Payjoin V2

Oblivious, Serverless, Asynchronous Batching

Dan Gould

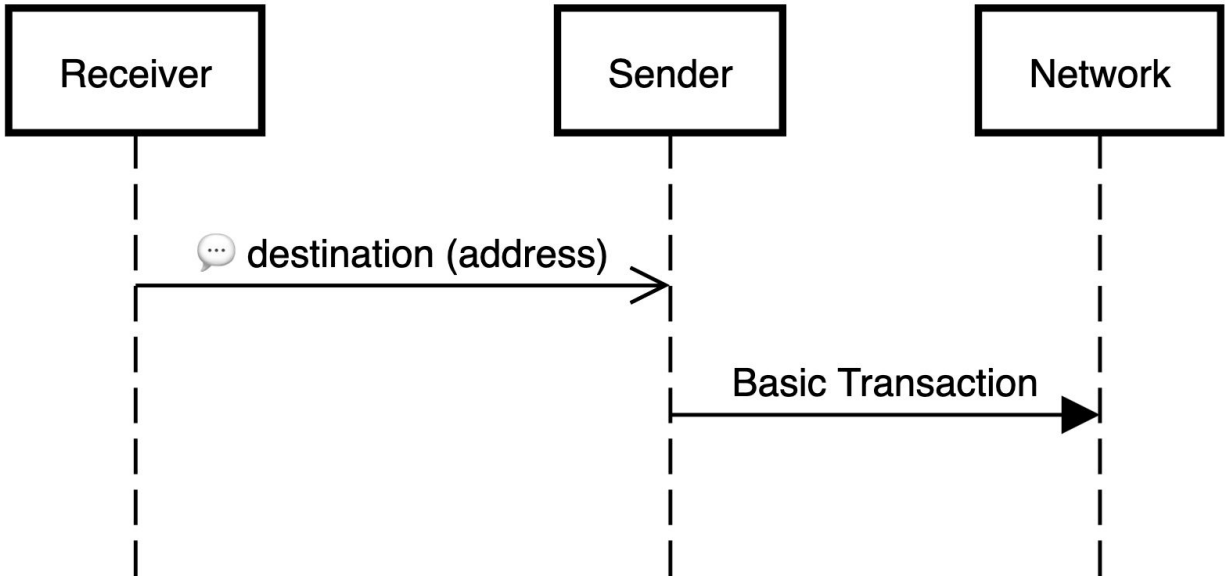
Table of Contents

Why Payjoin?
Bitcoin transactions are naive.

- Bitcoin blockchain is a constrained database
- Batched transactions scale databases the easy way
- Payjoin is simplest way to batch blockchain transactions

