CashWeb：面向联邦式反垃圾消息与发布-订阅系统的加密货币集成协议

Shammah Chancellor
shammah.chancellor@proton.me
https://t.me/TheLotusNetwork

2026 年 2 月 21 日
Version 1.1

摘要

本文介绍 CashWeb——一套综合性协议套件，将基于加密货币的经济机制与联邦式消息基础设施相结合，在无需中心化内容审核的前提下实现抗垃圾消息通信。该协议采用一种新颖的"燃烧发言"（burn-to-speak）反垃圾机制：消息发送方附带少量加密货币支付，该支付可通过密码学方式验证，且无需可信中间方。我们为三个核心组件提供正式规范：（1）将链上价值与链下行为相关联的支付证明（POP）协议，（2）具有密码学身份管理的联邦式消息传递，以及（3）面向基于主题广播的发布-订阅系统。经济分析表明，燃烧机制能够对垃圾消息实现最优威慑，同时保持合法通信的可及性。系统架构充分利用已成熟的网络标准（HTTP/2、WebSockets）和密码学原语，以实现与现有基础设施的无缝集成。理论分析表明，该协议以随网络规模次线性增长的代价实现抗垃圾能力；实证部署则验证了其在实时消息应用中的实际可行性。

1 引言

1.1 中心化问题

互联网通信协议的演进揭示了去中心化与可用性之间一个根本性的张力。早期系统如 Usenet Kohn et al. [2009], Lindsey and Allbery [2009]、SMTP Resnick [2008], Klensin [2008] 以及 XMPP Saint-Andre [2004a,b] 均被设计为联邦式网络，无需中央权威机构即可实现点对点通信。然而，这些协议内在的非对称成本结构——消息处理成本由接收方承担，而发送成本几乎可忽略不计——为滥用行为创造了经济激励，最终驱使用户转向中心化平台。

到 2020 年，通信基础设施的集中程度已达到前所未有的水平：谷歌、苹果和微软合计控制了 85% 的电子邮件客户端市场份额 Labs [2020]，Facebook 报告拥有超过 20 亿用户，Gmail 服务超过 15 亿活跃账户 Gmail [2018]。这种中心化虽然为用户提供了便利，却也引入了系统性风险，包括审查、监控以及单点故障，从而损害了分布式互联网通信的原始愿景。

1.2 经济性反垃圾机制

去中心化消息系统面临的根本挑战是在没有中心化过滤的情况下防范垃圾消息。传统方法依赖计算成本（Hashcash Back [2002]）或需要持久身份验证的声誉系统。尽管这些机制提供了一定程度的保护，但它们存在明显局限：基于计算的工作量证明在现代硬件加速下扩展性较差；而声誉系统为新用户设置了准入门槛，同时仍易受女巫攻击。

加密货币网络的出现，尤其是比特币 Nakamoto [2008]，开辟了一种新途径：基于经济的支付证明机制——通过密码学证明的价值销毁而非计算工作来验证消息真实性。这一思路最初由 Finney Finney [2004] 以" 可重复使用的工作量证明"（Reusable Proof of Work）的形式提出，如今已可借助去中心化加密货币网络无需可信中间方地实现。

1.3 主要贡献

本文介绍 CashWeb，这是一套通过三项核心贡献解决中心化问题的综合性协议套件：

1. **正式的支付证明协议**：我们规范了一套完整的密码学协议，用于将链上加密货币交易与链下消息行为相关联，在无需可信方的前提下实现可验证的反垃圾机制。
2. **联邦式基础设施设计**：我们提出了一种可扩展架构，将用于身份管理的密钥服务器与用于消息路由的中继服务器相结合，旨在支持广泛采用所需的交易量，同时保持去中心化属性。
3. **经济安全分析**：我们提供理论分析，证明适当的销毁率机制能够在垃圾消息威慑与合法用户可及性之间实现最优权衡，并给出攻击成本和成功概率的形式化界。

该协议面向实际部署而设计，充分利用成熟的网络标准，并保持与现有互联网基础设施的兼容性。Stamp 社交网络的实施经验验证了该方法的现实可行性。

本文是系列论文的第三篇。Chancellor [2026b] 奠定了理论基础：在任何无许可共识系统中，均衡安全支出以风险敞口价值为锚，工作量证明（PoW）通过对抗性揭示的工作信号，使该支出具有协议可读性。Chancellor [2026a] 推导出货币政策含义，规范了一种无预言机的双环机制：以 PoW 工作信号驱动自适应尾部排放，以燃烧发言作为内生供给汇。本文将燃烧发言机制完整规范为一套联邦式消息协议。因此，燃烧发言在整个技术栈中身兼二职：在应用层实现垃圾消息威慑，在协议层实现内生货币供给管理。

1.4 论文结构

第 2 节回顾反垃圾机制和联邦式消息领域的相关工作。第 3 节建立系统模型和威胁假设。第 4 节提供核心协议的正式规范。第 5 节分析经济属性和抗垃圾保证。第 6 节考察安全属性和抗攻击能力。第 7 节讨论实际部署注意事项。第 9 节给出理论与实证评估结果。

