CashWeb: A Cryptocurrency-Integrated Protocol for Federated Anti-Spam Messaging and Publish-Subscribe Systems

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

February 21, 2026
Version 1.1

Abstract

We present CashWeb, a comprehensive protocol suite that integrates cryptocurrency-based economic mechanisms with federated messaging infrastructure to enable spam-resistant communication without centralized moderation. The protocol employs a novel "burn-to-speak" anti-spam mechanism where message senders attach small cryptocurrency payments that are cryptographically verifiable but do not require trusted intermediaries. We provide formal specifications for three core components: (1) a Proof-of-Payment (POP) protocol that links on-chain value to off-chain actions, (2) federated messaging with cryptographic identity management, and (3) a publish-subscribe system for topic-based broadcasting. Our economic analysis demonstrates that the burn mechanism creates optimal deterrence against spam while preserving accessibility for legitimate communication. The system architecture leverages established web standards (HTTP/2, WebSockets) and cryptographic primitives to enable seamless integration with existing infrastructure. Theoretical analysis shows the protocol achieves spam resistance with costs scaling sublinearly in network size, while empirical deployment demonstrates practical viability for real-time messaging applications.

1 Introduction

1.1 The Centralization Problem

The evolution of internet communication protocols reveals a fundamental tension between decentralization and usability. Early systems like Usenet Kohn et al. [2009], Lindsey and Allbery [2009], SMTP Resnick [2008], Klensin [2008], and XMPP Saint-Andre [2004a,b] were designed as federated networks enabling peer-to-peer communication without central authorities. However, the asymmetric cost structure inherent in these protocols—where message processing costs are borne by recipients while sending costs remain negligible—created economic incentives for abuse that ultimately drove users toward centralized platforms.

By 2020, the concentration of communication infrastructure had reached unprecedented levels: Google, Apple, and Microsoft collectively controlled 85% of email client market share Labs [2020], while Facebook reported over 2 billion users and Gmail served 1.5 billion active accounts Gmail [2018]. This centralization, while providing user convenience, introduced systemic risks including censorship, surveillance, and single points of failure that compromise the original vision of distributed internet communication.

1.2 Economic Anti-Spam Mechanisms

The fundamental challenge in decentralized messaging systems is spam prevention without centralized filtering. Traditional approaches rely on either computational costs (Hashcash Back [2002]) or reputation systems that require persistent identity verification. While these mechanisms provide some protection, they suffer from significant limitations: computational proof-of-work scales poorly with modern hardware acceleration, and reputation systems create barriers to entry for new users while remaining vulnerable to Sybil attacks.

The advent of cryptocurrency networks, particularly Bitcoin Nakamoto [2008], enables a new approach: economic proof-of-payment mechanisms where message authenticity is verified through cryptographic proof of value destruction rather than computational work. This approach, originally conceived by Finney Finney [2004] as "Reusable Proof of Work," can now be implemented without trusted intermediaries using decentralized cryptocurrency networks.

1.3 Contributions

This paper presents CashWeb, a comprehensive protocol suite that addresses the centralization problem through three primary contributions:

1. **Formal Proof-of-Payment Protocol**: We specify a complete cryptographic protocol for linking on-chain cryptocurrency transactions to off-chain messaging actions, enabling verifiable anti-spam mechanisms without requiring trusted parties.

2. **Federated Infrastructure Design**: We present a scalable architecture combining keyservers for identity management and relay servers for message routing, designed to support the transaction volumes necessary for widespread adoption while maintaining decentralization.

3. **Economic Security Analysis**: We provide theoretical analysis demonstrating that appropriate burn rate mechanisms achieve optimal trade-offs between spam deterrence and legitimate user accessibility, with formal bounds on attack costs and success probabilities.

The protocol is designed for deployment on the Lotus cryptocurrency network Lotus Development Team [2021], a Proof-of-Work chain with native support for value-bearing `OP_RETURN` outputs and an adaptive monetary policy as specified in Chancellor [2026a]. Lotus is currently operational;

CashWeb provides the application-layer protocol completing the economic stack. The protocol leverages established web standards and maintains compatibility with existing internet infrastructure. Deployment experience with the Stamp social network provides early validation of the core approach.

This paper is the third in a series. Chancellor [2026b] establishes the theoretical foundation: in any permissionless consensus system, equilibrium security expenditure is anchored to expected value at risk, and Proof-of-Work uniquely renders this expenditure protocol-legible via an adversarially revealed work signal. Chancellor [2026a] develops the monetary policy implications, specifying an oracle-free two-loop mechanism that uses the PoW work signal for adaptive tail emission and burn-to-speak as an endogenous supply sink. The present paper specifies the burn-to-speak mechanism in full as a federated messaging protocol. Consequently, burn-to-speak serves a dual role across the stack: spam deterrence at the application layer and endogenous monetary supply management at the protocol layer.

