗余	无（一对一映射在秘密与资产之间)	多对一（多条路径映射到单个 REV)
条件激活	不支持	通过休眠授权路径原生实现路径
撤销	需要密钥轮换或资产迁移	通过因子销毁立即实现（无状态）

8.2 多重签名与门限方案

多重签名和门限方案 [11, 12, 13] 分散了控制权，但要求持续密钥可用性，且无法表示条件性非活跃控制权。策略变更需要重新共享或资产迁移。CT-DAP 支持在激活前即存在的休眠路径，只有在条件满足后才会生效，且无需持续参与。

8.3 基于智能合约的控制

智能合约 [17]支持可编程控制，但依赖链上状态、预言机和特定的执行模型。CT-DAP 完全运行于密码学授权层，无需链上状态变更，也不存在合约逻辑依赖。

8.4 托管与可信中介模型

托管解决方案将控制权委托给可信中介。在 CT-DAP 中，托管人仅持有单独的授权因子，无法行使单方面控制——他们充当条件释放者，而非控制者。

8.5 结构比较

9 说明性场景

为展示 CT-DAP 的通用性，本节呈现跨越不同运营与法律上下文的场景。

9.1 场景 A：附条件继承

资产所有者建立多条与指定受益人相对应的休眠授权路径。每条路径包含一项用户持有的凭证，以及由独立保管人（例如遗产律师）持有的管理授权因子。

在所有者生前，所有与继承相关的路径在密码学上保持未激活状态。经核实一项法律认可的继承事件后，保管人释放相应的授权因子。此举激活该路径，使受益人能够行使密码学控制。

不会发生资产迁移或重新分配；控制权仅通过路径激活来实现。

9.2 场景 B：受监管或监督式控制

某组织运营受监管监督的资产。控制权需要同时具备内部凭证和由监督实体持有的授权因子。

在正常情况下，监督因子始终可用。若发生监管干预，则销毁该监督因子，从而立即禁用控制，而不修改资产所有权或密码学身份。

9.3 场景 C：委托式限时控制

资产所有者将控制权委托给第三方，期限有限。一个休眠授权a- tion路径通过时间条件授权因子建立。

只有在时间条件满足时，该因子才会被释放。到期后，该因子将被销毁，使该路径永久失效——无需持续信任或事后撤销程序。

9.4 场景 D：多方治理

去中心化自治组织或企业财库要求对资产操作进行 m-of-n 审批。 n 名治理成员中的每一位都持有一个授权因子。只有当 m 个因子同时被释放时，该路径才会被激活，其余 $n-m$ 个因子则继续保持锁定。

不同于传统多重签名，各治理路径都可以通过销毁单个因子而独立撤销，无需重新共享或进行链上重新配置。

10 讨论

10.1 实际部署注意事项

10.1.1 托管方信任与激励协调

系统依赖托管人忠实保管并有条件地释放管理员因子$_i$。实际部署应包括：

- 通过法律协议约束托管人遵守预先设定的释放条件。
- 对管理员因子$_i$ 进行多托管方分片（例如，使用门限密码学实现因子释放），以降低单一托管方故障的影响。
- 对因子释放/销毁操作进行可审计日志记录；这得益于 ACE-GF 与可信执行环境（TEEs） [9]的兼容性。

10.1.2 条件验证机制

该方法的有效性取决于能否准确验证外部触发条件。对于具有法律约束力的条件，TTP 验证方（例如公证人、监管机构）会提供经密码学签名的证明。对于透明且防篡改的条件，去中心化预言机网络[8] 可以取代 TTP。

10.1.3 与现有资产系统的互操作性

由于 ACE-GF 采用上下文隔离派生，CT-DAP 不受底层资产类型的限制：

- **基于区块链的资产：** Key$_{AP}$ 可通过 ACE-GF 的传统密钥嵌入机制映射到标准账户地址（例如，比特币/以太坊使用的 Secp256k1）。
- **中心化数字资产：** Key$_{AP}$ 可用作 API 认证令牌，资产提供方则通过签名验证实施密码学控制。