2 背景与相关工作

2.1 经济性反垃圾机制

在分布式系统研究中，利用经济激励防范垃圾消息有着丰富的历史。Dwork 和 Naor Dwork and Naor [1992] 最早提出要求在发送电子邮件时附带计算工作量证明的方案，该方案通过 SHA-256 哈希原像难题在 Hashcash Back [2002] 中得以实现。然而，计算型方案在实践中存在若干局限：（1）计算成本因硬件差异而大幅变化，导致不公平；（2）现代 ASIC 和 GPU 加速能够以比消费级硬件快若干数量级的速度求解谜题；（3）最优难度需要随计算能力的演进持续调整。

Laurie 和 Clayton Laurie and Clayton [2004] 提出使用内存密集函数来削弱硬件优势，但这些方法仍需消耗大量能量，并对资源受限设备造成障碍。基于声誉的替代方案 Golbeck and Hendler [2005] 需要持久身份系统，与隐私目标相冲突，并为合法新用户设置了较高的准入门槛。

2.2 基于加密货币的访问控制

可编程加密货币网络的出现使更复杂的经济机制成为可能。Miller 等人 Miller and LaViola Jr [2014] 提出利用比特币进行匿名微支付，但其方案需要交互式协议，无法扩展到消息应用场景。支付通道 Poon and Dryja [2016] 和状态通道 Dziembowski et al. [2018] 的最新研究提供了低延迟微支付机制，但预资金通道的要求带来了可用性障碍。

我们的方法与上述工作的不同之处在于：采用不可恢复的价值销毁（”燃烧”）而非价值转移，从而无需接收方支付基础设施，同时维持强大的反滥用经济激励。

2.3 联邦式消息系统

联邦式消息架构在去中心化优势与实际可扩展性需求之间寻求平衡。XMPP Saint-Andre [2004a] 证明了联邦式实时消息的可行性，但缺乏经济性反垃圾机制。Matrix Foundation [2019] 提供了具备端到端加密的现代联邦式消息，但依赖中心化身份提供商，且缺乏速率限制以外的反垃圾机制。

近期基于区块链的消息系统包括 Status Network [2017] 和 Session Foundation [2020]，它们提供去中心化身份管理，但未能解决驱动中心化的经济激励问题。我们的方案将经济机制直接整合进协议设计，以解决这些根本性的激励错位问题。

3 系统模型与架构

3.1 网络参与者

CashWeb 网络由四类参与者组成：

定义 1 (网络参与者)**.** *设 $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ 为用户集合，$\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ 为中继服务器集合，$\mathcal{K} = \{k_1, k_2, \ldots, k_\ell\}$ 为密钥服务器集合，$\mathcal{T} = \{t_1, t_2, \ldots, t_p\}$ 为发布-订阅主题集合。*

用户 ($u_i \in \mathcal{U}$)：发送和接收消息的终端参与者。每位用户控制一个加密货币钱包，并能够生成密码学支付证明。用户通过客户端软件与网络交互，该软件负责管理密钥材料和支付交易。

中继服务器（$r_i \in \mathcal{R}$）：代表用户存储和转发加密消息的联邦式服务器。中继服务器在接受消息前验证支付证明，提供第一道反垃圾防线。由于采用端到端加密，中继服务器无法访问消息内容。

密钥服务器（$k_i \in \mathcal{K}$）：维护用户公钥及关联元数据全局注册表的分布式服务器。密钥服务器使用户能够在无需事先联系的情况下发现中继服务器并建立安全通信信道。

主题（$t_i \in \mathcal{T}$）：发布-订阅系统中用于广播通信的具名频道。主题是去中心化的——任何用户均可创建主题，任何中继服务器均可根据订阅者列表传播主题消息。

3.2 威胁模型

我们考虑拜占庭对抗环境，其中参与者可能任意偏离协议规范。我们的威胁模型包括：

垃圾消息攻击者：试图发送大量未经请求的消息以破坏通信或浪费网络资源的攻击者。我们假设攻击者拥有相当但有限的计算和财务资源。

女巫攻击者：创建大量虚假身份以扩大消息发送能力或操纵声誉系统的攻击者。

网络基础设施攻击者：控制中继服务器或密钥服务器，可能尝试审查、流量分析或拒绝服务攻击的攻击者。

经济攻击者：试图操纵经济机制的攻击者，包括通过挖矿或验证者控制权回收已销毁资金的手续费回收攻击。

我们假设底层加密货币网络（如比特币、Lotus）提供标准安全保证，包括交易不可篡改性和抗双重支付。我们不考虑针对加密货币网络本身的攻击。

3.3 设计原则

协议设计遵循以下几项关键原则：

经济安全性：反垃圾保护应源于经济激励而非计算障碍或中心化内容审核，确保攻击成本高昂而合法使用保持低廉。

隐私优先设计：消息内容和通信模式应对第三方保持私密，包括网络基础设施提供商。

联邦式可扩展性：系统应通过联邦架构支持全球规模的采用，同时保持去中心化属性。

向后兼容性：协议应与现有互联网基础设施和开发框架集成，以尽量降低采用门槛。

4 核心协议规范

4.1 支付证明（POP）协议

支付证明协议使密码学验证成为可能：证明特定数量的加密货币已被销毁（永久销毁）且与某一特定链下行为相关联。与将价值在各方之间转移的传统支付系统不同，POP 协议创建可验证的价值销毁证明，该证明无法被恢复或双重支付。

