Introduction

The document will refer to the description of the envisaged securities in Hodler.Tech's hardware wallet (hereinafter referred to as HT). Some of the safety features may be changed due to the early stage of production. Strong cryptography, limited interference with software and hardware by third parties to the maximum extent possible, time delays and autodestruction of keys will be the main operation principle of these security features.

1 - Pseudorandom Number Generator

One of the potential problems of today's hardware wallets is the pseudorandom number generator. With simpler structures, it would be possible to reproduce the generated number by a third party. This problem results from the fact that computers create pseudorandom numbers on the basis of an algorithm. Our wallet here adds randomness based on interaction with the user. Here you can apply collecting coordinates of a finger on the screen (x, y) during motion, and based on this data enrich your drawing numbers. This will generate a safer seed for the future private key.

2 - Private Key Protection

It is important to ensure that the seed cannot flow out anywhere. Its direct form must not remain in memory for too long. We predict only two cases where the seed, after decryption, is placed in the device's RAM memory for several CPU cycles, when its apparent form is required by the logic of the wallet:

1 - Following the command to generate a new address for a coin 2 - When signing a transaction The seed must then be cleaned from the device's memory in decoded form. You cannot allow all or parts of it to remain in the RAM memory, because if a third party is able to retrieve a memory dump, private keys may be stolen. In any other case than the above situations, the key remains secretly encrypted in a device. We also foresee a seed destruction mechanism when the software detects unwanted access, e.g. when you enter a bad access password to your wallet ten times in a row.

Classified seed will be protected by a password set by the user. In order to prevent brute force attempts, we will introduce a time delay during an attempt to enter the password. The following will not be the case:

```
password=userInput() sleep(5000); decryptSeed(password);
```

The above solution allows to bypass the protection in conditions, e.g. when a third party, being in possession of the classified seed, is able to emulate the wallet and deceive the system clock. We assume the use of an algorithm (or algorithms) that uses hardware resources and time, and the result of which will be needed to decode the seed.

Example:

```
password = userInput();
pair1 = scrypt(password, n=2^18, r=8, p=1, dkLen=32);
pair2 = pbkdf2(pair1, c=2^16, dkLen=32);
decryptSeed(pair1+pair2);
```

Such a procedure extends the time of one test, depending on computational power, from a second to a dozen or so times. Imagine such pessimistic situations:
* A user was robbed of the wallet with a lot of funds.
* A user has secured the seed only with an alphanumeric password of 5 characters in length. * A thief has successfully acquired the seed file and already has a prepared tool for brute password cracking, and high computing power.

The time for a thief to crack a weak password is:
len=5 (password length)
range=60 (possible characters, [a-z, A-Z, 0-9])
x=range^len x=777 600 000
t=0.5s (single try time)

So, assuming that he uses only one machine and can check only 2 passwords within a second, it will take him more than 12 years. I think the owner has already managed to restore the seed from the backup and transfer funds to a safe place. Additional complexity may be that an alphanumeric password is not necessarily a password. It can be a zigzag known from screen locks on Android, saved in binary form. In this case, the difficulty for cracking will increase. As the creators, we have a huge margin for action here due to the 2.4-inch touchscreen display.

3 - Sterile Environment

Our wallet will have a Linux-based operating system with a graphical environment for wallet application support. It will be devoid of debugging interface and the ability to install applications other than those designed to maintain functionality. This will eliminate the risk of exploiting security gaps in other applications for data theft. Due to the fact that the wallet will require an Internet connection (WiFi/GSM), the system will allow only connection to the node server of our wallet. Even if someone takes over this connection, the only thing they can do is to discover our public keys. They will not be able to send messages on the device screen, steal a private key, or intercept transactions. This is due to the nature of cryptocurrencies. The system will only have our wallet application and drivers installed for operation purposes.

HODLER arrives at your address sealed. If seal hasn't been broken it means that no third party handling of the device occured on the way between manufacturing and delivery.