1.4 Paper Organization

Section 2 reviews related work in anti-spam mechanisms and federated messaging. Section 3 establishes our system model and threat assumptions. Section 4 provides formal specifications of the core protocols. Section 5 analyzes the economic properties and spam resistance guarantees. Section 6 examines security properties and attack resistance. Section 7 discusses practical deployment considerations. Section 9 presents theoretical and empirical evaluation results.

2 Background and Related Work

2.1 Economic Anti-Spam Mechanisms

The use of economic incentives for spam prevention has a rich history in distributed systems research. Dwork and Naor Dwork and Naor [1992] first proposed requiring computational proof-of-work for email sending, implemented in Hashcash Back [2002] through SHA-256 hash preimage puzzles. While theoretically sound, computational approaches suffer from several practical limitations: (1) the cost of computation varies dramatically across different hardware, creating inequity, (2) modern ASICs and GPU acceleration can solve puzzles orders of magnitude faster than consumer hardware, and (3) the optimal difficulty level requires constant adjustment as computational capabilities evolve.

Laurie and Clayton Laurie and Clayton [2004] proposed proof-of-work systems using memory-hard functions to reduce hardware advantages, but these approaches still require significant energy expenditure and create barriers for resource-constrained devices. Alternative reputation-based approaches Golbeck and Hendler [2005] require persistent identity systems that conflict with privacy goals and create high barriers to entry for legitimate new users.

2.2 Cryptocurrency-Based Access Control

The emergence of programmable cryptocurrency networks enables more sophisticated economic mechanisms. Miller et al. Miller and LaViola Jr [2014] proposed using Bitcoin for anonymous micropayments, but their approach requires interactive protocols that do not scale to messaging applications. Recent work on payment channels Poon and Dryja [2016] and state channels Dziembowski et al. [2018] provides mechanisms for low-latency micropayments, but the requirement for pre-funded channels creates usability barriers.

Our approach differs by using unrecoverable value destruction ("burning") rather than value transfer, eliminating the need for recipient payment infrastructure while maintaining strong economic incentives against abuse.

2.3 Federated Messaging Systems

Federated messaging architectures balance the benefits of decentralization with practical scalability requirements. XMPP Saint-Andre [2004a] demonstrates the viability of federated real-time messaging, but lacks economic spam prevention mechanisms. Matrix Foundation [2019] provides modern federated messaging with end-to-end encryption, but relies on centralized identity providers and lacks anti-spam mechanisms beyond rate limiting.

Recent blockchain-based messaging systems include Status Network [2017] and Session Foundation [2020], which provide decentralized identity management but do not address the economic incentive problems that drive centralization. Our approach integrates economic mechanisms directly into the protocol design to address these fundamental incentive misalignments.

3 System Model and Architecture

3.1 Network Participants

The CashWeb network comprises four types of participants:

Definition 1 (Network Participants)**.** *Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of users, $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ be the set of relay servers, $\mathcal{K} = \{k_1, k_2, \ldots, k_\ell\}$ be the set of keyservers, and $\mathcal{T} = \{t_1, t_2, \ldots, t_p\}$ be the set of publish-subscribe topics.*

Users ($u_i \in \mathcal{U}$): End participants who send and receive messages. Each user controls a cryptocurrency wallet and can generate cryptographic proofs of payment. Users interact with the network through client software that manages key material and payment transactions.

Relay Servers ($r_i \in \mathcal{R}$): Federated servers that store and forward encrypted messages on behalf of users. Relay servers verify payment proofs before accepting messages, providing the first line of spam defense. They do not have access to message contents due to end-to-end encryption.

Keyservers ($k_i \in \mathcal{K}$): Distributed servers that maintain a global registry of user public keys and associated metadata. Keyservers enable users to discover relay servers and establish secure communication channels without requiring prior contact.

Topics ($t_i \in \mathcal{T}$): Named channels in the publish-subscribe system that enable broadcast communication. Topics are decentralized—any user can create a topic and any relay server can propagate topic messages based on subscriber lists.

3.2 Threat Model

We consider a Byzantine adversarial environment where participants may deviate arbitrarily from protocol specifications. Our threat model includes:

Spam Attackers: Adversaries attempting to send large volumes of unsolicited messages to disrupt communication or waste network resources. We assume attackers have significant but finite computational and financial resources.

Sybil Attackers: Adversaries who create large numbers of fake identities to amplify their message-sending capacity or manipulate reputation systems.

