BLACKPAPER

Midnight Messenger

Private messaging where the chain proves who you are, and never sees what you say

Version 1.0

September 2026

CONTENTS

Summary

Most private messengers start by asking for a phone number. That one requirement undoes much of what they promise: a phone number is a real-world identity, and whoever verifies it learns who talks to whom. Messengers that avoid it usually ask for an account on their server instead, which moves the problem rather than removing it.

Midnight Messenger takes a third path. Your identity is a zero-knowledge proof on the Midnight blockchain: you prove you control a key, and nothing else is revealed. Your messages are end-to-end encrypted with MLS, the IETF standard for secure group messaging, and never touch the chain. The two halves are kept apart by design, and neither can be used to rebuild the other.

Signing up takes a username and nothing else. No phone number, no email, no wallet, no tokens. A real on-chain registration happens on your behalf, and you never see it.

---

1 Principles

Five rules shape every decision in this design.

1. **Nothing private goes on-chain.** A public ledger is the worst possible place for private data, so it holds only what must be publicly checkable.
2. **Signing up reveals nothing.** No phone number, no email, no identifying account.
3. **Servers carry ciphertext they cannot read.** Every service is built on the assumption that it could be compromised.
4. **Check it, do not trust it.** Anything that matters can be verified by anyone, on-chain or in the open protocol.
5. **No single operator.** The network is designed to run on infrastructure that many people own, so it does not depend on any one of them.

---

2 The split

On-chain, one contract, per identity:

- a commitment: a hash of a secret and a nonce that only your device holds
- a status: active or revoked
- the public key your device signs with

That is the whole record. No usernames, no profiles, no contacts, no messages, no timestamps of conversations. The contract has no field that could hold them.

Off-chain:

- messages, encrypted end to end with MLS
- relays, which forward ciphertext and briefly hold it for people who are offline
- a directory, which maps a hashed username to a public key

The chain does the one thing a chain is genuinely good at here: letting anyone verify a claim without trusting a third party. It is never used as a database.

---

3 Identity

Everything starts from one secret: a 32-byte seed created on your device and shown to you once as 24 words. Those words are your account. Every key the app uses is derived from them, so the same 24 words restore your identity and your username on a new phone.

Your on-chain identity is a commitment:

```
commitment = persistentHash("midnight-messenger:identity:", nonce, secret)
```

The secret and the nonce are derived from your seed and never leave your device. The chain stores only the commitment. Proving that an identity is yours means producing a zero-knowledge proof that you know what it was built from, which the contract checks without learning it.

The contract has four operations: register, prove control, rotate, and revoke. One deployed contract serves everyone, keyed by commitment.

Rotation replaces one commitment with a new one in a single transaction, revoking the old and activating the new together. Each rotation step is derived from the same seed, so your words recover your current identity no matter how many times you have rotated.

Revocation is permanent. A revoked identity stays in the registry as revoked, so anything reading it checks status, not mere presence.

---

4 Signing up without a wallet

A registration is a real transaction, and a real transaction costs DUST, Midnight's fee resource. Asking a new person to buy tokens before sending their first message would defeat the purpose, so someone else pays.

Your device builds and proves the transaction itself, sealed but unfunded. It hands that to a **sponsor**, which adds the fee from its own DUST and submits it. Your keys never leave your device, the sponsor never learns your secrets, and it cannot change what it is paying for. It can only decline.

The sponsor protects its budget: it pays only for the identity contract's own operations, limits how often any source can ask, rejects replayed requests, and caps what it spends per day.

The economics work. DUST is not spent like cash. It regenerates continuously from NIGHT that is held. Sponsoring signups therefore costs capital that is kept, not money that drains away: a given holding supports a given rate of registrations indefinitely. Free signup is a bounded, predictable cost rather than an open-ended one.

---

5 Finding people

People find each other by name, but a server holding everyone's names is exactly what this design avoids.

So the directory stores `hash(username)` mapped to a public key. If you know a name, you can compute its hash and look it up. The server never sees a name, so it cannot list them.

The hash reuses the contract's own `persistentHash`, domain-separated from the identity commitment so the two can never collide. Claiming a name requires a signature from the identity key being bound, so nobody can claim a name on someone else's behalf. Once taken, a name is never reassigned: a recycled name would let a stranger inherit someone's contacts.

Short and all-number names are reserved.

---

6 Messages

Messages are encrypted with **MLS (RFC 9420)** through a single Rust implementation, compiled to WebAssembly and to native libraries, so every platform runs the same cryptography instead of several versions that must agree.

No room codes, no invitations. Two people who have looked each other up each hold the other's public key. From those two keys alone, both derive the same private room and agree which of them starts the encrypted group. Nothing else is exchanged, and no server decides who talks to whom.

Every handshake is signed. Relays are not trusted, so every handshake message is signed by the sender's identity key and checked against the key the directory holds for the person you chose. The signature covers the room, both names and a timestamp, so it cannot be replayed into another conversation, reused later, or passed off as a different kind of message.

Relays check who is connecting. A relay sends each connection a one-time challenge, which the app signs with its identity key. The room's own name tells the relay who may enter: a private room is a hash of its two members' keys, so only those two can sign in to it. An inbox, where chat requests arrive, can be read only by its owner; anyone else may leave a request and see nothing, not even whether the owner is online. The signature also names the relay it was made for, so one relay cannot replay it at another. None of this needs a database or an account system, which is what lets relays be run by anyone (section 10).

Messages wait for you. When you are offline, the relay holds what arrives, encrypted, and hands it over when you return. It deletes each message the moment your device confirms it has it, and anything never collected expires after seven days. Your device keeps its encryption state, so closing the app never costs you the keys to read what comes next.

