

# Grin BTC Swap Android App

---

## Overview

---

An Android application should be implemented, which can receive, store, and send Grin Coins and Bitcoin. Furthermore, this application should be able to execute Atomic Swaps between two users in a decentralized way. (Without the use of a server)

## Functional Requirements

---

### Milestone A

After this Milestone, the application should be able to create and recover keys from a seed phrase. And be able to sign transaction slate files (as used in the Grin cryptocurrency) to receive coins.

- **A0:** General Project setup
- **A1:** Generate a random seed phrase, from which secp256k1 keypairs can be generated and recovered
- **A2:** Receive and safely store Grin Coins from a transaction slate delivered as a file via E-Mail. In detail, the application needs to be able to read the data from the data, create a new output coin, and fill the file with the required data, specifically the signature.
- **A3:** Send back the updated transaction slate via E-Mail.

### Milestone B

After this Milestone, the application should be able to initiate Grin transactions. Meaning creating an initial transaction slate, sending it to a user (either via HTTP or e-mail attachment) as well as finalizing a final transaction slate.

- **B1:** Send Grin to a remote location via HTTP.
- **B2:** Send Grin to a remote location via Email attachment.
- **B3:** Send a finalized Grin transactions to the Grin network. (By sending it to a Grin node)

### Milestone C

After this, Milestone, users should be able to retrieve, transmit, and safely store Bitcoin on the application.

- **C1:** Create Bitcoin keypairs and Bitcoin addresses
- **C2:** Create and send P2PK Bitcoin transactions

### Milestone D

After this, Milestone Atomic Swaps between Bitcoin and Grin should be possible using E-Mail attachments as communication between participants.

- **D1:** Create and publish time-locked multiparty Outputcoins. Interaction between participants via E-Mail attachments. (Grin side of the swap)

- **D2:** Create a secret scalar  $x$  and  $x*G$ . ( $x$  is the secret transferred to the second party finalizing the exchange)
- **D3:** Create a Bitcoin transaction with an output spendable by the receiver if he gets to know  $x$  or by the sender after a timeout. (BTC side of the swap)
- **D4:** Initiate the process of spending the multiparty Grin coin while sending  $xG$  to the other party to check.
- **D5:** Finalizing the swap by retrieving  $x$  from the Grin signature and creating a BTC transaction to spend the Bitcoin output.

## Milestone E

After this, Milestone, participants can discover each other by connecting to a (centralized) registry. Furthermore, this registry will hold public keys, and therefore enables the creation of secure channels.

- **E1:** Registry to upload, hold, and retrieve public keys of identities.
- **E2:** Registry to keep update and retrieve the state of open / fulfilled / canceled Atomic Swaps.
- **E3:** Generate session keys from identity keypairs to create secure channels.
- **E4:** Execute Atomic Swaps via Secure Channels instead of plain text.