XRPL Privacy and Confidential Transactions

→ To submit transactions to the XRP Ledger, you need a way to digitally sign them without compromising the security of your secret keys. (If others gain access to your secret keys, they have as much control over your accounts as you do, and can steal or destroy all your money.) This page summarizes how to set up such an environment so you can sign transactions securely.

→ There are several configurations with varying levels of security that may be acceptable for your situation. Choose one of the following that best fits your needs:

Run rippled locally, or in the same LAN.

Use a client library that can do local signing.

Use a dedicated signing device that supports XRP Ledger signatures.

Use a secure VPN to connect to a remote rippled machine you trust.

Insecure configurations

→ Any configuration in which outside sources may gain access to your secret key is dangerous, and is likely to result in a malicious user stealing all your XRP (and anything else your XRP Ledger address has). Examples of such configurations include ones where you use the sign method of someone else's rippled server over the internet, or you send your secret key in plain text over the internet to your own server.

Run rippled Locally

→ In this configuration, you run rippled on the machine that generates the transactions. Since the secret key never leaves your machine, no one without access to your machine can get access to the secret key. You should, of course, follow industry-standard practices for securing your machine. To use this configuration

1. Install rippled.

Be sure that your local machine meets the minimum system requirements for rippled.

2. When you need to sign transactions, connect to your server on localhost or 127.0.0.1. Use the sign method (for single signatures) or sign_for method (for multi-signatures).

3. Maintain the server to keep it running, updated, and in sync with the network while you're using it.

Run rippled on the same LAN

→ In this configuration, you run a rippled server on a dedicated machine in the same private local area network (LAN) as the machine that generates the transactions to be signed. This configuration

lets you assemble transaction instructions on one or more machines with very modest system specs, while using a single dedicated machine for running rippled. This may appeal to you if you run your own datacenter or server room.

→ To use this configuration, set the rippled server to accept wss and https connections within your LAN. You can use a self-signed certificate if you use certificate pinning , or you can use a certificate signed by an in-house or well-known Certificate Authority. Some certificate authorities, such as Let's Encrypt issue certificates automatically for free.

Security best practices for signing libraries

→ Make sure the signing library you use has properly and securely implemented its signing algorithm(s). For example, if the library uses the default ECDSA algorithm, it should also use deterministic nonces as described in RFC-6979 .

All of the published libraries listed above follow industry best practices.

Keep your client library updated to the latest stable version.

For enhanced security, you can load your secret keys from a management tool such as Vault.

Local signing examples

→ // Sample code demonstrating secure offline signing using xrpl.js library.

```
const xrpl = require('xrpl')


// Load seed value from an environment variable:
const my_wallet = xrpl.Wallet.fromSeed(process.env['MY_SEED'])


// For offline signing, you need to know your address's next Sequence number.
// Alternatively, you could use a Ticket in place of the Sequence number.
// This is useful when you need multiple signatures and may want to process transactions out-of-order.
// For details, see: https://xrpl.org/tickets.html
let my_seq = 21404872


// Provide *all* required fields before signing a transaction
const txJSON = {
  "Account": my_wallet.address,
```

```
  "TransactionType":"Payment",
  "Destination":"rf1BiGeXwwQoi8Z2ueFYTEXSwuJYfV2Jpn",
  "Amount":"13000000",
  "Flags":2147483648,
  "LastLedgerSequence":7835923, // Optional, but recommended.
  "Fee":"13",
  "Sequence": my_seq
}

const signed = my_wallet.sign(txJSON)
console.log("tx_blob is:", signed.tx_blob)
console.log("tx hash is:", signed.hash)
```

Use a dedicated signing device

→ Some companies sell dedicated signing devices, such as the Ledger Nano S , which are capable of signing XRP Ledger transactions using a secret key that never leaves the device. Some devices may not support all types of transactions.

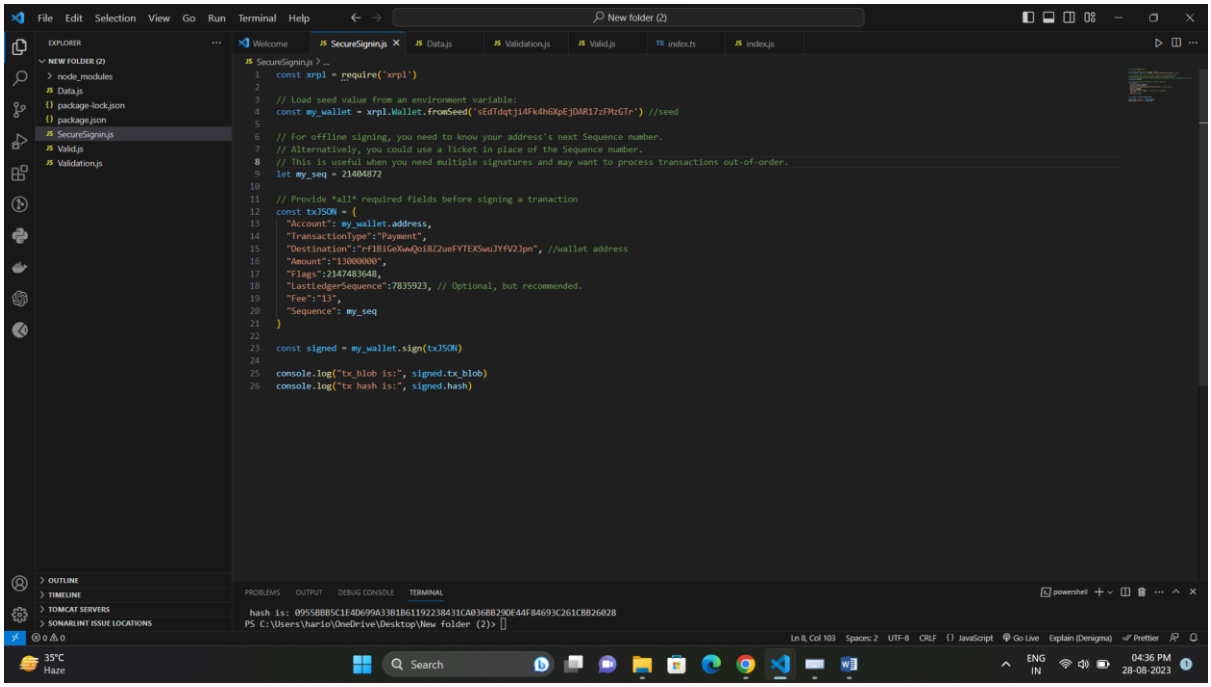
→ Setting up this configuration depends on the specific device. You may need to run a "manager" application on your machine to interact with the signing device. See the manufacturer's instructions for how to set up and use such a device.

Use a secure VPN with a remote rippled server

→ This configuration uses a rippled server hosted remotely, such as in a colocation facility or a distant datacenter, but connects to it securely using an encrypted VPN.

→ To use this configuration, follow the steps for running rippled on a private LAN, but use a VPN to connect to the LAN of the remote rippled server. Instructions for setting up the VPN are specific to your environment and are not described in this guide.

Code snippets



Output