定义 2（支付证明协议）. *POP 协议是一个三元组* $(\textsc{Setup}, \textsc{Commit}, \textsc{Verify})$，*其中*：

- $\textsc{Setup}(1^\lambda) \to pp$：*针对安全参数* λ *生成公共参数。*
- $\textsc{Commit}(sk, m, v) \to (\pi, \tau)$：*给定私钥* sk、*消息* m *和价值* v，*输出证明* π *和交易* τ。
- $\textsc{Verify}(\pi, \tau, m, v) \to \{0, 1\}$：*验证交易* τ *销毁了价值* v *且与消息* m *存在密码学关联。*

4.1.1 不可花费输出的构造

在给出算法之前，我们首先定义核心原语：

定义 3 (不可花费地址 / OP_RETURN 输出)**.** *给定 32 字节承诺值 h_m，$\textsc{UnspendableAddress}(h_m)$ 表示输出脚本 OP_RETURN $\langle h_m \rangle$。发送至该脚本的金额为 v 的交易输出可被证明为永久销毁：(i) OP_RETURN 输出在共识上作为花费输入无效，因此任何一方均无法构造有效的花费交易；(ii) 合规节点将此类输出标记为 UTXO 豁免，永久从流通中移除该价值。32 字节的 SHA-256 哈希载荷在 80 字节的 OP_RETURN 数据限制内，留有 4 字节用于协议标识符和版本字段。*

本文所针对的部署目标 Lotus 网络通过共识规则支持具有非零价值的 `OP_RETURN` 输出，因此上述方式为主要构造。对于禁止 `OP_RETURN` 输出具有非零价值的链（标准比特币策略），备选构造是将 v 发送至地址 P2PKH($\textsc{Hash160}(\texttt{0x0000...00} \| h_m)$)：一个语法上有效但对应私钥不可访问的 P2PKH 地址，从而实现概率意义上的不可花费性，而非共识层面强制执行的不可花费性。

4.1.2 构造

我们的 POP 协议构造利用比特币式交易中的 OP_RETURN 输出来嵌入承诺数据，同时确保资金销毁：

Algorithm 1 支付证明构造

Require: 消息 m，销毁金额 v，用户私钥 sk
Ensure: 证明 π 和交易 τ

$pk \leftarrow \textsc{DerivePublic}(sk)$
$h_m \leftarrow \textsc{Hash}(m \| pk \| \textsc{Timestamp}())$
$addr_{burn} \leftarrow \textsc{UnspendableAddress}(h_m)$
$\tau \leftarrow \textsc{CreateTransaction}(sk, v, addr_{burn}, h_m)$
$\sigma \leftarrow \textsc{Sign}(sk, \tau \| m)$
$\pi \leftarrow (\tau, \sigma, m, v, pk)$
return π, τ

该构造确保若干关键属性：

不可恢复性：发送至 $addr_{burn}$ 的资金可被证明无法花费，因为该地址由一个不存在已知原像的哈希值派生而来。

唯一性：每个消息-支付对生成唯一的承诺，无法被重用于不同消息。

可验证性：任何一方均可通过在区块链上核查交易并验证密码学签名来验证该证明。

4.2 身份管理与密钥注册

CashWeb 中的用户基于公钥密码学维护假名身份。每个身份由密钥对 (sk, pk) 组成，公钥作为用户的全局标识符。

4.2.1 身份注册

新身份通过以下协议向密钥服务器网络注册：

Algorithm 2 身份注册

Require: 用户密钥对 (sk, pk)，中继服务器地址 $addr_{relay}$，注册费 v_{reg}
Ensure: 身份记录已录入密钥服务器网络

$metadata \leftarrow \{\text{“}relay\text{”} : addr_{relay}, \text{“}timestamp\text{”} : \textsc{Now}()\}$
$(\pi_{reg}, \tau_{reg}) \leftarrow \text{POP.}\textsc{Commit}(sk, metadata, v_{reg})$
$record \leftarrow (pk, metadata, \pi_{reg})$
for each keyserver $k_i \in \mathcal{K}$ **do**
    $\textsc{Send}(record, k_i)$
end for

4.2.2 密钥轮换与恢复

该协议支持密钥轮换，以便在密钥被盗或丢失时进行恢复。用户可预先注册恢复密钥，或使用分层确定性密钥派生实现无缝轮换：

Algorithm 3 密钥轮换

Require: 当前密钥对 (sk_{old}, pk_{old})，新密钥对 (sk_{new}, pk_{new})，轮换费 v_{rot}
Ensure: 身份记录已更新

$rotation\_msg \leftarrow (pk_{old}, pk_{new}, \textsc{Now}())$
$\sigma_{old} \leftarrow \textsc{Sign}(sk_{old}, rotation\_msg)$
$\sigma_{new} \leftarrow \textsc{Sign}(sk_{new}, rotation\_msg)$
$(\pi_{rot}, \tau_{rot}) \leftarrow \text{POP.}\textsc{Commit}(sk_{new}, rotation\_msg, v_{rot})$
$update \leftarrow (pk_{old}, pk_{new}, \sigma_{old}, \sigma_{new}, \pi_{rot})$
for each keyserver $k_i \in \mathcal{K}$ **do**
    $\textsc{Send}(update, k_i)$
end for

