8 November 2016.

8:15 pm.

Self-custody answers who controls your money.

Not who can see it.

Receiving is the hard part.

Approach	Interaction	On-chain footprint	Payments link?	Cost
Reuse one address	None	None	Yes, all of them	Your privacy
Fresh address per payment	Invoice or xpub server	None	No	Interaction, or trusting a server
BIP-47 payment codes	One-time setup	Notification transaction	Not after setup	A notification per sender
BIP-352 silent payments	None	None	No	Receiver scans every block

BIP-352 moves the cost from interaction to scanning.

This is

Shroud.

A mobile wallet built around BIP-352.

Full send and receive. On by default. iOS and Android.

How a payment finds you.

Receiver publishes

Two public keys. Nothing else.

```
sp1… = (B_scan, B_spend)
```

Sender computes

a = sum of input private keys.

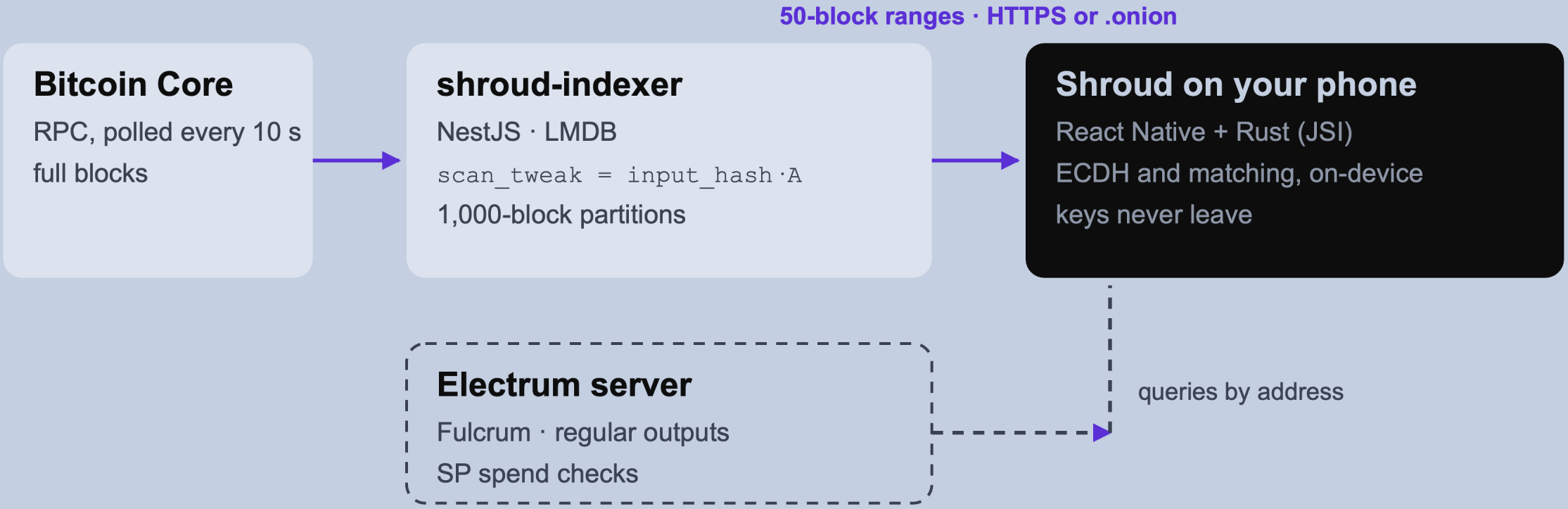
```
A = Σ eligible input pubkeys
input_hash = H(outpoint_L ‖ A)
S = input_hash · a · B_scan
P_k = B_spend + H(S ‖ k) · G
```

Receiver recomputes

Same secret, other side.

```
S = input_hash · b_scan · A
P_k on-chain? It's yours.
```

The architecture.



github.com/CypherCommons/shroud · github.com/CypherCommons/shroud-indexer · built on BlueWallet and Bitshala's silent-pay libraries

Inside the indexer.

Which transactions qualify

- At least one P2TR output
- No input spends a SegWit v2+ output
- Input pubkey extracted per script type; NUMS script paths skipped
- A = sum of input pubkeys; skip if infinity
- scan_tweak = HInputs(outpointL ‖ A) · A

One transaction in a “silent block”

`32 B`	txid
`varint`	n eligible outputs
`44 B × n`	value 8 · x-only pubkey 32 · vout 4
`33 B`	scan_tweak

Immutable once written: a spend never changes it.

Mainnet indexing starts at block 842,578, the height the wallet also uses as its scan floor.

Storage: partitioned LMDB.

