Building a cryptocurrency wallet in React Native

Ололо пыщь пыщь

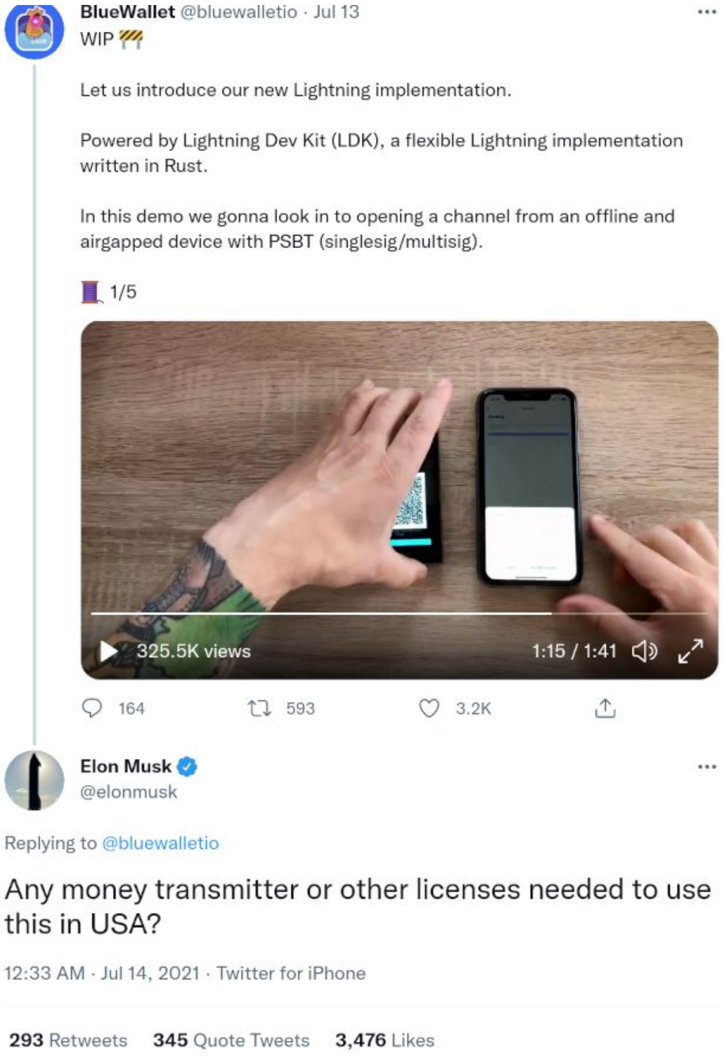
ver 1.0
Igor Korsakov / bluewallet.io / London / January, 2023

Plan

- About me
- What is a wallet?
- Importance of entropy
- Secure storage
- Bitcoinjs and crypto libs in general
- Network requests
- Dependencies are a liability
- etc

About

- Started BlueWallet in 2017
- One of the first cryptocurrency wallets built with RN



BlueWallet @bluewalletio · Jul 13

WIP 🚧

Let us introduce our new Lightning implementation.

Powered by Lightning Dev Kit (LDK), a flexible Lightning implementation written in Rust.

In this demo we gonna look in to opening a channel from an offline and airgapped device with PSBT (singlesig/multisig).

🧵 1/5

164 593 3.2K

Elon Musk
@elonmusk

Replying to @bluewalletio

Any money transmitter or other licenses needed to use this in USA?

12:33 AM · Jul 14, 2021 · Twitter for iPhone

293 Retweets **345** Quote Tweets **3,476** Likes

What is a wallet?