4.3 燃烧发送的联邦式消息传递

核心消息协议将密码学消息投递与经济性反垃圾保护相结合。消息经端到端加密，并附带支付证明以表明发送方的承诺。

Algorithm 4 消息发送协议

Require: 接收方公钥 pk_{recv}，消息内容 $content$，销毁金额 v_{msg}
Ensure: 消息已投递至接收方的中继服务器

$(pk_{sender}, sk_{sender}) \leftarrow \textsc{GetUserKeys}()$
$relay_{recv} \leftarrow \textsc{KeyserverLookup}(pk_{recv})$
$k_{shared} \leftarrow \text{ECDH}(sk_{sender}, pk_{recv})$
$msg_{encrypted} \leftarrow \textsc{Encrypt}(k_{shared}, content)$
$message \leftarrow (pk_{sender}, pk_{recv}, msg_{encrypted}, \textsc{Timestamp}())$
$(\pi_{msg}, \tau_{msg}) \leftarrow \text{POP.}\textsc{Commit}(sk_{sender}, message, v_{msg})$
$delivery\_req \leftarrow (message, \pi_{msg})$
$\text{HTTP}\textsc{Post}(delivery\_req, relay_{recv})$

4.3.1 消息验证与存储

中继服务器在存储前验证传入消息：

Algorithm 5 中继服务器的消息验证

Require: 消息投递请求 $delivery\_req = (message, \pi_{msg})$
Ensure: 消息被接受或拒绝

$(message, \pi_{msg}) \leftarrow delivery\_req$
$(pk_{sender}, pk_{recv}, msg_{encrypted}, timestamp) \leftarrow message$
if POP.Verify$(\pi_{msg}, message) \neq 1$ **then**
  return “已拒绝：支付证明无效”
end if
if BurnAmount$(\pi_{msg}) < v_{min}$ **then**
  return “已拒绝：销毁金额不足”
end if
StoreMessage$(message, \pi_{msg})$
return “已接受”

4.4 燃烧广播的发布-订阅系统

发布-订阅系统将消息协议扩展至支持基于主题的广播。用户可订阅主题并接收发布到该主题的所有消息，消息优先级由销毁金额决定。

定义 4 (主题订阅)**.** *主题订阅是一个三元组* (u_i, t_j, r_k)*，表示用户* u_i *通过中继服务器* r_k *订阅主题* t_j*。*

Algorithm 6 主题消息广播

Require: 主题标识符 $topic\_id$，消息内容 $content$，销毁金额 $v_{broadcast}$
Ensure: 消息已分发至所有主题订阅者

$(pk_{sender}, sk_{sender}) \leftarrow$ GetUserKeys()
$topic\_msg \leftarrow (pk_{sender}, topic\_id, content,$ Timestamp())
$(\pi_{broadcast}, \tau_{broadcast}) \leftarrow$ POP.Commit$(sk_{sender}, topic_msg, v_{broadcast})$
$broadcast\_req \leftarrow (topic\_msg, \pi_{broadcast})$
for each relay server $r_i \in \mathcal{R}$ **do**
  SendToRelayIfSubscribers$(broadcast_req, topic_id, r_i)$
end for

广播机制包含基于销毁金额的优先级排序和速率限制：

5 经济分析与反垃圾属性

5.1 销毁率经济学与均衡分析

燃烧发言机制的有效性取决于制定能使垃圾攻击在经济上不可行，同时为合法用户保持可及性的销毁率。我们将其建模为博弈论均衡问题，遵循 Becker [Becker, 1968] 的最优威慑框架，并将其扩展至去中心化场景——在该场景中，执行通过密码学证明的价值销毁来实现。

Algorithm 7 主题消息优先级排序

Require: 主题消息集合 $M = \{m_1, m_2, \ldots, m_k\}$ 及对应销毁金额 $\{v_1, v_2, \ldots, v_k\}$
Ensure: 已排定优先级的消息投递计划

```
priority_queue ← EmptyQueue()
for each message m_i ∈ M do
    priority_i ← f(v_i)，其中 f 为单调递增函数
    Insert(priority_queue, m_i, priority_i)
end for
while NotEmpty(priority_queue) and BandwidthAvailable() do
    m_next ← PopMax(priority_queue)
    DeliverMessage(m_next)
end while
```

定义 5 (垃圾攻击成本)**.** 对于试图以每条消息销毁金额 B 发送 N 条垃圾消息的攻击者，总攻击成本为：

$$C_{attack}(N, B) = N \cdot B + C_{operational}(N)$$

其中 $C_{operational}(N)$ 表示计算和基础设施成本。

定义 6 (合法用户效用)**.** 合法用户以销毁金额 B 发送一条消息的效用为：

$$U_{legit}(B) = V_{communication} - B - C_{friction}(B)$$

其中 $V_{communication}$ 是成功消息投递带来的价值，$C_{friction}(B)$ 表示可用性成本。

最优销毁率 B^* 在最大化合法用户采用率的同时最小化垃圾消息的可行性。为获得易处理的形式，我们指定具体函数形式：

定义 7 (社会福利函数)**.** 定义：

$$W(B) = U_0 - \alpha B - \frac{D\sigma}{B},$$

其中 $U_0 > 0$ 为代表性合法用户的每条消息价值，$\alpha > 0$ 为其销毁的边际成本，$D > 0$ 为每条垃圾消息的社会损害，$\sigma > 0$ 为单位销毁成本时的垃圾消息量。αB 项捕捉较高销毁成本对合法用户剩余的减少；$D\sigma/B$ 项捕捉总垃圾损害——在理性垃圾发送者经济学（垃圾发送者将边际垃圾收益等于边际销毁成本）下，这以 $1/B$ 的速率下降。