Network Infrastructure Attackers: Adversaries who control relay servers or keyservers and may attempt censorship, traffic analysis, or denial of service attacks.

Economic Attackers: Adversaries who attempt to manipulate the economic mechanisms, including fee recycling attacks where attackers recover burned funds through mining or validator control.

We assume the underlying cryptocurrency network (e.g., Bitcoin, Lotus) provides standard security guarantees including transaction immutability and resistance to double-spending attacks. We do not consider attacks on the cryptocurrency network itself.

3.3 Design Principles

The protocol design adheres to several key principles:

Economic Security: Anti-spam protection should emerge from economic incentives rather than computational barriers or centralized moderation, ensuring that attacks become prohibitively expensive while legitimate use remains affordable.

Privacy by Design: Message contents and communication patterns should remain private to third parties, including network infrastructure providers.

Federated Scalability: The system should support global-scale adoption through federated architecture while maintaining decentralization properties.

Backward Compatibility: The protocol should integrate with existing internet infrastructure and development frameworks to minimize adoption barriers.

4 Core Protocol Specifications

4.1 Proof-of-Payment (POP) Protocol

The Proof-of-Payment protocol enables cryptographic verification that a specific amount of cryptocurrency has been burned (permanently destroyed) in association with a particular off-chain

action. Unlike traditional payment systems that transfer value between parties, the POP protocol creates verifiable proof of value destruction that cannot be recovered or double-spent.

Definition 2 (Proof-of-Payment Protocol)**.** *The POP protocol is a tuple* (Setup, Commit, Verify) *where:*

- $\text{Setup}(1^\lambda) \to pp$*: Generates public parameters for security parameter* λ*.*
- $\text{Commit}(sk, m, v) \to (\pi, \tau)$*: Given secret key* sk*, message* m*, and value* v*, outputs proof* π *and transaction* τ*.*
- $\text{Verify}(\pi, \tau, m, v) \to \{0, 1\}$*: Verifies that transaction* τ *burns value* v *and is cryptographically linked to message* m*.*

4.1.1 Unspendable Output Construction

Before specifying the algorithm, we define the core primitive:

Definition 3 (Unspendable Address / OP_RETURN Output)**.** *Given a 32-byte commitment value* h_m*,* $\text{UnspendableAddress}(h_m)$ *denotes the output script* `OP_RETURN` $\langle h_m \rangle$*. A transaction output of value* v *sent to this script is verifiably destroyed: (i)* `OP_RETURN` *outputs are consensus-invalid as spending inputs, so no party can construct a valid spending transaction; (ii) conforming nodes mark such outputs as UTXO-exempt, permanently removing the value from circulation. A 32-byte SHA-256 hash payload fits within the 80-byte* `OP_RETURN` *data limit, leaving room for a 4-byte protocol identifier and version field.*

The Lotus network, the intended deployment target for CashWeb, supports nonzero-value `OP_RETURN` outputs by consensus rule, making the above the primary construction. For chains that prohibit nonzero-value `OP_RETURN` outputs (standard Bitcoin policy), an alternative construction sends v to address $\text{P2PKH}(\text{Hash160}(\texttt{0x0000...00} \| h_m))$: a syntactically valid P2PKH address for which no private key exists (since finding a preimage of Hash160 is computationally infeasible). This provides probabilistic unspendability rather than the consensus-enforced unspendability of the primary construction.

4.1.2 Construction

Our POP protocol construction leverages OP_RETURN outputs in Bitcoin-style transactions to embed commitment data while ensuring fund destruction:

The construction ensures several critical properties:

Unrecoverability: Funds sent to $addr_{burn}$ are provably unspendable because the address is derived from a hash with no known preimage.

Uniqueness: Each message-payment pair produces a unique commitment that cannot be reused for different messages.

Verifiability: Any party can verify the proof by checking the transaction on the blockchain and validating the cryptographic signatures.

Algorithm 1 Proof-of-Payment Construction

Require: Message m, burn amount v, user secret key sk
Ensure: Proof π and transaction 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

4.2 Identity Management and Key Registration

Users in CashWeb maintain pseudonymous identities based on public-key cryptography. Each identity consists of a key pair (sk, pk) where the public key serves as the user's global identifier.

4.2.1 Identity Registration

New identities are registered with the keyserver network through the following protocol:

Algorithm 2 Identity Registration

Require: User key pair (sk, pk), relay server address $addr_{relay}$, registration fee v_{reg}
Ensure: Identity record in keyserver network

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

4.2.2 Key Rotation and Recovery

The protocol supports key rotation to enable recovery from key compromise or loss. Users can pre-register recovery keys or use hierarchical deterministic key derivation for seamless rotation:

4.3 Federated Messaging with Burn-to-Send

The core messaging protocol integrates cryptographic message delivery with economic anti-spam protection. Messages are encrypted end-to-end and include proof-of-payment to demonstrate sender commitment.

