A future with silent payments

For wallet developers, the biggest challenge with silent payments is supporting the ability to receive. However, there are several projects under active development to help tackle this and a number of wallets that support silent payments today.

Silent payments have massive potential to offer users a solution that is not only convenient but also privacy preserving, a rare combination. For more on the UX benefits, check out "Yashraj's Pocket Guide to Silent Payments - UX edition".

Thanks for reading! That was just a high level overview of silent payments. If you enjoyed it, there's much more to dig into. Ready to keep going? Want to download free copies of this and other zines? Visit

https://satsie.dev/zines

SATSIE'S POCKET GUIDE to SILENT PAYMENTS

A short zine about silent payments: what they are, why they're cool, and how they work

@satsie ☆ https://satsie.dev

SILENT PAYMENTS

Silent payments let you create static, reusable addresses that don't compromise your privacy. They were developed by Ruben Somsen and Josie Baker in March 2023 under BIP352. They are based on stealth addresses by Peter Todd (2014).

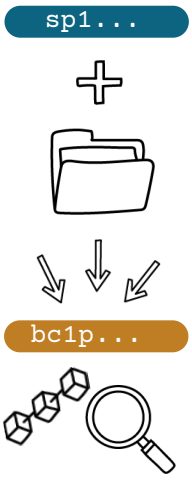
How it works (the basics)

1. Receiver creates a special reusable silent payment address.

2. Sender derives a unique Taproot address for the receiver by combining the silent payment address with a transaction input they plan to use.

3. Sender creates and broadcasts a transaction to the on-chain address.

4. Receiver checks the blockchain for payments to the on-chain address. More on this later!

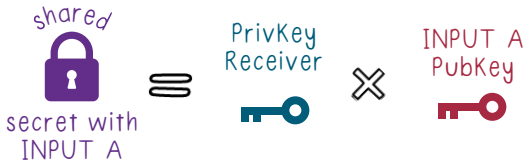


Now both parties have the shared secret. The last step is to add the receiver's public key to it. This results in an on-chain address for the receiver, one that's uniquely determined by a transaction input!

To find payments, the receiver observes the blockchain and uses their scanning key to re-derive addresses that money could have been sent to.

"Easier said than done!!"

This is difficult because the receiver doesn't know which, if any, transactions were meant for them. They must examine every Taproot transaction in every new block. For each transaction input, the receiver computes the shared secret:



Wouldn't it be great if you could reuse bitcoin addresses? What a smooth user experience that would be!

"Actually, you can, but it's disastrous for privacy. Anyone can see who paid you, how much, and when. Address reuse compromises you AND the parties you transact with."

Since address reuse is such a bad idea, most users get a new address from their wallet each time they need to receive a payment. At first it may not seem like much, but the extra effort adds up. Bob can't just pay Alice whenever he wants, he has to ask her for an address and wait for her response.

There have been a number of proposals to improve this pain point. A particularly interesting one is...

Unlike its predecessors, silent payments require little to no interaction between sender and receiver.

The silent payment address never shows up on the blockchain and the transaction looks like any other Taproot payment. One silent payment address can be reused an infinite number of times by an infinite number of senders!

What's the trade off?

The receiver doesn't know any of the on-chain addresses that senders derive. Receivers need additional computation to scan the blockchain and re-derive all potential on-chain addresses.

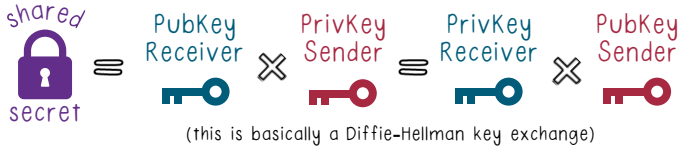
A deeper dive

Receivers start by generating 2 public/private key pairs: a spending keypair and a scanning keypair. The public keys are combined to create the silent payment address. They are also used by senders to derive unique on-chain addresses for the receiver.

The first step to doing this is to use the public key information from the silent payment address to create a shared secret.

A shared secret?

Yes! The sender and receiver share a secret value that no one else knows. Here's how it's made:



(this is basically a Diffie-Hellman key exchange)

Recall that silent payments are non-interactive and the sender and receiver never exchange keys. The shared secret is only possible because the sender's public/private keypair comes from one of the transaction inputs. Once a transaction is broadcast, the public keys for all the inputs are revealed. That means before the sender derives the on-chain address, they must select the transaction inputs (aka coins) that they plan to use. It also means the receiver cannot know the sender's public key until the transaction is broadcast.

After adding the shared secret to the receiver's public key, the receiver checks if the resulting on-chain address matches any of the transaction outputs.

About that trade off...

All of this is significantly more effort when compared to BIP32 HD wallets. Instead of deriving a list of addresses and checking for matches in the transaction outputs, silent payments involve looking at the public key of every input and performing elliptic curve multiplication to get the shared secret. This is why it takes a lot more work for receivers to find their incoming silent payments.

If a wallet's node is up-to-date, the additional work to scan a new block every 10 minutes isn't a big deal. There are also shortcuts that can be taken, like ignoring dust and transactions from before the wallet was created.