Delivery, not reading. A message shows as delivered when the other device has received it. Whether it shows as read is up to the reader: read receipts are a setting, off unless they turn it on.

---

7 Notifications

When a message is waiting for you, your phone shows an alert that says "New message". It never says who sent it or what it says, so neither Apple's notification service nor anyone glancing at your lock screen learns anything about the conversation. Your device registers for alerts only over a connection that has already proven it owns your identity, so nobody can point your alerts at their own phone.

---

8 On your device

Your device holds the only copy of your seed, your conversation keys and your message history. On iOS the seed lives in the Keychain, bound to that device and excluded from backups, and everything else is encrypted with a Keychain key before it touches storage. The app can lock behind Face ID, the app switcher shows a cover instead of your chats, and your recovery phrase hides whenever the screen is being recorded.

Signing out erases the device completely. Your 24 words are the only way back, and nobody, including us, can reset them for you.

---

9 Design trade-offs

Every system makes trade-offs. These are ours, stated plainly.

Timing and pairing. A relay that holds messages learns that some pair of keys exchanged something at some time, though it can read nothing. Deleting on collection keeps that record short, and spreading relays across many operators keeps any one of them from seeing much of it. It is not zero. Apple likewise learns when your phone receives an alert.

A public, permanent registry. Anyone can register, the contract has no delete, and rotation is publicly linked: the old and new commitments appear in the same transaction. An active identity means a registry state, not a reputation or proof that someone is who you think.

Guessable names. Hashing hides the list of usernames, not a name that is easy to guess. Anyone can test whether a common name exists, exactly as they could by trying to message it.

Proof generation. A zero-knowledge proof is computed from the secrets it protects. Wherever a proof is generated, that machine sees them. Proving happens on the device, or on a proof service run by the same operator as the sponsor, and moves fully onto the device as mobile proving matures.

Testnet first. The network launches on Midnight's Preprod testnet. Mainnet follows once the system has been proven there.

---

10 A network nobody owns

A messenger that one company runs can be shut down, pressured or watched by that company. Midnight Messenger is designed to outlive any single operator. This is where the network goes in Phase 3 of the roadmap.

Relays run by node operators. Midnight's node operators already run reliable, always-on infrastructure with public addresses and the skills to run it. A relay is tiny next to a node: it holds encrypted messages for at most a week and deletes them as they are collected. Because a relay only ever sees ciphertext, and checks every connection against the room's own name, an operator learns no more than any relay does.

Your messages wait at your relays. Each person publishes a small set of relays that hold messages for them. To reach you, a sender delivers to your relays; you collect from whichever is up. If an operator disappears, the others still have your messages, and you choose another.

Balancing in the app, not in the middle. There is no load balancer to own or attack. Each app keeps a list of relays, measures them, uses the fastest, and moves when one fails.

Discovery on-chain. Operators and each person's chosen relays are listed on-chain in a registry kept separate from the identity contract, so the network can grow without touching anyone's identity.

One command to run it. Operators get a pre-built, self-contained release and one configuration file. Infrastructure that is hard to run stays centralised whatever its licence says.

The directory becomes a cache. Once the name-to-key binding is itself on-chain, a directory holds nothing anyone has to trust, and anyone can run one.

---

11 Economics

The model, built out in Phase 4 of the roadmap.

Messaging is free. Ordinary signup stays free and wallet-free, because that is the point.

Verified mark. People who want a verified mark next to their name pay a few NIGHT for it. This is the revenue model, rather than charging for messaging.

Premium names. Short and all-number usernames are reserved and offered separately.

Operators earn. Operators sponsor signups for their own users from the DUST their NIGHT already generates, and share in the verified-mark revenue of the people they host. Running a relay alongside a node costs almost nothing and pays.

Abuse resistance. Free, permissionless signup attracts automated accounts. Each registration carries a small proof of work: a moment for a person, real cost for a bot creating thousands, with no captcha, no wallet and no third party. Charging tokens to sign up was considered and rejected, because it would bring back exactly the barrier this design removes.

---

12 Roadmap

Phase 1. Foundation. The identity contract, sponsored registration, hashed usernames, end-to-end encrypted chat with authenticated relays and held messages, on Preprod.

Phase 2. The apps. iOS and Android, with alerts, Face ID lock, recovery by phrase, and read receipts as a choice.

Phase 3. The network. Relays run by node operators, the on-chain relay registry, relays chosen by each app, and the one-command operator release.

Phase 4. The economy. Verified marks, premium names, operator revenue sharing, and proof-of-work signup.

Phase 5. Mainnet.

13 Verify it yourself

The identity contract on Midnight Preprod:

`feab00bb69fc260340b63c0ceeac395b1ee2846b61a13987003ffee20dcb12c3`

preprod.midnightexplorer.com/contracts/feab00bb69fc260340b63c0ceeac395b1ee2846b61a13987003ffee20dcb12c3

Two registrations, each a different device registering itself:

preprod.midnightexplorer.com/transactions/692066ed5c5b7d31fb02258e17dd08a6a1452680ce35cbc4127228fb6730db91

preprod.midnightexplorer.com/transactions/1c585fde80e76363dccb8fa7c56b121b3fe3661eed403aa894774f3af3e9db40

Look at them. What the chain shows is a commitment and a public key. What it does not show is who either person is, and nothing in the contract could.

14 Closing

The claim here is narrow on purpose. This is not a claim to have solved private communication. It is a demonstration that a chain can carry proof of identity without carrying identity, that anyone can be given a real on-chain account without ever holding a token, and that the whole system can be checked by anyone rather than taken on trust.