Crypto Wallets

□□ Crypto Wallets — Chapter Description

This chapter documents the setup, use, and maintenance of cryptocurrency wallets, with a focus on self-custody, security, and interoperability across devices and nodes.

It covers both software wallets (e.g. Sparrow Wallet, Ledger Live) and hardware wallets (e.g. Coldcard, Ledger Stax, Trezor Model T), including their integration with personal infrastructure such as full nodes.

The emphasis is on maintaining control over private keys, ensuring verifiable and reproducible setups, and documenting practical workflows such as transactions, updates, and recovery procedures.

Topics include wallet installation and upgrades, hardware wallet pairing, node connectivity, backup and recovery strategies, and security considerations.

This chapter reflects a self-sovereign approach to digital assets, prioritising open-source tools, local verification, and minimal trust in third parties.

Ledger Live AppImage on Linux Mint — Persistent Launcher & Update Workflow

? Objective

Create and maintain a stable Ledger Live setup on Linux Mint that:

- uses a **custom persistent icon**
- survives **log out / reboot**
- continues to work after **AppImage updates**
- avoids repeated manual configuration

? Overview

Ledger Live is run as an AppImage stored in a fixed location, with a `.desktop` launcher pointing to:

- a **static AppImage filename**
- a **separate, permanent icon file**

This ensures updates only affect the binary, not the launcher or icon.

?? Folder Structure

```
~/AppImages/
├── ledger-live.AppImage
└── icons/
└── ledger.png
```

- `ledger-live.AppImage` → replaced on updates
- `ledger.png` → permanent icon (never changes)

? Desktop Launcher

Location:

~/Desktop/ledger-live.desktop

Contents:

[Desktop Entry]
Type=Application
Name=Ledger Live
Exec=/home/coolbaron/AppImages/ledger-live.AppImage
Icon=/home/coolbaron/AppImages/icons/ledger.png
Terminal=false
Categories=Finance;

? Key Principle

The setup separates responsibilities:

- **AppImage = replaceable component (software)**
- **Icon = static asset (visual identity)**
- **Launcher = fixed reference (glue)**

? Update Workflow

1. Download new version

Saved to:

~/Downloads/ledger-live-desktop-<version>.AppImage

2. Verify download (recommended)

ls -lh ~/Downloads | grep ledger

Expected size: ~150MB+

3. Replace existing AppImage (safe method)

mv ~/Downloads/ledger-live-desktop-2*.AppImage ~/AppImages/ledger-live.AppImage
chmod +x ~/AppImages/ledger-live.AppImage

4. Test before use

~/AppImages/ledger-live.AppImage

?? Icon Handling

Icon location:

/home/coolbaron/AppImages/icons/ledger.png

? Behaviour

- Not affected by updates
- Not tied to Ledger Live version
- No need to re-download

? Do NOT

- delete or move the icon
- change the icon path

? Auto vs Manual Updates

? In-App Auto Update Behaviour

When clicking **“Update” inside Ledger Live**:

- downloads a new AppImage (usually to `~/Downloads` or `/tmp`)
- launches the new version
- does **not reliably**:
  - replace existing AppImage
  - preserve filename
  - remove older versions

?? Impact on This Setup

Your launcher points to:

Exec=/home/coolbaron/AppImages/ledger-live.AppImage

If auto-update downloads a new version:

- Desktop shortcut → still opens old version
- New version → only runs manually
- Multiple AppImages may accumulate

? Why Auto Update Is Not Ideal

- breaks single-file structure
- introduces version drift
- creates duplicate binaries

- reduces predictability

? Recommended Approach — Manual Update

mv ~/Downloads/ledger-live-desktop-2*.AppImage ~/AppImages/ledger-live.AppImage
chmod +x ~/AppImages/ledger-live.AppImage

? Benefits

- deterministic behaviour
- single source of truth
- launcher always correct
- no duplicates

? Optional Hybrid Approach

1. Allow Ledger Live to download update
2. Then manually replace:

ls -lh ~/Downloads | grep ledger
mv ~/Downloads/ledger-live-desktop-2*.AppImage ~/AppImages/ledger-live.AppImage
chmod +x ~/AppImages/ledger-live.AppImage

?? Common Pitfalls

Corrupted AppImage (e.g. 75 bytes)

Symptoms:

ls -lh ~/AppImages/ledger-live.AppImage

Shows very small file size

➡ Cause:

- incomplete download
- incorrect wildcard match

➡ Fix:

- delete file
- re-download
- move explicitly

Duplicate AppImages

ledger-live-desktop-2.x.x.AppImage

➡ Remove duplicates and keep:

ledger-live.AppImage

Launcher failure

Error:

> “There was an error launching the application”

➡ Cause:

- broken AppImage

➡ Fix:

- replace AppImage

? Verification Commands

ls -lh ~/AppImages/ledger-live.AppImage
ls -l ~/AppImages/icons/ledger.png
cat ~/Desktop/ledger-live.desktop

? Troubleshooting

chmod +x ~/Desktop/ledger-live.desktop
cinnamon --replace &

? Maintenance Notes

- Only one AppImage should exist
- Icon remains unchanged indefinitely
- Launcher requires no further edits

? Conclusion

This setup provides a clean, reliable, and controlled way to run Ledger Live on Linux Mint.

By separating:

- executable (AppImage)
- icon (static asset)
- launcher (fixed reference)

you achieve a **stable, repeatable, low-maintenance workflow**.

? Update Command (Final)

```
mv ~/Downloads/ledger-live-desktop-2*.AppImage ~/AppImages/ledger-live.AppImage
chmod +x ~/AppImages/ledger-live.AppImage
```

? Guiding Principle

> One AppImage. One path. One launcher.

? SHA256 Verification

? Objective

Verify that the downloaded Ledger Live AppImage has not been corrupted or altered before replacing the existing binary.

This adds a simple integrity check to the update workflow.

? Why This Matters

An AppImage is a standalone executable.

Before moving it into the permanent location, it is good practice to verify:

- the download completed properly
- the file is not corrupted
- the checksum matches the official release value

This is especially useful when manually managing binaries.

? Step 1 — Generate the local SHA256 hash

Run:

sha256sum ~/Downloads/ledger-live-desktop-2*.AppImage

Example output:

abcd1234... /home/coolbaron/Downloads/ledger-live-desktop-2.145.0-linux-x86_64.AppImage

---

? Step 2 — Compare with the official hash

Check the SHA256 value published by Ledger for that exact version and verify that:

- the version matches
- the filename matches
- the hash matches exactly

If the hashes differ, do **not** use the file.

?? Important

A valid file size alone is not enough.

Example:

- correct size → suggests a complete download
- matching SHA256 → confirms integrity

Both checks are useful, but SHA256 is the stronger verification.

? Recommended Verification Flow

1. Check file size

ls -lh ~/Downloads | grep ledger

2. Check SHA256

sha256sum ~/Downloads/ledger-live-desktop-2*.AppImage

3. Compare with official release hash

4. Move only after verification

mv ~/Downloads/ledger-live-desktop-2*.AppImage ~/AppImages/ledger-live.AppImage
chmod +x ~/AppImages/ledger-live.AppImage

? If the Hash Does Not Match

Do not move the file into `~/AppImages`.

Instead:

rm ~/Downloads/ledger-live-desktop-2*.AppImage

Then download it again and repeat the check.

? Practical Note

For routine use, checking file size may be enough to catch obvious failures such as the earlier **75-byte corrupted AppImage**.

For stronger assurance, especially with wallet-related software, SHA256 verification is the better habit.

? Guiding Principle

> Verify first. Replace second.