在此函数形式下，最优销毁率为：

$$B^* = \sqrt{\frac{D\sigma}{\alpha}},$$

以闭合形式平衡了威慑与可及性。较高的社会垃圾损害 D 或较高的垃圾倾向 σ 意味着更高的 B^*；较低的合法用户边际成本 α 也会提高 B^*。

定理 8 (最优销毁率)**.** 最优销毁率 B^* 满足：

$$\frac{\partial}{\partial B}\left[\sum_{i=1}^{n} U_{legit}^{i}(B) - \alpha \cdot E[N_{spam}(B)]\right] = 0$$

其中 α 表示垃圾消息的社会成本，$E[N_{spam}(B)]$ 是销毁率为 B 时预期的垃圾消息数量。

*证明. 存在性。*当 $B \to 0$ 时，垃圾消息不受约束，$W(B) \to -\infty$。当 $B \to \infty$ 时，合法用户因成本过高而退出，$W(B) \to -\infty$。由于 W 是连续函数，根据极值定理，在包含最大值的任意紧子区间上存在内部最大值。

*一阶条件。*在内部最优点 B^* 处，所述条件由求导直接得出。

*唯一性*在额外假设 $W(B)$ 严格凹的条件下成立，当合法效用关于 B 是凹的且垃圾消息量关于 $1/B$ 是凸的时，该假设成立——这是最优威慑文献中的标准假设 [Becker, 1968]。 □

5.2 女巫抵抗量化

燃烧机制提供固有的女巫抵抗能力，因为每个身份需要经济承诺而非仅仅计算工作：

命题 9 (女巫攻击界)**.** *对于预算为 $\mathcal{B}$ 的攻击者，女巫身份的最大数量受以下约束：*

$$N_{sybil} \leq \frac{\mathcal{B}}{v_{reg} + k \cdot v_{msg}}$$

其中 v_{reg} 为身份注册费，k 为每个身份预期的消息数量。

该界表明，女巫攻击的规模随攻击者预算线性增长，而非随计算资源增长，从而提供可预测的抵抗保证。

5.3 手续费回收攻击防护

一个关键的安全考量是防止攻击者通过控制挖矿或验证基础设施回收已销毁的资金。我们通过部分手续费燃烧来解决这一问题：

定义 10 (部分手续费燃烧机制)**.** *对于手续费为 F 的每笔交易，比例 $\beta \in (0, 1]$ 被销毁，其余 $(1-\beta)F$ 支付给矿工：*

$$B_{total} = B_{explicit} + \beta \cdot F$$

其中 $B_{explicit}$ 为显式销毁金额，F 为交易手续费。

参数 β 与 Chancellor [2026a] 中的自适应货币政策框架共享，在该框架中，相同的手续费燃烧比例同时充当内生供给通缩机制。因此，反垃圾角色与货币角色在结构上得以统一：单一协议常数 β 同时提供应用层的攻击成本下限和货币层的供给管理。

定理 11 (手续费回收抵抗)**.** *在部分手续费燃烧机制下，控制网络哈希算力比例 μ 的攻击者最多能够回收其攻击成本的 $(1-\beta)\mu$，从而确保在 $\mu < \beta$ 时净攻击成本保持正值。*

均衡分析。定理 11 给出了净攻击成本保持正值的充分条件（$\mu < \beta$）。挖矿博弈中的均衡 μ 是否满足该条件，取决于挖矿收益与垃圾攻击收益的相对大小。矿工-攻击者需比较诚实挖矿利润（与 μ 成正比）和垃圾攻击利润（与消息量成正比，受攻击者基础设施约束）。由于诚实挖矿收益随 μ 线性增长而垃圾攻击收益不然，大矿工相对于小矿工发动垃圾攻击的激励更弱。这表明对于合理的 β 值，在典型场景下 $\mu < \beta$ 条件具有自我强化性，但形式化证明需要显式规定挖矿和垃圾攻击的成本函数——留待未来工作完成。

5.4 无预言机的价格响应性

该协议通过以加密货币单位标记所有参数，同时允许市场参与者基于外部价值评估调整销毁金额，在无需外部预言机的情况下实现价格稳定：

命题 12 (价格响应性)**.** *当加密货币的外部价格上涨因子 γ 时，理性用户将其销毁金额降低约 $1/\gamma$，无需更改协议即可维持稳定的法币计价成本。*

该机制无需预言机输入或治理决策即可自动适应外部价格变化。无预言机属性是与 Chancellor [2026a] 自适应货币框架共同的设计要求；使无预言机货币设计成为可能的 PoW 安全支出可读性由 Chancellor [2026b] 建立。

6 安全分析

6.1 密码学安全属性

该协议提供标准密码学安全保证：

定理 13 (消息机密性)**.** *在判定性 Diffie-Hellman 假设下，对于没有发送方或接收方私钥访问权限的攻击者而言，消息内容在计算上与随机值不可区分。*