Key	Lives in	Value
tx:<txid>	partition	73 B: height, block hash, block time, scan tweak
out:<txid><vout>	partition	40 B: x-only pubkey, value
idx:h:<height><txid>	partition	empty value; drives per-block reads
idx:bh:<hash>	global	block hash → height
idx:bt:<time><height>	global	one per block; timestamp → height
bs: · os: · meta:	global	block state, resume pointer, layout stamp

1,000 heights per partition

parts/NNNNNN, one LMDB environment each. Writes touch only the current one.

8 open, least recently used

Leases stop an environment closing under a live cursor.

The width is stamped

Change partitionBlocks and it refuses to start. That means a full reindex.

Sync, reorgs and crash safety.

1

Resume

Restore the resume pointer, or start at block 842,578 on mainnet. Purge anything above the indexed tip.

2

Trace reorgs

Every 10 s, walk bs: backwards until a stored hash matches the node. Delete the heights that don't.

3

Index

Up to 25 blocks per batch, clamped to a partition boundary. Each block writes its own bs: record.

4

Commit and push

Partitions in ascending order, global last. Then each new silent block goes out over WebSocket.

LMDB has no atomicity across environments. **Commit order is the atomicity:** the global resume pointer never gets ahead of the data.

The API.

Route	Served from	Notes
GET /transactions/range	LMDB	Max 50 blocks. What Shroud uses to sync.
GET /silent-block/height/:h	LMDB	Binary silent block. Also by hash; plus latest-height.
GET /transactions/timestamp-to-height	LMDB	Wallet birthday lookup.
GET /transactions/txid/:txid	Live from the node	Needs txindex=1. 1 h cache, 2 requests a second.
WebSocket	On each commit	Pushes every new silent block.

Burst and sustained throttling, a 5-second response cache, /health exempt. Txids aren't indexed, so the txid route recomputes from the node.

One ECDH per transaction.

```
let S = ecdh(b_scan, scan_tweak); // once per tx
for k in 0..outputs.len() {
let t_k = tagged_hash("BIP0352/SharedSecret", S ‖ k);
for B in [B_spend, B_change_label0] {
let P = B + t_k·G;
if let Some(o) = outputs.get(xonly(P)) { keep(o, t_k) }
}
}
```

Fast path

Rust + libsecp256k1 over JSI, parallel across transactions with rayon.

Change

Tries the main spend key and the label-0 change key.

Spendable

Keeps t_k, outpoint and value. Enough to sign later.

What it costs on a phone.

40 KB

of tweak data per block

4 min

to sync 10,000 blocks on an iPhone 16

Compact tweak data instead of full blocks.

Two keys, two jobs.

Scan key · on the phone

Detects payments.

Can find your outputs. Cannot spend them.

Spend key · on hardware (next)

Signs transactions.

Stays off the phone once hardware support ships.

BIP-352 separates detection from spending. Shroud's hardware design builds on that.

What silent payments don’t buy you.

Leak	What happens	Status
Amount analysis	Values are public; round payments and change amounts tell stories	Splitting makes it harder, not impossible
Sender-side leaks	The sender knows the output they paid, and can reveal it	Cryptography can’t help
Opsec	Screenshots, reused identities, publishing your own history	Still on you
Spending outputs together	Common-input ownership re-links them	Inherent. Spend with care.

What our architecture leaks today.

Where	What happens	What it learns	Mitigation today
Spend checks	To detect spends, the wallet asks Electrum about each found output's address	That server can link your silent payment outputs	Your own Electrum server. Tor hides your IP, not the link.
Wallet birthday	Timestamp-to-height lookup and the first range requested	Indexer learns roughly when the wallet was created	Open
Txid lookups	Checking one transaction asks the indexer for that txid	Indexer learns which transaction you care about	Tor, or don't use it
Network layer	Indexer and Electrum see IP and sync timing	Who and when	Tor via Orbot; Tor-only mode
A dishonest indexer	It could omit tweaks or ranges	You miss payments. It can't learn which are yours or spend them	Run your own: one env variable

Don’t trust. Verify.

MIT licensed

Read every line.

Reproducible builds

Rebuild any release byte-for-byte.

Signed releases

Check what you installed.

github.com/CypherCommons/shroud

What’s shipped, and what’s next.

Shipped

- Silent payments, send and receive
- Tor, custom nodes, payment splitting

Next

- Independent security audit, published in full
- Hardware wallet support

Exploring

- OpenSwap, Ecash, multisig

No promises until the design is sound.

Your keys. Your coins.

One address to rule them all.

Shroud · shroudwallet.com · github.com/CypherCommons/shroud

Get Shroud. Then verify it.

App Store
iOS 16 and later

Google Play
Android 10 and later

shroudwallet.com · github.com/CypherCommons/shroud