4.3.1 Message Verification and Storage

Relay servers verify incoming messages before storage:

Algorithm 3 Key Rotation

Require: Current key pair (sk_{old}, pk_{old}), new key pair (sk_{new}, pk_{new}), rotation fee v_{rot}
Ensure: Updated identity record

$rotation\_msg \leftarrow (pk_{old}, pk_{new}, \text{NOW}())$
$\sigma_{old} \leftarrow \text{SIGN}(sk_{old}, rotation\_msg)$
$\sigma_{new} \leftarrow \text{SIGN}(sk_{new}, rotation\_msg)$
$(\pi_{rot}, \tau_{rot}) \leftarrow \text{POP.COMMIT}(sk_{new}, rotation\_msg, v_{rot})$
$update \leftarrow (pk_{old}, pk_{new}, \sigma_{old}, \sigma_{new}, \pi_{rot})$
for each keyserver $k_i \in \mathcal{K}$ **do**
    $\text{SEND}(update, k_i)$
end for

Algorithm 4 Message Sending Protocol

Require: Recipient public key pk_{recv}, message content *content*, burn amount v_{msg}
Ensure: Message delivered to recipient's relay server

$(pk_{sender}, sk_{sender}) \leftarrow \text{GETUSERKEYS}()$
$relay_{recv} \leftarrow \text{KEYSERVERLOOKUP}(pk_{recv})$
$k_{shared} \leftarrow \text{ECDH}(sk_{sender}, pk_{recv})$
$msg_{encrypted} \leftarrow \text{ENCRYPT}(k_{shared}, content)$
$message \leftarrow (pk_{sender}, pk_{recv}, msg_{encrypted}, \text{TIMESTAMP}())$
$(\pi_{msg}, \tau_{msg}) \leftarrow \text{POP.COMMIT}(sk_{sender}, message, v_{msg})$
$delivery\_req \leftarrow (message, \pi_{msg})$
$\text{HTTPPOST}(delivery\_req, relay_{recv})$

Algorithm 5 Message Verification by Relay Server

Require: Message delivery request $delivery\_req = (message, \pi_{msg})$
Ensure: Message accepted or rejected

$(message, \pi_{msg}) \leftarrow delivery\_req$
$(pk_{sender}, pk_{recv}, msg_{encrypted}, timestamp) \leftarrow message$
if $\text{POP.VERIFY}(\pi_{msg}, message) \neq 1$ **then**
    return "Rejected: Invalid payment proof"
end if
if $\text{BURNAMOUNT}(\pi_{msg}) < v_{min}$ **then**
    return "Rejected: Insufficient burn amount"
end if
$\text{STOREMESSAGE}(message, \pi_{msg})$
return "Accepted"

4.4 Publish-Subscribe with Burn-to-Broadcast

The publish-subscribe system extends the messaging protocol to support topic-based broadcasting. Users can subscribe to topics and receive all messages posted to those topics, with message priority determined by burn amounts.

Definition 4 (Topic Subscription)**.** *A topic subscription is a tuple* (u_i, t_j, r_k) *indicating that user* u_i *subscribes to topic* t_j *through relay server* r_k*.*

Algorithm 6 Topic Message Broadcasting

Require: Topic identifier *topic_id*, message content *content*, burn amount $v_{broadcast}$
Ensure: Message distributed to all topic subscribers
1: $(pk_{sender}, sk_{sender}) \leftarrow$ GetUserKeys()
2: $topic\_msg \leftarrow (pk_{sender}, topic\_id, content,$ Timestamp())
3: $(\pi_{broadcast}, \tau_{broadcast}) \leftarrow$ POP.Commit$(sk_{sender}, topic_msg, v_{broadcast})$
4: $broadcast\_req \leftarrow (topic\_msg, \pi_{broadcast})$
5: **for** each relay server $r_i \in \mathcal{R}$ **do**
6: SendToRelayIfSubscribers$(broadcast_req, topic_id, r_i)$
7: **end for**

The broadcast mechanism includes priority-based message ordering and rate limiting based on burn amounts:

Algorithm 7 Topic Message Prioritization

Require: Set of topic messages $M = \{m_1, m_2, \ldots, m_k\}$ with burn amounts $\{v_1, v_2, \ldots, v_k\}$
Ensure: Prioritized message delivery schedule
1: $priority\_queue \leftarrow$ EmptyQueue()
2: **for** each message $m_i \in M$ **do**
3: $priority_i \leftarrow f(v_i)$ where f is monotonically increasing
4: Insert$(priority_queue, m_i, priority_i)$
5: **end for**
6: **while** NotEmpty$(priority_queue)$ **and** BandwidthAvailable() **do**
7: $m_{next} \leftarrow$ PopMax$(priority_queue)$
8: DeliverMessage(m_{next})
9: **end while**

5 Economic Analysis and Anti-Spam Properties

5.1 Burn Rate Economics and Equilibrium Analysis

The effectiveness of the burn-to-speak mechanism depends on establishing burn rates that make spam attacks economically unviable while preserving accessibility for legitimate users. We model this as a game-theoretic equilibrium problem, following the optimal deterrence framework of Becker [Becker, 1968], extended to a decentralized setting where enforcement is implemented through cryptographic proof of value destruction.

Definition 5 (Spam Attack Cost)**.** *For an attacker attempting to send N spam messages with burn amount B per message, the total attack cost is:*

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

where $C_{operational}(N)$ represents computational and infrastructure costs.

Definition 6 (Legitimate User Utility)**.** *A legitimate user's utility from sending a message with burn amount B is:*

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

where $V_{communication}$ is the value derived from successful message delivery and $C_{friction}(B)$ represents usability costs.

The optimal burn rate B^* maximizes legitimate user adoption while minimizing spam viability. To obtain a tractable form, we specify functional forms:

Definition 7 (Social Welfare Function)**.** *Define:*

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

where $U_0 > 0$ is per-message value for a representative legitimate user, $\alpha > 0$ is their marginal cost of burn, $D > 0$ is the social damage per spam message, and $\sigma > 0$ is the spam volume at unit burn cost. The term αB captures the reduction in legitimate-user surplus from higher burn; the term $D\sigma/B$ captures total spam damage, which falls as $1/B$ under rational spammer economics (spammer equates marginal spam revenue to marginal burn cost).

Under this specification, the optimal burn rate is:

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

balancing deterrence against accessibility in closed form. Higher social spam damage D or higher spam propensity σ call for higher B^*; lower legitimate-user marginal cost α also raises B^*.

Theorem 8 (Optimal Burn Rate)**.** *The optimal burn rate B^* satisfies:*

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

where α represents the social cost of spam and $E[N_{spam}(B)]$ is the expected number of spam messages at burn rate B.

Proof. Existence. As $B \to 0$, spam is unconstrained and $W(B) \to -\infty$. As $B \to \infty$, legitimate users are priced out and $W(B) \to -\infty$. Since W is continuous, an interior maximum exists by the extreme value theorem on any compact subinterval containing the maximum.

First-order condition. At an interior optimum B^*, the stated condition holds by differentiation.

Uniqueness obtains under the additional assumption that $W(B)$ is strictly concave, which holds when legitimate utility is concave in B and spam volume is convex in $1/B$—standard assumptions in the optimal deterrence literature [Becker, 1968]. □

5.2 Sybil Resistance Quantification

The burn mechanism provides inherent Sybil resistance because each identity requires economic commitment rather than just computational work:

Proposition 9 (Sybil Attack Bounds)**.** *For an attacker with budget $\mathcal{B}$, the maximum number of Sybil identities is bounded by:*

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

where v_{reg} is the identity registration fee and k is the expected number of messages per identity.

This bound demonstrates that Sybil attacks scale linearly with attacker budget rather than computational resources, providing predictable resistance guarantees.

5.3 Fee Recycling Attack Prevention

A critical security consideration is preventing attackers from recovering burned funds through control of mining or validation infrastructure. We address this through partial fee burning:

Definition 10 (Partial Fee Burn Mechanism)**.** *For each transaction with fees F, a fraction $\beta \in (0, 1]$ is burned while the remainder $(1 - \beta)F$ is paid to miners:*

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

where $B_{explicit}$ is the explicit burn amount and F is the transaction fee.

The parameter β is shared with the adaptive monetary policy framework in Chancellor [2026a], where the same fee burn fraction simultaneously functions as an endogenous supply deflation mechanism. The anti-spam and monetary roles are therefore structurally unified: a single protocol constant β provides both application-layer attack-cost floors and monetary-layer supply management.