定理 14 (支付不可抵赖性)**.** *在数字签名不可伪造性假设下，没有付款方私钥访问权限的攻击者无法伪造支付证明。*

定理 15 (身份真实性)**.** *在哈希函数抗碰撞性和签名不可伪造性假设下，攻击者在没有合法用户私钥的情况下无法冒充合法用户。*

6.2 针对理性攻击者的经济安全性

我们分析针对具有经济动机的攻击者的安全性：

定理 16 (垃圾攻击无利可图性)**.** *对于每条成功消息提取价值为 v_{spam}、成功概率为 $p_{success}$ 的垃圾攻击，当满足以下条件时垃圾攻击无利可图：*

$$B > \frac{v_{spam} \cdot p_{success}}{1 - \beta\mu}$$

其中 β 为燃烧比例，μ 为攻击者的挖矿算力比例。

这为参数选择提供了具体的界，以确保经济安全性。

6.3 隐私与匿名保证

尽管 CashWeb 不提供完全匿名性（公钥作为持久标识符），但它提供若干隐私保护：
消息内容隐私：所有消息均经端到端加密，防止中继服务器和密钥服务器访问内容。
通信模式隐私：中继服务器仅能看到其托管用户的加密消息，限制了全局流量分析能力。

假名不可关联性：用户可生成多个假名而不暴露其相互关联，实现分区身份管理。

元数据隐私的局限性。内容隐私并不意味着元数据隐私。销毁交易在区块链上公开可见；能够同时观察区块链和自身接入消息流量的中继服务器，可将销毁交易时间与消息到达时间进行关联，从而将付款方身份与假名发送者联系起来。密钥服务器还能获知公钥与中继服务器地址之间的映射关系。本系统提供内容隐私和假名发送者身份，但不能防范中继服务器或密钥服务器对手的关联分析。需要更强匿名性保证的用户应通过混合网络或其他与本协议正交的匿名化层进行路由。这是当前设计的已知局限；更强的隐私属性留待未来工作处理。

6.4 联合攻击抵抗能力

我们考虑由恶意网络参与者联合发动的攻击：

命题 17 (部分共谋下的中继服务器可用性)**.** *在以下条件下：(i) 用户独立地与 $k \geq 2$ 台中继服务器保持连接；(ii) 每台中继服务器独立地以概率 $f < 1/2$ 为恶意节点；(iii) 恶意服务器静默丢弃消息：投递失败的概率至多为 f^k，当 $k \geq \log(1/\delta)/\log(1/f)$ 时该概率低于目标值 δ。*

形式化拜占庭协议保证（在标准 BFT 假设 [Lamport et al., 1982] 下，$f < 1/3$ 拜占庭节点时的安全性和活性）适用于密钥服务器网络。中继层要求可用性而非一致性；冗余投递即已足够。

密钥服务器时延说明。跨地理分布式密钥服务器的 *BFT* 共识会引入固有的往返时延——对于全球分布式节点集，通常为 *50–500* 毫秒，且最终确认需要多轮通信。因此，密钥注册和更新在密钥服务器网络中的传播时延约为秒级。这对于身份管理（注册不频繁）是可接受的，但无法提供实时密钥撤销保证。中继服务器在信任新注册密钥前应设置短暂宽限期，并对高价值或初次联系交互实施密钥新鲜度检查。

定理 18 (密钥服务器联合抵抗)**.** *只要不超过 1/3 的密钥服务器为诚实节点，密钥服务器网络即利用标准拜占庭容错（BFT）技术维持可用性和一致性。*

7 实现注意事项

7.1 协议消息格式与接口

CashWeb 使用 Protocol Buffers Google [2015] 进行结构化消息序列化，使用 RESTful HTTP 接口进行网络通信。主要消息格式如下：

身份注册：

```
message IdentityRegistration {
  bytes public_key = 1;
  string relay_address = 2;
  ProofOfPayment proof = 3;
  int64 timestamp = 4;
}
```

加密消息：

```
message EncryptedMessage {
  bytes sender_key = 1;
  bytes recipient_key = 2;
  bytes encrypted_content = 3;
  ProofOfPayment burn_proof = 4;
  int64 timestamp = 5;
}
```

支付证明:

```
message ProofOfPayment {
  bytes transaction_id = 1;
  bytes signature = 2;
  uint64 burn_amount = 3;
  bytes commitment_data = 4;
}
```

7.2 中继服务器经济学与激励机制

中继服务器必须在保护用户隐私的同时维持经济可持续性。经济模型包括:

收入来源:

- 新用户注册费
- 可选增值服务（扩大存储容量、优先投递）
- 交易手续费分成（适用于同时参与挖矿的中继服务器）

成本结构:

- 加密消息的存储成本
- 消息投递的带宽成本
- 支付验证的计算成本

竞争动态：用户可通过标准 API 在中继服务器之间迁移，形成竞争性定价和服务质量的市场压力。

7.3 客户端燃烧管理与用户体验

面向用户的客户端必须在保证安全的同时抽象加密货币管理的复杂性:

自动销毁金额选择：客户端可根据以下因素实现销毁金额的自动选择:

- 消息优先级（紧急消息使用较高销毁率）
- 接收方关系（与陌生人首次联系使用较高销毁金额）

- 网络拥塞程度（根据观测到的投递时间动态调整）

钱包集成：通过标准化接口与加密货币钱包无缝集成，同时支持托管式和非托管式钱包架构。

隐私保护：客户端软件内置防流量分析和时序攻击保护，通过消息批处理和随机延迟实现。

7.4 与现有基础设施的集成

该协议设计为可与现有通信系统增量部署：

电子邮件网关：CashWeb 消息可通过处理加密货币操作的网关服务与传统电子邮件双向桥接。

Web 集成：JavaScript 库通过 WebSocket 连接实现 CashWeb 在 Web 应用程序中的直接集成。

移动端支持：原生移动端 SDK 提供省电实现，并为移动环境配备适当的密钥管理方案。

8 威胁模型与攻击分析

8.1 垃圾消息攻击与拒绝服务攻击

直接垃圾消息攻击：攻击者试图发送大量未经请求的消息。燃烧机制使此类攻击代价高昂：以最低销毁金额 B_{min} 发送 N 条垃圾消息至少需要花费 $N \cdot B_{min}$。垃圾攻击要实现盈利，攻击者每条成功消息必须提取超过 B_{min} 的价值，这对大多数垃圾消息类别而言难以实现。

资源耗尽攻击：攻击者试图以消息处理请求压垮中继服务器。支付要求限制了攻击规模，同时中继服务器可根据支付金额实施额外的速率限制。

存储耗尽攻击：攻击者发送合法的已付费消息以填满中继服务器存储空间。中继服务器可实施存储管理策略，包括自动删除旧消息和提供付费存储层级。

8.2 经济攻击

手续费回收攻击：控制挖矿基础设施的攻击者试图通过交易手续费回收已销毁的资金。部分手续费燃烧机制（第 5 节）将回收量限制在 $(1-\beta)\mu$，其中 β 为燃烧比例，μ 为攻击者的哈希算力比例。

女巫攻击：攻击者创建大量虚假身份以扩大攻击能力。每个身份均需通过注册费进行经济承诺，使攻击预算与女巫身份数量之间呈线性关系（命题 5）。

市场操纵：攻击者试图操纵加密货币价格以影响销毁经济学。无预言机设计使系统能够响应外部价格变化，但不依赖于外部价格操纵。

8.3 基础设施攻击

中继服务器审查：恶意中继服务器拒绝投递特定发送方的消息。用户可通过投递确认检测审查行为，并迁移至其他中继服务器。

密钥服务器操纵：攻击者试图发布虚假身份信息。密码学签名要求防止身份冒充，而分布式密钥服务器网络则提供抵御单个服务器被攻陷的抗毁能力。

网络分区：攻击者试图将用户或服务器与更广泛的网络隔离。联邦架构提供多条通信路径，而加密货币网络则提供全局协调机制。

8.4 隐私攻击

流量分析：攻击者监控网络流量以推断通信模式。端到端加密保护消息内容，而客户端可使用洋葱路由或类似技术提供额外保护。

时序攻击：攻击者关联消息发送与接收时间以识别通信关系。客户端可实施随机延迟和消息批处理以降低时序关联性。

支付分析：攻击者分析区块链交易以将支付与消息相关联。为每笔燃烧交易使用新鲜地址以及适当的交易混淆技术可提供额外的隐私保护。

9 评估与讨论

9.1 理论分析结果

我们的理论分析证明了若干关键属性：

垃圾抵抗可扩展性：垃圾攻击成本随攻击规模线性增长，提供可预测的保护保证。以最低销毁率 $B_{min} = \$0.01$ 每条消息计算，发送 100 万条垃圾消息至少需要花费 \$10,000，尚不包括运营开销。

合法用户可及性：对于每条消息 \$0.01 至 \$0.10 范围内的销毁率，合法用户的经济门槛仍然极低（与短信费用相当），同时提供实质性的垃圾消息威慑。

网络效应收益：随着网络采用率的提高，合法消息的价值比垃圾攻击效率增长更快，为系统增长创造正向反馈。

9.2 与现有系统的比较

表 1 将 CashWeb 与现有反垃圾和消息方案进行对比：

系统	去中心化	反垃圾	隐私	可扩展性
电子邮件（SMTP）	部分	差	差	高
Hashcash	是	中	好	差
Matrix	部分	差	好	中
Signal	否	好	优秀	中
CashWeb	是	好	好	高

表 1: 消息系统关键属性对比

9.3 实现经验

Stamp 社交网络提供了 CashWeb 核心机制的早期部署经验。本文的协议规范是对原型架构的形式化补全，该原型已实现燃烧发言、联邦中继及密钥服务器身份等核心机制。部署观察包括：

用户接受度：当销毁金额低于每条消息约 \$0.05 时，基于销毁的消息传递对用户不构成明显摩擦；更高金额对日常通信会产生明显阻力。

垃圾消息减少：经济销毁要求能有效遏制自动化垃圾消息；每条消息的经济成本引入了一个高于典型垃圾活动预期收益的门槛。在成熟的大规模部署中进行系统测量留待未来的实证工作。

