Satnam NFC Provisioning Guide

NTAG424 NFC Physical MFA • Privacy-first • Zero-knowledge

> **Goal**: Provision your NTAG424 tag securely for Satnam NFC Physical MFA without a separate hardware bridge. All verification is performed by Satnam's Netlify Functions using local SUN/CMAC checks.

1) Provision with Boltcard (Android only)

Single Path

Android (Boltcard Programming App)

- Android phone with NFC
- App: Boltcard NFC Programming App
- Fastest and officially supported path

Download:

- Releases (APK): https://github.com/boltcard/bolt-nfc-android-app/releases
- Play Store: https://play.google.com/store/apps/details?id=com.lightningnfcapp

> **Next step:** After provisioning your Name Tag, return to the Satnam landing page and click *"Register/Signin Your Name Tag"* to complete registration/authentication.
> **Reference:** External Boltcard setup guide: ereignishorizont.xyz/en/boltcard_en/ (https://ereignishorizont.xyz/en/boltcard_en/)

> **iOS users**: Borrow an Android device or visit a community provisioning station to program your tag. After programming, your tag works with iOS for tapping/verification.

> **Security critical**: Never share your provisioning blob (K0/K1/K2, SDM config) outside your secure device. Store it in a password manager. If you lose the keys after changing them on the tag, you will not be able to re-program that tag.

2) Get your provisioning blob from Satnam

In Satnam: Security → NFC Physical MFA → "Provision new tag". This calls `/nfc-unified/initialize` and provides a JSON blob (client-side only) with:

- `url_base` — the NDEF URL to write (e.g., https://www.satnam.pub/nfc/scan)

- `k0`, `k1`, `k2` — AES keys (hex)
- `sdm` — SUN/SDM enable + offsets (PICC/CMAC insertion)

2) Provisioning steps (Boltcard Android)

1. Install the Boltcard app (APK/Play) and enable NFC.
2. Open *Key change* → apply `K0/K1/K2` exactly as provided by Satnam (from the provisioning blob).
3. Enable **SDM/SUN** and set offsets per the blob so that `PICC` (UID) and `CMAC` are appended to the URL.
4. Set the NDEF URL to your base: `https://www.satnam.pub/t/<duid>`. The SDM parameters will be appended on read as `?sdm=...&u=...`.
5. Optional: Enable UID randomization for privacy (irreversible).
6. Hold the tag still on the phone until programming succeeds.

4) Verify your tag

- **Quick**: Tap with any NFC phone. The opened URL should include dynamic SDM params (e.g., CMAC/PICC).
- **Satnam end-to-end**: Tap → app opens Satnam → frontend calls `/nfc-unified/verify` with SDM fields → success response.
- **Low-level**: Use TagXplorer to confirm keys, SDM enable, and NDEF URL record.

5) How it integrates (at a glance)

```
User Tap → Phone opens NDEF url_base
   → Frontend captures SDM params (PICC/CMAC)
   → Netlify Function /nfc-unified/verify
   → SUN/CMAC verification via Netlify function (hardware bridge enforced when conf
   → Session/auth success
   → (Optional) LNbits Boltcards used for wallet/registry metadata only
```

6) Troubleshooting

- **Write failed / Tag moved**: Keep the tag fully still; retry. Ensure phone NFC coil aligns with tag antenna.
- **CMAC missing in URL**: SDM not enabled or offsets incorrect. Re-apply SDM config per blob.
- **Cannot re-program**: Keys changed but lost. The tag cannot be re-provisioned without correct keys.

- **Desktop reader issues**: Use PN532 or an NXP reader with NTAG424 support; avoid older ACR122U.
- **Android errors**: Reinstall app; ensure device NFC is on and no other NFC app is interfering.

7) Best practices

- Provision in a controlled environment (no unknown NFC devices nearby).
- Store the provisioning blob in a password manager; never email or chat it.
- Test with a spare tag first; record success logs from `/nfc-unified/verify`.