10.1.4 旧版迁移与地址保留

系统支持“锚定旧版”模式，可从现有的 BIP-39 助记词导入熵，以生成锚定旧版的 REV_{root}。该方法采用标准的 BIP-32/SLIP-10 派生路径，确保地址与原钱包确定性地保持一致，使用户无需进行链上迁移即可将现有资产纳入 CT-DAP 框架。

10.2 通过并行授权路径实现容错

可销毁因子的一项关键隐患是意外丢失可能导致资产永久锁定。CT-DAP 通过支持并行、冗余授权路径来缓解这一问题。

形式上，资产所有者可以建立 m 条彼此独立的授权路径：

$$S_{\mathcal{AP}} = \{\mathcal{AP}_1, \mathcal{AP}_2, \ldots, \mathcal{AP}_m\}$$

其中，每条授权路径j 均由唯一的凭证组合 $\text{Cred}_{\text{composite},\ j}$ 和上下文 Ctx$j$定义。这种架构具备以下特性：

1. **冗余恢复：**一条用于日常操作的主路径授权路径$_{\text{primary}}$ ，以及紧急恢复路径授权路径$_{\text{recovery}}$。销毁授权路径$_{\text{primary}}$ 中的某个因子不会影响授权路径$_{\text{recovery}}$。
2. **密码学隔离：**根据定理 7.5，攻破一条路径不会泄露其他路径的信息。
3. **基于策略的可用性：**不同路径可以要求不同的安全级别（例如，“继承路径”所需的因子数量可以多于“用户同意路径”）。

10.3 局限性

10.3.1 凭证熵依赖性

$\text{Cred}_{\text{composite}}$ 的安全性继承了 ACE-GF 对凭证熵的依赖。低熵的 UserCred 或 Ad minFactor$_i$ 会增加其遭受离线暴力破解攻击的风险。部署时必须强制执行最低熵要求（例如，UserCred 的最小熵为 128 位），并利用 Argon2id 的内存困难特性。

10.3.2 侧信道攻击漏洞

虽然 ACE-GF 确保 REV_{root} 在内存中是临时的，但在不可信环境中，解封和派生操作仍容易受到侧信道攻击（例如冷启动攻击、时序分析）。可通过在可信执行环境（例如 Intel SGX、ARM TrustZone）中执行关键操作来缓解此问题 [9]。

10.3.3 不可逆撤销风险

通过销毁管理员因子$_i$ 进行的撤销是永久性的。实际系统应为管理员因子$_i$ 实施安全备份机制（例如加密分片），并按照第 10.2 小节所述设置恢复路径。

10.3.4 条件验证器信任

该方法的安全性取决于条件验证器的完整性（假设 3.3）。遭到入侵的验证者可能触发因子提前释放。多验证器证明和阈值验证方案可以降低这一风险，但会增加操作复杂性。

表 4：核心 CT-DAP 操作的延迟（Apple M2，单核）。

操作	Mean	标准差
设置（密钥生成）	0.02 ms	<0.01 ms
Argon2id（$m=256$ MiB，$t=3$，$p=4$）	482 毫秒	11 毫秒
封装（AES-256-GCM-SIV，32 B 有效载荷）	0.003 毫秒	<0.001 毫秒
解封（AES-256-GCM-SIV，32 B 有效载荷）	0.003 毫秒	<0.001 毫秒
派生（HKDF-SHA256，一个上下文）	0.005 毫秒	<0.001 毫秒
路径设置（端到端）	483 毫秒	11 毫秒
激活（端到端）	483 毫秒	11 毫秒

10.4 未来研究方向

1. **去中心化条件验证：** 探索基于去中心化自治组织的条件验证，采用由社区治理的智能合约，并使用零知识证明来实现隐私保护。

2. **后量子优化：**针对后量子密码算法（例如 ML-DSA、ML-KEM）优化派生效率，包括定制HKDF参数，并在量子威胁下验证上下文隔离。

3. **跨链与多资产控制：**在各区块链之间标准化上下文元组，并通过跨链消息协议（例如 IBC）实现可互操作的管理员因子$_i$ 释放。

4. **动态授权路径：**支持对管理员因子$_i$ 集合进行动态修改（例如添加/移除托管人），而无需重新派生REV$_{\text{root}}$ 或迁移资产。