基础设施成本：初步测试中，中继服务器资源消耗随用户数量大致线性增长。支付证明验证相较于基础消息路由增加的计算开销适度——主要由区块链状态查询而非密码学验证主导。

以上观察均基于小规模部署的非正式记录。它们验证了该方案的可行性，但不能替代规模化的严格性能基准测试。

9.4 局限性与权衡

CashWeb 方法涉及若干固有权衡：

经济门槛：尽管对合法用户而言销毁金额极小，但对经济欠发达地区的用户而言仍可能构成障碍。未来工作可探索浮动销毁率或替代性价值证明机制。

加密货币依赖性：该系统需要访问加密货币网络，这可能在访问受限地区或对加密货币感到不适的用户群体中限制采用。

恢复复杂性：密钥丢失场景需要比传统中心化系统更复杂的恢复程序，可能对非技术用户造成可用性挑战。

9.5 未来扩展与应用

CashWeb 协议为众多扩展提供了基础：

声誉系统：基于收到的消息反馈的用户声誉评分，可实现动态销毁率调整和改进的垃圾过滤。

内容市场：燃烧机制可扩展至内容微支付，为信息共享和媒体分发开辟新的经济模式。

物联网通信：具备自动支付能力的机器间通信，可为设备支付带宽和处理资源的物联网应用开辟新场景。

去中心化社交网络：发布-订阅系统为具有经济激励对齐的完全去中心化社交媒体平台提供基础设施。

10 结论

我们介绍了 CashWeb——一套综合性协议套件，解决了驱动互联网通信系统中心化的根本性经济激励问题。通过对燃烧发言反垃圾机制的正式规范、联邦式基础设施设计以及经济安全分析，我们证明加密货币集成能够恢复去中心化互联网通信的原始愿景，同时提供优于现有方案的反垃圾能力。

该协议强调基于经济而非计算或监管的反垃圾机制，带来若干优势：可预测的保护保证、支持全球部署的可扩展性，以及对影响计算型方案的技术军备竞赛的抵抗能力。Stamp 社交网络的实施经验验证了该方法的实际可行性。

未来工作应着重降低弱势群体的经济门槛、改善密钥管理的可用性，并探索将该协议应用于消息传递之外、同样面临类似中心化压力的其他通信与协调问题。

CashWeb 的终极目标不仅仅是技术创新，更是恢复用户对数字通信的主权——在应对驱动原始中心化的经济现实的同时，回归互联网去中心化与用户赋权的创始原则。

参考文献

Adam Back. Hashcash—a denial of service counter-measure. Web document, 2002. URL `http://www.hashcash.org/papers/hashcash.pdf`.

Gary S. Becker. Crime and punishment: An economic approach. *Journal of Political Economy*, 76 (2):169–217, 1968.

Shammah Chancellor. Adaptive PoW monetary policy without oracles: A constructive mechanism for pseudo-stability via work-coupled tail emission and burn. Preprint, 2026a.

Shammah Chancellor. Security expenditure, energy, and issuance legibility in permissionless consensus. Preprint, 2026b.

Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. *Annual International Cryptology Conference*, pages 139–147, 1992.

Stefan Dziembowski, Sebastian Faust, and Kristina Hostáková. General state channel networks. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 949–966, 2018.

Hal Finney. Reusable proofs of work. Web document, 2004. URL `https://nakamotoinstitute.org/finney/rpow/index.html`.

Loki Foundation. Session protocol specification. Technical documentation, 2020. URL `https://getsession.org/`.

Matrix.org Foundation. Matrix specification. Web document, 2019. URL `https://matrix.org/docs/spec/`.

Gmail, October 2018. URL `https://twitter.com/gmail/status/1055806807174725633`.

Jennifer Golbeck and James Hendler. Computing and applying trust in web-based social networks. *University of Maryland*, 2005.

Google. Protocol buffers version 3 language specification. Web document, 2015. URL `https://developers.google.com/protocol-buffers/docs/reference/proto3-spec`.

Dr. John C. Klensin. Simple Mail Transfer Protocol. RFC 5321, October 2008. URL `https://rfc-editor.org/rfc/rfc5321.txt`.

Dan Kohn, Ken Murchison, and Charles Lindsey. Netnews Article Format. RFC 5536, November 2009. URL `https://rfc-editor.org/rfc/rfc5536.txt`.

Litmus Labs. Email client market share. Web document, 2020. URL `https://nakamotoinstitute.org/finney/rpow/index.html`.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

Ben Laurie and Richard Clayton. Proof-of-work proves not to work. *Workshop on Economics and Information Security*, 2004.

Charles Lindsey and Russ Allbery. Netnews Architecture and Protocols. RFC 5537, November 2009. URL `https://rfc-editor.org/rfc/rfc5537.txt`.

Andrew Miller and Joseph J LaViola Jr. Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin. *University of Central Florida*, 2014.

Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Web document, 2008. URL `https://bitcoin.org/bitcoin.pdf`.

Status Network. Status: A mobile ethereum os. Whitepaper, 2017. URL `https://status.im/whitepaper.pdf`.

Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. 2016.

Pete Resnick. Internet Message Format. RFC 5322, October 2008. URL `https://rfc-editor.org/rfc/rfc5322.txt`.

Peter Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. RFC 3920, October 2004a. URL `https://rfc-editor.org/rfc/rfc3920.txt`.

Peter Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 3921, October 2004b. URL `https://rfc-editor.org/rfc/rfc3921.txt`.