How Payjoin Works

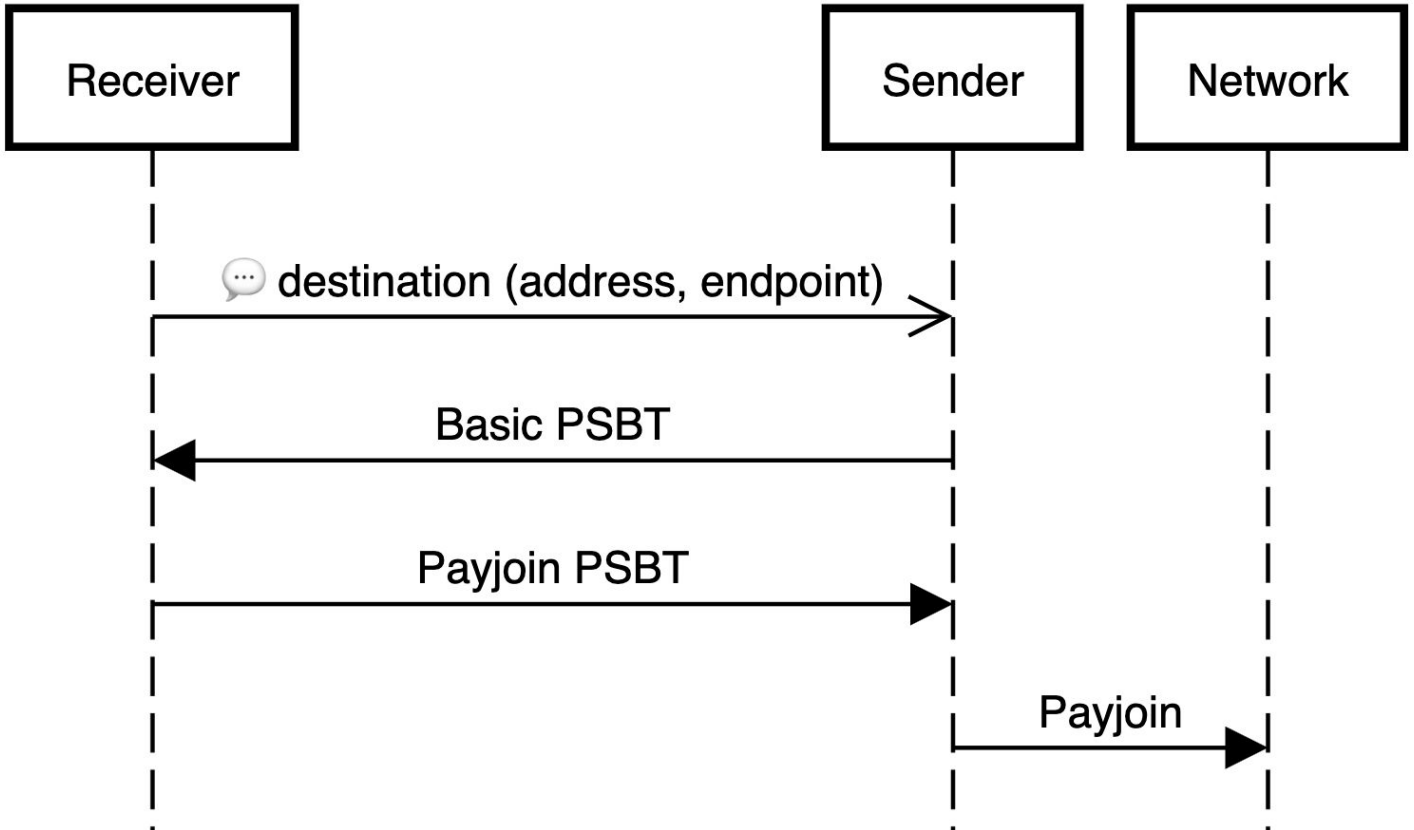
Basic Transaction



Basic Transactions are Basic

- They're costly
  - 1 transaction intent = 1 transaction.
- Their privacy is 💩
  - Inputs necessarily belong to the sender.

