

Crypto Management

☐ 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](#)
- [Ledger Live Update \(Linux Mint\) Quick Guide](#)
- [☐ SHA256 Verification](#)
- [Start9 Electrs Startup Failure After StartOS Upgrade](#)

Ledger Live ApplImage 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 **ApplImage updates**
 - avoids repeated manual configuration
-

📋 Overview

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

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

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

⚙️ Folder Structure

```
~/ApplImages/  
├─ ledger-live.ApplImage  
├─ 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>.ApplImage

2. Verify download (recommended)

ls -lh ~/Downloads | grep ledger

Expected size: ~150MB+

3. Replace existing ApplImage (safe method)

```
mv ~/Downloads/ledger-live-desktop-2*.ApplImage ~/ApplImages/ledger-live.ApplImage  
chmod +x ~/ApplImages/ledger-live.ApplImage
```

4. Test before use

~/ApplImages/ledger-live.ApplImage

Icon Handling

Icon location:

✓ 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 Applimages 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*.ApplImage ~/Applimages/ledger-live.ApplImage  
chmod +x ~/Applimages/ledger-live.ApplImage
```

✓ 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*.ApplImage ~/ApplImages/ledger-live.ApplImage
chmod +x ~/ApplImages/ledger-live.ApplImage
```

Common Pitfalls

Corrupted ApplImage (e.g. 75 bytes)

Symptoms:

```
ls -lh ~/ApplImages/ledger-live.ApplImage
```

Shows very small file size

→ Cause:

- incomplete download
- incorrect wildcard match

→ Fix:

- delete file
 - re-download
 - move explicitly
-

Duplicate ApplImages

```
ledger-live-desktop-2.x.x.ApplImage
```

→ Remove duplicates and keep:

```
ledger-live.ApplImage
```

Launcher failure

Error:

“There was an error launching the application”

→ Cause:

- broken ApplImage

→ Fix:

- replace ApplImage
-

Verification Commands

```
ls -lh ~/ApplImages/ledger-live.ApplImage
```

```
ls -l ~/ApplImages/icons/ledger.png
```

```
cat ~/Desktop/ledger-live.desktop
```

Troubleshooting

```
chmod +x ~/Desktop/ledger-live.desktop
```

```
cinnamon --replace &
```

Maintenance Notes

- Only one ApplImage 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 (ApplImage)
- icon (static asset)
- launcher (fixed reference)

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

□□ Update Command (Final)

```
mv ~/Downloads/ledger-live-desktop-2*.ApplImage ~/ApplImages/ledger-live.ApplImage
chmod +x ~/ApplImages/ledger-live.ApplImage
```

□□ Guiding Principle

“ One ApplImage. One path. One launcher.

Ledger Live Update (Linux Mint) Quick Guide

1. Download into the same folder

- `ledger-live-desktop-*-linux-x86_64.AppImage`
- `ledger-live-desktop-*.sha512sum`
- `ledger-live-desktop-*.sha512sum.sig`
- `ledgerlive.pem`

2. Verify Ledger's signature

```
openssl dgst -sha256 -verify ledgerlive.pem \  
-signature ledger-live-desktop-*.sha512sum.sig \  
ledger-live-desktop-*.sha512sum
```

Expected output:

```
Verified OK
```

3. Verify the Linux AppImage

```
grep "linux-x86_64.AppImage" ledger-live-desktop-*.sha512sum | sha512sum -c
```

Expected output:

```
ledger-live-desktop-*-linux-x86_64.AppImage: OK
```

4. Replace the old AppImage

Single-line version:

```
mv ledger-live-desktop-*-linux-x86_64.AppImage ~/AppImages/ledger-live.AppImage
```

Split-line version:

```
mv ledger-live-desktop-*-linux-x86_64.AppImage \  
~/AppImages/ledger-live.AppImage
```

5. Make it executable

```
chmod +x ~/AppImages/ledger-live.AppImage
```

6. Launch Ledger Live

```
~/AppImages/ledger-live.AppImage
```

Notes

- The `*` wildcard means you don't need to edit the version number for each release.
- The wildcard works as long as there is only **one** Ledger Live download in your `Downloads` folder.
- If older versions are still present, remove them first or specify the version explicitly.
- When splitting a command across two lines, the backslash (`\`) **must be the final character on the first line**, with **no spaces after it**.

☐☐ SHA256 Verification

☐☐ Objective

Verify that the downloaded Ledger Live ApplImage 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 ApplImage 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*.ApplImage
```