5. **形式化验证：**应用自动化验证工具（例如 ProVerif、Tamarin），针对更广泛的一类对抗行为对协议进行形式化验证。

11 性能评估

我们通过基准测试一个基于标准密码学库构建的原型实现，评估 CT-DAP 核心操作的计算成本。

11.1 实验设置

该实现使用以下具体原语:

- **封装/解封：**通过 `libsodium` [3] 使用 AES-256-GCM-SIV。
- **密钥派生：**通过 OpenSSL [4] 使用 HKDF-SHA256。
- **凭证组合：**通过参考实现 [2, 16] 使用 Argon2id。

所有测量均在运行 macOS 14 的 Apple M2 处理器（3.49 GHz）单核上进行，并在缓存预热后对 1 000 次迭代的结果取平均值。Argon2id参数设置为$m = 256$MiB、 $t = 3$ 迭代次数、$p = 4$ 通道（遵循 OWASP 对凭证哈希的建议）。

11.2 结果

表 4 报告了每项操作的延迟。

表 5：Argon2id 延迟与内存参数的关系（$t=3$、 $p=4$，Apple M2）。

m (MiB)	延迟（ms）	内存成本	使用场景
64	118	Low	移动设备 / 物联网设备
128	239	中等	桌面钱包
256	482	High	托管基础设施
512	971	非常高	冷存储

11.3 分析

在路径设置和激活中，占主导地位的成本都是 Argon2id 凭证组合步骤，占总延迟的 $>$99.9%。其余操作——AES-256-GCM-SIV 加密/解密和 HKDF 派生——均在微秒级完成，可以忽略不计。

安全性与性能的权衡。 表 5 展示了 Argon2id 的内存参数 m 如何控制抗暴力破解能力与激活延迟之间的权衡。

所有配置在 $m = 256$MiB 或更低时，激活延迟均低于一秒。即使是最保守的设置（$m = 512$MiB），延迟也不到一秒，这证实了 CT-DAP 适用于交互式使用。无论路径数量多少，路径派生步骤（派生）带来的开销都是固定的，因为每次激活只会重建一个路径密钥。

12 结论

本文介绍了条件触发休眠授权路径（CT-DAP），一种基于可销毁授权因子的密码学资产控制方法。CT-DAP 将控制权建模为休眠授权路径，这些路径通过激活而非转移来启用，从而将资产身份与条件依赖控制解耦。

我们明确给出了信任假设和算法规范，对系统模型进行了形式化，并在标准密码学假设下证明了三项核心安全属性——未授权控制抵抗性、跨路径隔离和无状态撤销。我们使用 ACE-GF 框架实现了该方法，并展示了其在受监管托管、继承规划、委托控制和多方治理中的适用性。

更广泛地说，本研究表明，加密资产控制不必局限于静态密钥持有模型。休眠授权路径的条件触发式激活提供了一种通用抽象，用于在多样化的操作与监管环境中表达合法合规且密码学强制执行的控制语义。

参考文献

[1] Wang, J. S. (2025). ACE-GF：原子密码实体生成框架。*arXiv* 预印本, arXiv:2511.20505。`https://arxiv.org/abs/2511.20505`.[2] Biryukov, A., Dinu, D., and Khovratovich, D. (2016). Argon2：用于密码哈希及其他应用的新一代内存困难函数。*IEEE Euro S&P 2016*。[3] Gueron, S., Langley, A., and Lindell, Y. (2019). RFC 8452：AES-GCM-SIV：防随机数误用的认证加密。IETF。[4] Krawczyk, H., and Eronen, P. (2010). RFC 5869：基于HMAC的提取与扩展密钥派生函数（HKDF）。IETF。