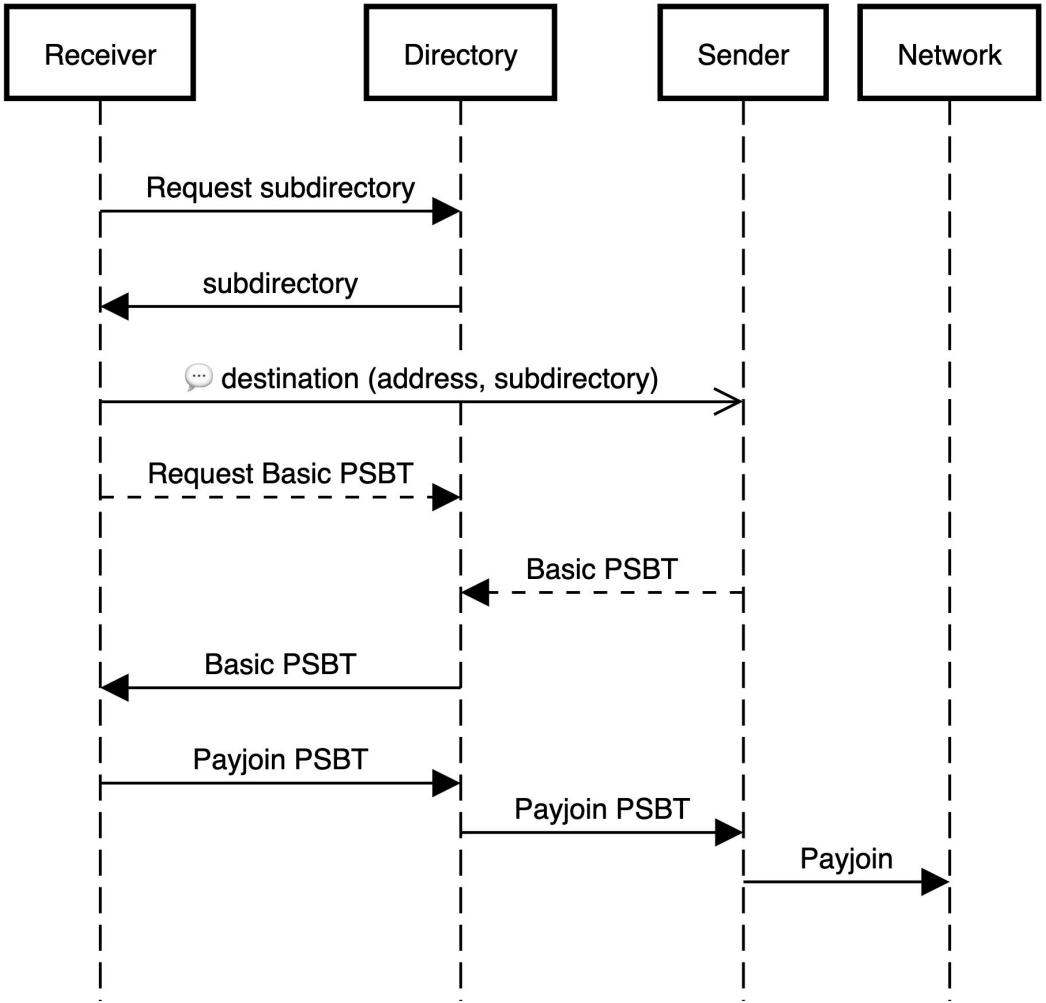
Payjoin V1


Receiver
Sender
Network
destination (address, endpoint)
Basic PSBT
Payjoin PSBT
Payjoin