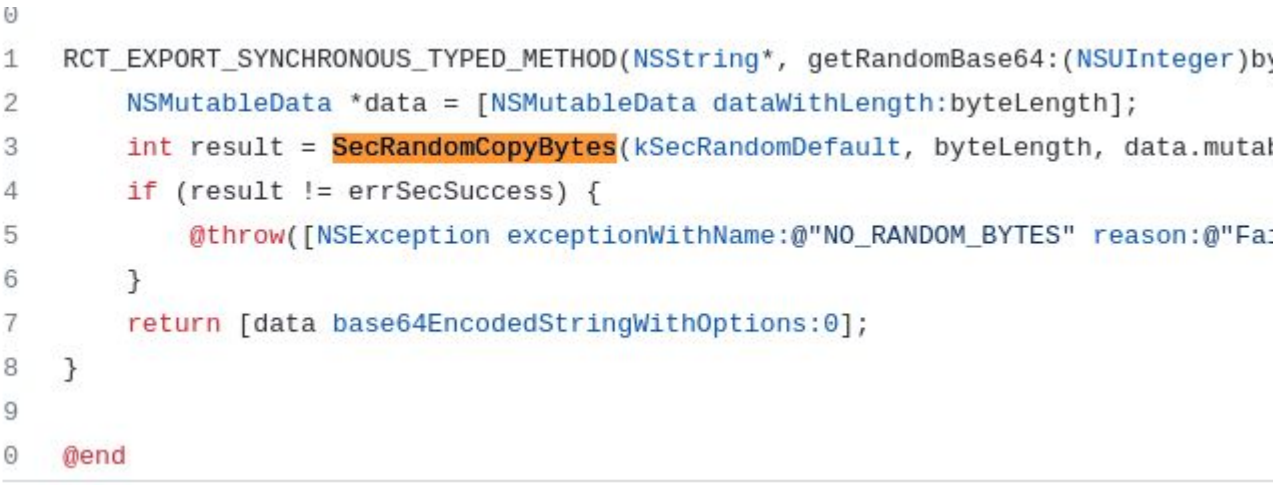
- Acquire good-quality entropy
- Safeguard this entropy
- Do operations with this entropy (usually digital signature)

Entropy

- Math.random() - NO!
- CSPRNG - YES!

- Entropy ~= randomness ~= private key ~= mnemonic seed phrase
- 2^256 - 1 and 77 zeroes
- Example: 0x3705005a6896d9814a1a28271cde62d247408957940478c8ce7c3ef3d3d1a017

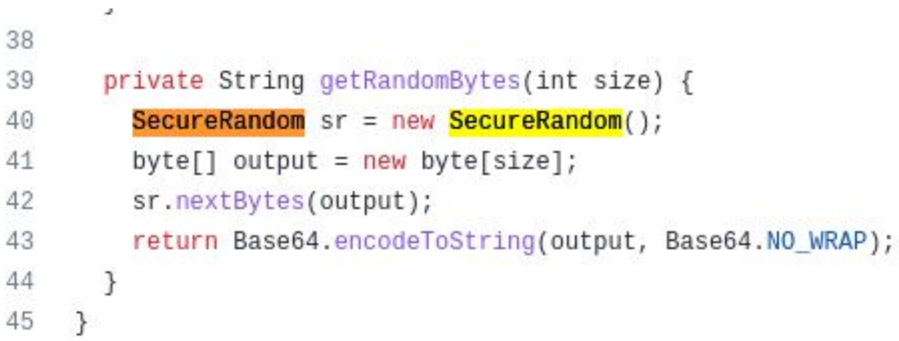
iOS



```
RCT_EXPORT_SYNCHRONOUS_TYPED_METHOD(NSString*, getRandomBase64:(NSUInteger)by
    NSMutableData *data = [NSMutableData dataWithLength:byteLength];
    int result = SecRandomCopyBytes(kSecRandomDefault, byteLength, data.mutab
    if (result != errSecSuccess) {
        @throw([NSException exceptionWithName:@"NO_RANDOM_BYTES" reason:@"Fa
    }
    return [data base64EncodedStringWithOptions:0];
}

@end
```

Android



```java
  }

  private String getRandomBytes(int size) {
    SecureRandom sr = new SecureRandom();
    byte[] output = new byte[size];
    sr.nextBytes(output);
    return Base64.encodeToString(output, Base64.NO_WRAP);
  }
}
```

Secure storage

- Keystore/keychain
- Encrypted database - AES-256 encryption key comes from CSPRNG and is stored in Keystore

Libraries

- Bitcoinjs (https://github.com/bitcoinjs)
- Example: BIP39 (entropy -> mnemonic seed words)
- There is no ‘crypto’ module in RN, you’ll have to shim it (hello `rn-nodeify / babel-plugin-module-resolver`)
- A lot of libraries weren’t built for RN, welcome to the bleeding edge (hello patches to `node_modules`)