[5] Wuille, P. (2012). BIP-32：分层确定性钱包。比特币改进提案。[6] Palatinus, M., Rusnak, P., Voisine, A., and Bowe, S. (2013). BIP-39：用于生成确定性密钥的助记码。比特币改进提案。[7] Palatinus, M., and Rusnak, P. (2014). BIP-44：确定性钱包的多账户层级结构。比特币改进提案。[8] Adler, J., Berryhill, R., Veneris, A., Poulos, Z., Veira, N., and Kastania, A. (2018). Astraea：去中心化区块链预言机。*IEEE iThings/GreenCom/CPSCom/SmartData 2018*，1145–1152。[9] Costan, V., and Devadas, S. (2016). Intel SGX 解析。*IACR* 密码学电子印本档案，2016(866)。[10] Shamir, A. (1979). 如何共享一个秘密。《*ACM* 通讯》，22(11)，612–613。[11] Gennaro, R., and Goldfeder, S. (2018). 快速信任设置下的快速多方门限 ECDSA。*ACM CCS 2018*，1179–1194。[12] Gennaro, R., and Goldfeder, S. (2020). 具有可识别中止的单轮门限 ECDSA。*IACR* 密码学电子印本档案，2020(540)。[13] Komlo, C., and Goldberg, I. (2020). FROST：灵活的轮次优化 Schnorr 门限签名。*SAC 2020*，《计算机科学讲义》12804，34–65。[14] Gennaro, R., Jarecki, S., Krawczyk, H., and Rabin, T. (1996). 稳健的门限 DSS 签名。*EUROCRYPT 1996*，《计算机科学讲义》1070，354–371。[15] Herzberg, A., Jarecki, S., Krawczyk, H., and Rabin, T. (1995). 主动秘密共享，或：如何应对持续泄漏。*CRYPTO 1995*，《计算机科学讲义》963，339–352。[16] Biryukov, A., Dinu, D., and Khovratovich, D. (2021). RFC 9106：用于密码哈希和工作量证明应用的 Argon2 内存困难函数。IETF。[17]Wood, G. (2014). 以太坊：一种安全的去中心化通用交易账本。以太坊项目黄皮书，151，1–32。[18] Szabo, N. (1997). 将公共网络上的关系形式化并加以保护。《*First Monday*》，2(9)。[19] Bethencourt, J., Sahai, A., and Waters, B. (2007). 密文策略属性基加密。*IEEE* 安全与隐私研讨会 *2007*，321–334。[20] Goyal, V., Pandey, O., Sahai, A., and Waters, B. (2006). 用于加密数据细粒度访问控制的属性基加密。*ACM CCS 2006*，89–98。[21] Sahai, A., Seyalioglu, H., and Waters, B. (2012). 属性基加密的动态凭证与密文委托。*CRYPTO 2012*，《计算机科学讲义》7417，199–217。[22] Rivest, R. L., Shamir, A., and Wagner, D. A. (1996). 时间锁谜题与定时发布密码学。*MIT/LCS/TR-684*。[23]Boneh, D., Bonneau, J., Bünz, B., and Fisch, B. (2018). 可验证延迟函数。*CRYPTO 2018*，《计算机科学讲义》10991，757–788。[24] Wöhrer, M., and Zdun, U. (2021). 基于智能合约的社交恢复钱包设计模式。*IEEE DAPPS 2021*，69–78。

[25] He, D., Deng, Z., Zhang, Y., Chan, S., Cheng, Y., 和 Guizani, N.（2020）。智能合约漏洞分析与安全审计。*IEEE Network*，34(5)，276–282。[26] Goldfeder, S., Gennaro, R., Kalodner, H., Bonneau, J., Kroll, J. A., Felten, E. W., 和 Narayanan, A.（2015）。通过一种新型 DSA/ECDSA门限签名方案保护比特币钱包。手稿，
`https://www.cs.princeton.edu/~stevenag/threshold_sigs.pdf`。[27] Lindell, Y.（2021）。采用实用分布式密钥生成的快速安全多方ECDSA及其在加密货币托管中的应用。*ACM CCS 2018*（2021 年更新）。[28] Ducas, L., 等（2022）。CRYSTALS-Dilithium：算法规范及支持文档（v3.1）。*NIST*后量子密码标准化。[29] Alwen, J., 和 Blocki, J.（2017）。迈向对 Argon2i 和气球哈希的实用攻击。*IEEE Euro S&P 2017*，142–157。