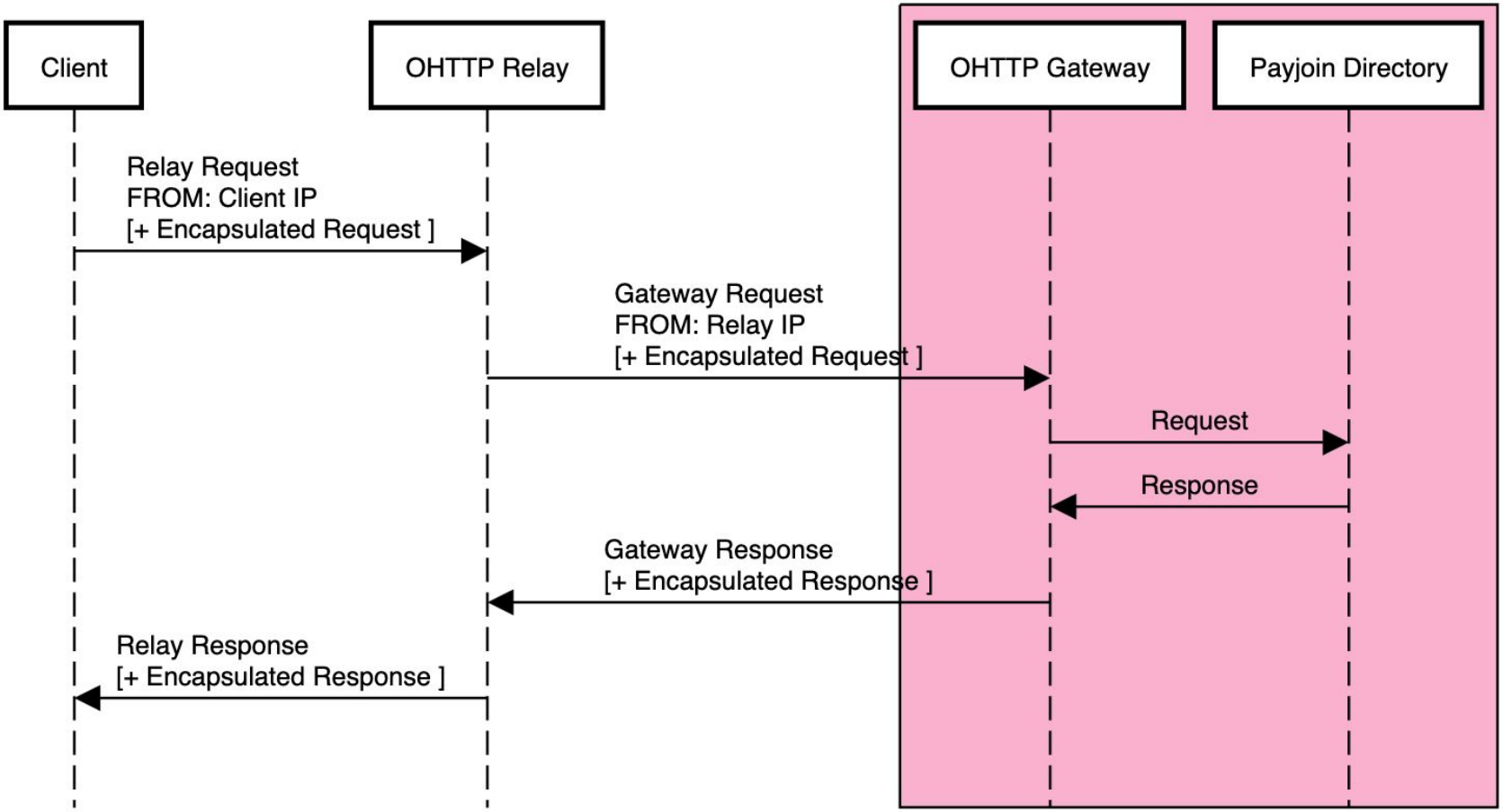
Payjoin V1 is too hard to use

- It’s synchronous
  - Sender and receiver must be online at the same time
- Payjoin V1 needs a static server certificate or Tor
- IP addresses leak everywhere

Payjoin V2


Receiver
Directory
Sender
Network
Request subdirectory
subdirectory
destination (address, subdirectory)
Request Basic PSBT
Basic PSBT
Basic PSBT
Payjoin PSBT
Payjoin PSBT
Payjoin

Oblivious HTTP



Payjoin V2 Ergonomics

Ergonomic Considerations

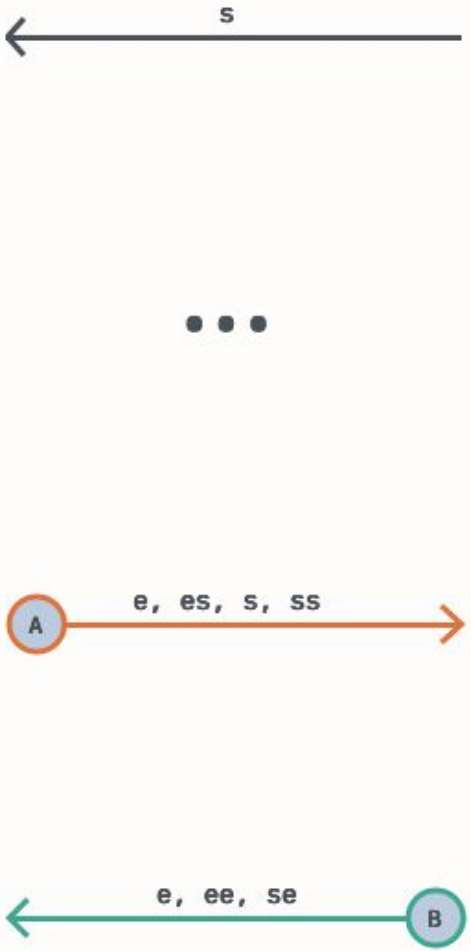
- Sender and Receiver HPKE should use secp256k1
  - An unusual dependency outside of Bitcoin space
- Oblivious HTTP needs bootstrapping
  - Fall back to TLS where it is already available
- Encoding bip21s

Sender & Receiver Authenticated Encryption

Hybrid Pubkey Encryption (HPKE)

- Secp256k1 🆕
- ChaCha20-Poly1305
- SHA-256

No TLS dependency



Oblivious HTTP Bootstrapping: Encrypt to the Directory

- CONNECT method proxy HTTPS-in-HTTP tunnel
- HTTPS-in-WS tunnel to get it from web env like Mutiny
- Cache, like DNS names in Bitcoin Core

There’s a whole Consistency document for this decentralized bootstrapping mechanism

Bitcoin URIs request payjoin (BIP 21)

```
bitcoin:tb1pjqhq9wut7d63xe0khnteylwyc93evl
kkgzw8q5y3nwtepvtfj09supnwrt?amount=1&pj=h
ttps://payjo.in/pk1qw24nx0wyntm7m0r4s7cspn
kdpdteufl5yvw3yzseue0wx6gxum9kemxllf&ohttp
=oh1qyqzpqxu3dz27jcadlvmnk8dty00f783vtd2pe
qjtd0ddn89qrkkf0p4qqzqqqqqqvjm2j8g
```

Bitcoin URIs request payjoin (BIP 21)

- Standard in dozens of apps
- Backwards compatible with v1
- Automatic at payment time
- What encoding to use?
  - base64Url has widespread support
  - Bech32m also has widespread support and case-insensitivity
  - Considerations for UR encoding

Thank you Gordian

Let’s review BIP 77