Theorem 11 (Fee Recycling Resistance)**.** *Under the partial fee burn mechanism, an attacker controlling fraction μ of network hash power can recover at most $(1 - \beta)\mu$ of their attack costs, ensuring net attack cost remains positive for $\mu < \beta$.*

Equilibrium analysis. Theorem 11 gives a sufficient condition ($\mu < \beta$) for net attack cost to remain positive. Whether the equilibrium μ in the mining game satisfies this condition depends on the relative returns to mining versus spamming. A miner-spammer compares honest mining profit (proportional to μ) against spam profit (proportional to message volume, bounded by attacker

infrastructure). Since honest mining scales with μ and spam does not, large miners have relatively less incentive to spam than small miners. This suggests the $\mu < \beta$ condition is self-enforcing in typical regimes for reasonable β, though a formal proof requires an explicit specification of mining and spamming cost functions—deferred to future work.

5.4 Oracle-Free Price Responsiveness

The protocol achieves price stability without external oracles by denominating all parameters in cryptocurrency units while allowing market participants to adjust burn amounts based on external value assessments:

Proposition 12 (Price Responsiveness)**.** *As the external price of the cryptocurrency increases by factor γ, rational users will decrease their burn amounts by approximately $1/\gamma$, maintaining constant fiat-denominated costs without protocol changes.*

This mechanism enables automatic adjustment to external price changes without requiring oracle inputs or governance decisions. The oracle-free property is a design requirement shared with the adaptive monetary framework of Chancellor [2026a]; the legibility of PoW security expenditure that makes oracle-free monetary design possible is established in Chancellor [2026b].

6 Security Analysis

6.1 Cryptographic Security Properties

The protocol provides standard cryptographic security guarantees:

Theorem 13 (Message Confidentiality)**.** *Under the Decisional Diffie-Hellman assumption, message contents are computationally indistinguishable from random to adversaries without access to sender or recipient private keys.*

Theorem 14 (Payment Non-Repudiation)**.** *Under the unforgeability of digital signatures, payment proofs cannot be forged by adversaries without access to the payer's private key.*

Theorem 15 (Identity Authenticity)**.** *Under the collision resistance of the hash function and unforgeability of signatures, adversaries cannot impersonate legitimate users without access to their private keys.*

6.2 Economic Security Against Rational Adversaries

We analyze security against economically motivated attackers:

Theorem 16 (Spam Attack Unprofitability)**.** *For spam attacks where the value extracted per successful message is v_{spam} and the success probability is $p_{success}$, spam attacks are unprofitable when:*

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

where β is the burn fraction and μ is the attacker's mining power fraction.

This provides concrete bounds for parameter selection to ensure economic security.

6.3 Privacy and Anonymity Guarantees

While CashWeb does not provide full anonymity (public keys serve as persistent identifiers), it provides several privacy protections:

Message Content Privacy: All messages are encrypted end-to-end, preventing relay servers and keyservers from accessing content.

Communication Pattern Privacy: Relay servers only see encrypted messages for their hosted users, limiting global traffic analysis.

Pseudonym Unlinkability: Users can generate multiple pseudonyms without revealing connections between them, providing compartmentalized identity management.

Metadata Privacy Limitations. Content privacy does not imply metadata privacy. Burn transactions are publicly visible on the blockchain; a relay server that observes both the blockchain and its own incoming message traffic can correlate burn transaction timing with message arrival times, potentially linking payer identity to pseudonymous sender. Keyservers additionally learn the mapping from public key to relay server address. The system provides *content* privacy and *pseudonymous* sender identity, but not full unlinkability against a colluding relay-server or keyserver adversary. Users requiring stronger anonymity guarantees should route through a mixnet or similar anonymization layer orthogonal to this protocol. This is a known limitation of the current design; stronger privacy properties are left to future work.

6.4 Resilience to Coalition Attacks

We consider attacks by coalitions of malicious network participants:

Proposition 17 (Relay Server Availability under Partial Collusion)**.** *Under the conditions: (i) users maintain connections to $k \geq 2$ relay servers selected independently, (ii) each relay server is malicious with probability $f < 1/2$ independently, (iii) malicious servers drop messages silently: the probability of delivery failure is at most f^k, falling below target δ for $k \geq \log(1/\delta)/\log(1/f)$.*

Formal Byzantine agreement guarantees (safety and liveness under $f < 1/3$ Byzantine nodes) apply to the keyserver network under standard BFT assumptions [Lamport et al., 1982]. The relay layer requires availability, not consistency; redundant delivery suffices.