Network requests

- Entropy -> priv key -> pub key -> address
- Access network to see what this address “owns” (balance, tx list)
- Network requests leak privacy

At this point, youre done with 75% of the wallet

Dependencies are a liability

Aka supply chain attack

Ways to combat:

- Pin dependencies; install with `npm ci`
- Monitoring bots (Snyk, Renovate, Dependabot, Socket.dev)
- Local packages/artifacts registry (Verdaccio, Artifactory, etc)
- Fork under your organization (read diffs before merging upstream!)
- Copy from `./node_modules/` to `./my_modules/` and commit to git
- Implement from scratch ¯_(ツ)_/¯

Don’t add dependencies, especially dependencies with dependencies (aka transitive dependencies). Read code of your dependencies if you do. Use `npm ls`

etc

Careful with crash & analytics services:

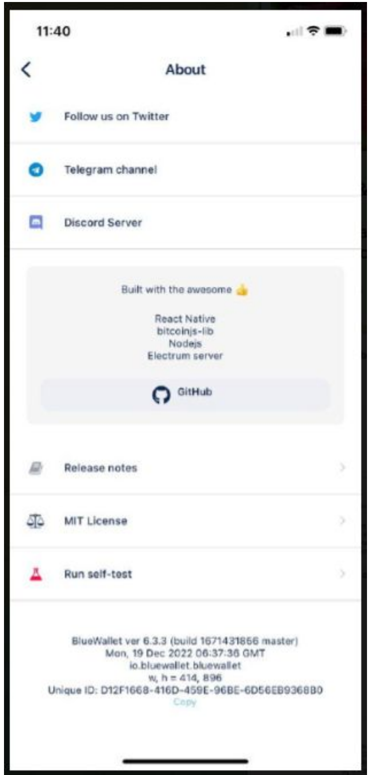
- Dont leak PII (disable IP tracking if you can etc)
- The less information you have on your users - the safer they are
- Dont leak secrets! (via `console.log()` kek)

Also,

- Over-The-Air updates is a bad idea (increases attack surface)
- You need security audit from security experts
- Apple/Google moderators review is more strict for financial apps (e.g. cant publish as an independent developer, need company with name resembling your app name)
- Adversarial thinking: always think how things can go wrong (mistakes are costly)
- Use standards (such as BIPs) for innteroperability, to not reinvent the wheel and not shoot yourself in the foot

etc

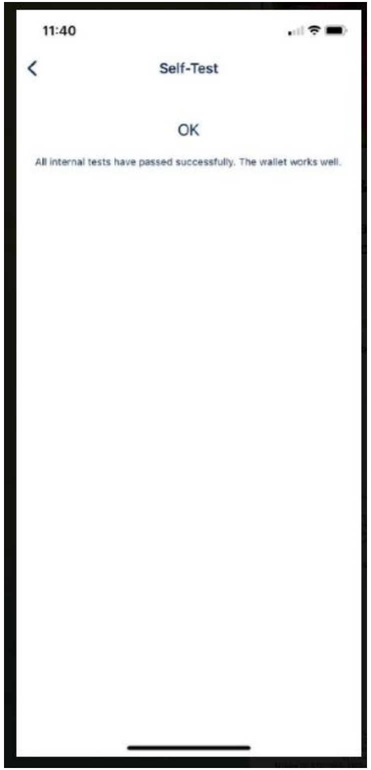
Screen that runs small subset of unit tests. Test it on e2e CI both iOS & Android



Built with the awesome 👍

React Native
bitcoinjs-lib
Nodejs
Electrum server

BlueWallet ver 6.3.3 (build 1671431856 master)
Mon, 19 Dec 2022 06:37:36 GMT
io.bluewallet.bluewallet
w, h = 414, 896
Unique ID: D12F1668-416D-459E-96BE-6D56EB9368B0



OK

All internal tests have passed successfully. The wallet works well.

“Not great, just ok”

“Only one coin”

“Just another wallet”

blue wallet

i@bluewallet.io

“Meh”

twitter.com/overtorment

“Some features”

Advanced stuff

- Mind the licenses of deps you use
- Release APKs & on F-Droid (F-Droid wont accept anything with Google deps/non-FOSS licenses)
- Reproducible builds are hard with RN
- With some effort, you could also ship MacOS (through Catalyst) app and Windows app!