Example output:

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

□ 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*.ApplImage
```

3. Compare with official release hash

4. Move only after verification

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

❑ If the Hash Does Not Match

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

Instead:

```
rm ~/Downloads/ledger-live-desktop-2*.ApplImage
```

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 ApplImage**.

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

📋 Guiding Principle

Verify first. Replace second.

Start9 Electrs Startup Failure After StartOS Upgrade

****Date:**** 25 July 2026

Summary

After upgrading StartOS, the ****Electrs**** service failed to start on ****both**** Start9 servers.

The issue was ****not**** related to the Bitcoin blockchain, Electrs index, SSD, hardware or network. The root cause was a configuration conflict introduced during the migration process.

Affected Systems

Server 1

- Hardware: AMD Ryzen 7 5825U
- Bitcoin implementation: Bitcoin Knots
- Electrs version: 0.11.1

Server 2

- Hardware: Intel Celeron N4505
- Bitcoin implementation: Bitcoin Core
- Electrs version: 0.11.1

Both systems exhibited the identical failure immediately after the StartOS upgrade.

Symptoms

Electrs repeatedly exited during startup.

The log reported:

...

Error: ambiguous configuration - auth and cookie_file can't be specified at the same time

...

Investigation

The Electrs configuration file was located at:

```

/media/startos/data/package-data/volumes/electrs/data/main/electrs.toml

```

Its contents were:

```toml

cookie\_file = "/mnt/bitcoind/.cookie"

daemon\_rpc\_addr = "10.0.3.1:63174"

daemon\_p2p\_addr = "10.0.3.1:58334"

network = "bitcoin"

electrum\_rpc\_addr = "0.0.0.0:50001"

log\_filters = "INFO"

auth = "bitcoin:xxxxxxxxxxxxxxxxx"

```

Electrs accepts **either**

- cookie authentication

or

- username/password authentication

but **not both simultaneously**.

The migration left both options in the configuration.

Resolution

Create a backup:

```bash

sudo cp \

/media/startos/data/package-data/volumes/electrs/data/main/electrs.toml \

/media/startos/data/package-data/volumes/electrs/data/main/electrs.toml.backup

```

Remove only the `auth` line:

```
```bash
sudo sed -i '/^auth = /d' \
/media/startos/data/package-data/volumes/electrs/data/main/electrs.toml
```
```

Verify:

```
```bash
sudo nl -ba \
/media/startos/data/package-data/volumes/electrs/data/main/electrs.toml
```
```

Correct result:

```
```toml
cookie_file = "/mnt/bitcoind/.cookie"
daemon_rpc_addr = "10.0.3.1:63174"
daemon_p2p_addr = "10.0.3.1:58334"
network = "bitcoin"
electrum_rpc_addr = "0.0.0.0:50001"
log_filters = "INFO"
```
```

Restart the Electrs service from the StartOS interface.

Electrs started immediately without rebuilding the index.

Root Cause

A StartOS migration bug appears to leave the legacy

```
```toml
auth = "bitcoin:..."
```
```

entry in `electrs.toml` while simultaneously enabling

```
```toml
cookie_file = "/mnt/bitcoind/.cookie"
```
```

This produces an invalid Electrs configuration.

Evidence

- Reproduced on **two independent Start9 servers**
- Different hardware platforms
- Different Bitcoin implementations (Core and Knots)
- Identical configuration conflict
- Identical error message
- Identical fix

This strongly indicates a migration bug rather than a hardware or installation issue.

Lessons Learned

- Do **not** delete the Electrs database.
- Do **not** resync Bitcoin.
- Do **not** rebuild the Electrs index.

Always inspect:

```

/media/startos/data/package-data/volumes/electrs/data/main/electrs.toml

```

If both `cookie\_file` and `auth` are present, remove only the `auth` entry.

Recommendation

Report the issue to the Start9 developers with:

- StartOS version
- Electrs version
- Exact error message
- Original `electrs.toml`
- Corrected `electrs.toml`
- Confirmation that the issue occurred on two separate servers and was resolved by removing the `auth` line.