



# The Ultimate Wallet

Designing a secure, safe and easy to use Bitcoin wallet.

Oleg Andreev

September 2014

# Goal

Bitcoin will be adopted only if it can be used as safely as conventional money and banks. Easily stolen or lost coins will be accessible only to a fraction of population limiting potential profit for investors.

We need a long-term strategy for easy to use, yet very robust way to store arbitrary amounts of bitcoins so more people can jump on it and make investors rich.

# also...

widespread adoption of Bitcoin stops wars, makes people richer, friendlier, happier, leaves more room for improvement in non-financial areas of life and leaves more money for your favorite charitable cause.

# Problem

Protection against stealing.

Protection against data loss.

Protecting privacy.

Easy to use for all of the above.

# Setup

Every person has a personal networking computer in their pocket.

Harder to steal.

Convenient to use.

# Threats

Device manufacturers (*bugs, leaks, backdoors*)

Wallet app developers (*bugs, leaks, backdoors*)

Physical thieves (*stealing funds, privacy invasion*)

Remote thieves (*malleability, tx history exploits*)

Yourself (*forgetting password, UI misuse*)

# Solution

1. Personal pocket device (iPhone, Android).
2. Secure UI.
3. Fully-auditable wallet behaviour.
4. Password-encrypted master key private backup.
5. Self-encrypted automatic wallet backup.
6. Unencrypted 2-of-3 paper master backup.
7. Two-tier keys (system-encrypted and user-encrypted).
8. Bitcoin Wallet API.
9. Blind multisignature custody for long-term savings.

# 1. Personal device

Harder to steal or lose.

Always with you to pay anywhere.

Networking — automatically backed up.

Encrypted and passcode/fingerprint-protected.





## 2. Secure UI

Simple, clear design to minimize user mistakes.

Secure defaults (unique keys for every payment, deterministic algorithms).

Cooperation with OS services for convenience and extra security layer (keychain, TouchID, sandboxed extensions).

Custom security features on top (passcode, multisig).

# 3. Fully auditable wallet

Protection against bugs and backdoors in a binary app / hardware.

System + application RNGs used only once.

Deterministic master key from RNG data and user's input.

Deterministic signatures.

Canonical data structures.

Compact, deterministic communications.

Can be discreetly observed at any time to verify lack of secret communications or backdoors in ECC.

## 4. Password-encrypted backup

Protection against device loss.

Backup is ensured before wallet is used.

User-defined location & one-time backup to reduce unwanted access.

Scrypt 80 Mb / 5 sec against brute force attacks.

# 5. Self-encrypted automatic backup

Protection against metadata loss (multisig keys, contracts, user notes and preferences).

Encrypted with master key (bruteforce impossible).

Automatic upload to multiple services before transaction broadcast.

Regular “heart-beat” retrieval and early warnings.

BIP proposal: <https://github.com/oleganza/bips/blob/master/bip-oleganza-backups.mediawiki>

## 6. Unencrypted 2-of-3 paper backup

Optional protection against password loss.

Printed once before wallet is used.

Can be stored as a single file or split to be put in several locations / custodies.

Optimized for “grandma scenario”.

Strong SSSS (1 piece leaks zero information).

## 7. Two-tier keys

Balance between convenience and security.

Small amounts available for one-tap payment.

Bigger amounts protected by a password.

Great for buying a drink or groceries.

90% of cash is better protected.

Wallet balances amounts and keys automatically.

## 8. Bitcoin Wallet API

Secure integration with 3rd party apps (identity, smart contracts etc.)

Apps do not need to reinvent a wallet or even keep users' keys.

Unified and convenient backup and security on wallet's side.

Apps only provide their own logic.

Spec draft: [https://github.com/bitcoin-wallet-api/spec/blob/gh-pages/core\\_spec.md](https://github.com/bitcoin-wallet-api/spec/blob/gh-pages/core_spec.md)

# 9. Blind multisignature custody

Ultimate security for large / long-term savings.

4-of-7 friends authorize each transaction in-person (or over a phone).

Least convenient (slow, interactive), but simple to use and understand.

Privacy via blind signatures. Custodians only authorize action, have no knowledge about amount, source or destination.

Funds are secure even when master key is stolen.

No magic, only time-tested simple EC math.

Spec: <http://oleganza.com/blind-ecdsa-draft-v2.pdf>

Implementation: <https://github.com/oleganza/CoreBitcoin/blob/master/CoreBitcoin/BTCBlindSignature.h>



# Example

User has 100 btc total.

90 btc are locked in several blind multisig transactions with 9 friends. Need 5 signatures to unlock.

≈9.5 btc are locked with a password.

≈0.5 btc available without extra authentication (device locked with TouchID).

Password-protected master key at home on a TimeCapsule.

Unencrypted master key is split in 3 paper parts:

- one part at home on a paper
- another part in a personal wallet
- third part kept with a trusted relative.

# Brief security analysis

Device died: restore everything from a personal backup.

Personal backup lost: create new backup.

Personal backup compromised: move funds to new wallet.

Need to move all funds: auth with friends.

Forgot password: restore from unencrypted backup.

Device stolen: higher risk of brute force access to privacy and 1% of funds, lower risk of brute force access to 9% of funds.

Unencrypted backup compromised, device/app compromised: privacy breach, 10% of funds stolen, 90% still locked up in multisig. Move to new wallet.

Some friends not available or collude: if below threshold, no threat; otherwise may block access, but not steal or learn about funds.

# Round up

User enjoys convenient general-purpose device.

UI guides through important steps to ensure safety and security.

Multiple levels of protection with gradual inconvenience and single place to manage them.

Shit work is automated out as much as possible.

Simple tasks are simple, harder tasks possible.

# Oleg Andreev

blog.oleganza.com

@oleganza



<http://oleganza.com/SecureWalletSeptember2014.pdf>