Keyserver Latency Note. *BFT consensus across geographically distributed keyservers incurs inherent round-trip latency—typically 50–500 ms for a globally distributed set, with multiple rounds required for finality. Key registrations and updates therefore propagate through the keyserver network on the order of seconds. This is acceptable for identity management (registration is infrequent) but precludes real-time key revocation guarantees. Relay servers should apply a short grace period before trusting newly registered keys, and should implement key freshness checks for high-value or first-contact interactions.*

Theorem 18 (Keyserver Coalition Resistance)**.** *The keyserver network maintains availability and consistency as long as $f < 1/3$ of keyservers are honest, using standard Byzantine fault tolerance techniques.*

7 Implementation Considerations

7.1 Protocol Message Formats and APIs

CashWeb leverages Protocol Buffers Google [2015] for structured message serialization and RESTful HTTP APIs for network communication. Key message formats include:

Identity Registration:

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

Encrypted Message:

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

Proof of Payment:

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

7.2 Relay Server Economics and Incentives

Relay servers must be economically sustainable while maintaining user privacy. The economic model includes:

Revenue Sources:

- Registration fees from new users
- Optional premium services (increased storage, priority delivery)
- Transaction fee sharing (for relay servers that also mine)

Cost Structure:

- Storage costs for encrypted messages
- Bandwidth costs for message delivery
- Computational costs for payment verification

Competitive Dynamics: Users can migrate between relay servers using the standard API, creating market pressure for competitive pricing and service quality.

7.3 Client-Side Burn Management and UX

User-facing clients must abstract the complexity of cryptocurrency management while maintaining security:

Automated Burn Amount Selection: Clients can implement automatic burn amount selection based on:

- Message priority (urgent messages use higher burn rates)
- Recipient relationship (higher burns for first contact)
- Network congestion (dynamic adjustment based on observed delivery times)

Wallet Integration: Seamless integration with cryptocurrency wallets through standardized interfaces, supporting both custodial and non-custodial wallet architectures.

Privacy Protection: Client software includes built-in protections against traffic analysis and timing attacks through message batching and random delays.

7.4 Integration with Existing Infrastructure

The protocol is designed for incremental deployment alongside existing communication systems:

Email Gateway: CashWeb messages can be bridged to/from traditional email through gateway services that handle cryptocurrency operations.

Web Integration: JavaScript libraries enable direct CashWeb integration in web applications through WebSocket connections.

Mobile Support: Native mobile SDKs provide battery-efficient implementations with appropriate key management for mobile environments.

8 Threat Model and Attack Analysis

8.1 Spam and DoS Attacks

Direct Spam Attacks: Adversaries attempt to send large volumes of unsolicited messages. The burn mechanism makes this attack expensive: sending N spam messages with minimum burn amount B_{min} costs at least $N \cdot B_{min}$. For spam to be profitable, attackers must extract value $V > B_{min}$ per successful message, which is unlikely for most spam categories.

Resource Exhaustion Attacks: Adversaries attempt to overwhelm relay servers with message processing requests. The payment requirement limits attack volume, while relay servers can implement additional rate limiting based on payment amounts.

Storage Exhaustion Attacks: Adversaries send legitimate paid messages to fill relay server storage. Relay servers can implement storage management policies including automatic deletion of old messages and premium storage tiers.

8.2 Economic Attacks

Fee Recycling Attacks: Adversaries who control mining infrastructure attempt to recover burned funds through transaction fees. The partial fee burn mechanism (Section 5) limits recovery to $(1 - \beta)\mu$ where β is the burn fraction and μ is the attacker's hash power fraction.

Sybil Attacks: Adversaries create large numbers of fake identities to amplify attack capacity. Each identity requires economic commitment through registration fees, creating linear scaling between attack budget and Sybil capacity (Proposition 5).

Market Manipulation: Adversaries attempt to manipulate cryptocurrency prices to affect burn economics. The oracle-free design makes the system responsive to but not dependent on external price manipulation.

8.3 Infrastructure Attacks

Relay Server Censorship:Malicious relay servers refuse to deliver messages from specific senders. Users can detect censorship through delivery confirmations and migrate to alternative relay servers.

Keyserver Manipulation:Adversaries attempt to publish false identity information. The cryptographic signature requirements prevent impersonation, while the distributed keyserver network provides resilience against individual server compromise.

Network Partitioning:Adversaries attempt to isolate users or servers from the broader network. The federated architecture provides multiple communication paths, while the cryptocurrency network provides a global coordination mechanism.

8.4 Privacy Attacks

Traffic Analysis:Adversaries monitor network traffic to infer communication patterns. End-to-end encryption protects message contents, while clients can use onion routing or similar techniques

for additional protection.

Timing Attacks:Adversaries correlate message sending and receiving times to identify communication relationships. Clients can implement random delays and message batching to reduce timing correlation.

Payment Analysis:Adversaries analyze blockchain transactions to link payments to messages. The use of fresh addresses for each burn transaction and appropriate transaction mixing can provide additional privacy.

9 Evaluation and Discussion

9.1 Theoretical Analysis Results

Our theoretical analysis demonstrates several key properties:

Spam Resistance Scaling: The cost of spam attacks scales linearly with attack volume, providing predictable protection guarantees. For a minimum burn rate of $B_{min} = \$0.01$ per message, sending 1 million spam messages costs at least $\$10,000$, excluding operational overhead.

Legitimate User Accessibility:For burn rates in the range \$0.01 - \$0.10 per message, the economic barrier remains minimal for legitimate users (comparable to SMS messaging costs) while providing substantial spam deterrence.

Network Effect Benefits:As network adoption increases, the value of legitimate messaging increases faster than spam attack efficiency, creating positive feedback for system growth.

9.2 Comparison with Existing Systems

Table 1 compares CashWeb with existing anti-spam and messaging approaches:

System	Decentralized	Anti-Spam	Privacy	Scalability
Email (SMTP)	Partial	Poor	Poor	High
Hashcash	Yes	Moderate	Good	Poor
Matrix	Partial	Poor	Good	Moderate
Signal	No	Good	Excellent	Moderate
CashWeb	Yes	Good	Good	High

Table 1: Comparison of messaging systems across key properties

9.3 Implementation Experience

The Stamp social network provides early deployment experience with CashWeb concepts. The protocol specification in this paper is the formal completion of an architecture whose core mechanisms—burn-to-speak, federated relay, keyserver identity—were prototyped in Stamp. Deployment observations include:

User Acceptance: Users adapt to burn-based messaging without significant friction when burn amounts remain below roughly $0.05 per message. Higher amounts produce noticeable hesitation for casual communication.

Spam Reduction: Burn requirements effectively deter automated spam; the economic cost per message introduces a floor that exceeds the expected return for typical spam campaigns. Systematic measurement across a production deployment at scale is deferred to future empirical work.

Infrastructure Costs: Relay server resource consumption scales approximately linearly with user count in initial testing. Payment proof verification adds modest computational overhead to basic message routing—dominated by blockchain state lookup rather than cryptographic verification.

These observations are informal and from a small-scale deployment. They validate the approach as viable but do not substitute for rigorous performance benchmarking at scale.

9.4 Limitations and Trade-offs

The CashWeb approach involves several inherent trade-offs:

Economic Barriers:While burn amounts are minimal for legitimate users, they may create barriers for users in economically disadvantaged regions. Future work could explore sliding-scale burn rates or alternative value-proof mechanisms.

Cryptocurrency Dependence:The system requires access to cryptocurrency networks, which may limit adoption in regions with restricted access or among users uncomfortable with cryptocurrency.

Recovery Complexity:Key loss scenarios require more complex recovery procedures than traditional centralized systems, potentially creating usability challenges for non-technical users.

9.5 Future Extensions and Applications

The CashWeb protocol provides a foundation for numerous extensions:

Reputation Systems:User reputation scores based on received message feedback could enable dynamic burn rate adjustments and improved spam filtering.

Content Markets:The burn mechanism could extend to content micropayments, enabling new economic models for information sharing and media distribution.

IoT Communication:Machine-to-machine communication with automatic payment could enable new Internet of Things applications where devices pay for bandwidth and processing resources.

Decentralized Social Networks:The publish-subscribe system provides infrastructure for fully decentralized social media platforms with economic incentive alignment.

10 Conclusion

We have presented CashWeb, a comprehensive protocol suite that addresses the fundamental economic incentive problems driving centralization in internet communication systems. Through for-

mal specification of burn-to-speak anti-spam mechanisms, federated infrastructure design, and economic security analysis, we demonstrate that cryptocurrency integration can restore the original vision of decentralized internet communication while providing superior spam resistance compared to existing approaches.

The protocol's emphasis on economic rather than computational or regulatory anti-spam mechanisms provides several advantages: predictable protection guarantees, scalability to global deployment, and resistance to technological arms races that affect computational approaches. Implementation experience with the Stamp social network validates the practical viability of the approach.

Future work should focus on reducing economic barriers for disadvantaged users, improving key management usability, and exploring applications beyond messaging to other communication and coordination problems that suffer from similar centralization pressures.

The ultimate goal of CashWeb is not merely technical innovation, but restoration of user sovereignty over digital communication—enabling a return to the internet's founding principles of decentralization and user empowerment while addressing the economic realities that drove the original centralization.

References

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.

Lotus Development Team. Lotus: A proof-of-work cryptocurrency with native op_return value support. Software release and documentation, 2021. URL https://givelotus.org.

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.