When you start HODLER for the first time, checksums of files in the system will be checked against those signed digitally by the company. If they don't match, it will be clear some manipulation and involvement of third party occured, what subsequently means your funds may be endangered. In this situation it is advised not to use the HODLER device and get in touch with the HODLER company immediately. Checking file integrity is crucial so additional verification mechanisms need to be implemented. HODLER is equipped with an unique version of the system (each device is different) based on specific identificators installed during the production process. Each device will be processed individually where after 3 days from confirmation of the delivery by the buyer, they will get 512 bit long password with secret key encrypted inside. Third party cannot know the password, but even if they manage to get hold of it, the delivery confirmation by the rightful buyer is required to start the verification process. In other words, the company is installing into the device (protected by encrypted password), known only to them, secret data (image, pictogram, easy to verify visually) to be display on its first run. When the rightful buyer confirms the delivery, they are then taken to the activation website page in order to verify all the secret data. This secret information would be shown only once.

Mismatch proves that system image have been spoofed. “Already activated” message - will mean that someone has already claimed this device. **Both cases are critical security flaws!** When this happens the rightful owner will be entitled to exchange their device for a brand new. Implementation of this process remove the danger called ‘Man in the middle’. Only verification by a rightful owner leads to installation of the HODLER system and creation of the random seed (not to be confused with pseudorandom).

Random seed creation: User will be asked to carry out a few activities like taking a random photo, rotate a device and type in words on a touch screen.These activities will be converted and added to seed sequences, providing a genuine randomness. We want to take a step away from seeds that are only pseudorandom (based on algorithms). Pseudorandom seeds have been proved not to be secure, as algorithms can be recreated and the same ‘random’ numbers generated.
A genuine randomness comes from the fact that it is impossible to repeat the same photo or make the same hand or finger gesture. With this genuine randomness HODLER creates unique private key which becomes a masterseed of wallets included inside. Next the newly created seed needs to be encrypted with time and work consuming algorithm (variation of Scrypt) and then stored encrypted in the memory of the device. Password to the algorithm will become PIN number for signing transactions.
In the time of seed creation, public keys (created with ECDSA and similar, depending on given coin code) for balance checking will be saved. Based on that built-in app will work without seed revealing.

Checking statements and last transactions doesn’t require private keys according to rules of cryptocurrencies. Internet connection is not required for viewing balances, they can be checked when device is fully offline by checking wallet address on other-party blocks explorers, with wallet’s address known to its user. If user want to check a balance (in case when they received new funds), the device can be connected to internet using baseband or WiFi. Then HODLER will send identification message which determines if the connection is legitimate. Verification is done by comparison of signatures. Each signature is unique and can be checked easily. HODLER can verify the current connection and approve or decline it when third party involvement has been detected.

When making a payment, the delivery address can be:

- Typed in manually
- Scanned with OCR
- Scanned with QR code reader
- Received via secure email (7bit ASCII encoded) into HODLER's sandboxed mail app

After address and quantity of a selected currencies has been entered, HODLER will start signing process. All processes, except the critically needed, will be stopped. HODLER will go offline and user will be asked to type in their password to decode the seed. Password is an entry data for algorithms with some time and resource complexity, which result is decrypting the seed. In this moment, signing of the transaction occurs and the seed gets wiped out from the memory. Next, the signed transaction is send to our node which then publishes it.

Private keys that are not related to the seed (impossible to be spawned by seed so won't be included in backup), can be imported similarly as above: with OCR, QR etc. Those keys will be securely stored on HODLER. To allow future recovery of wallets, HODLER sends files created during procedure described below to the backup cloud.

generatedHashAsKey = sha256(seed).repeat(100000)
blob=aes256CBC_encrypt(generatedHashAsKey,importedPrivateKey)

This blob is saved in the cloud through point-to-point encryption. Even if this file gets stolen, third party will not get an access to coins and wallets as HODLER seed is required. Cloud serves as cold storage for wallets, which are not the children of masterseed (imported ones)
The seed doesn't get sent anywhere and only remain in RAM memory for cycles of CPU.

In case of third parties gaining an access to HODLER wallet, they still need a password to decode the seed. HODLER has an autodestruction function which is executed after several incorrect password trials. However, in case of hijacking internal data with encrypted seed, there is still possibility to bruteforce attack, but even with a weak password, it takes a lot of time (few months or years) which gives the rightful owner enough time to recover their wallet from seed anywhere and transfer funds to non-exposed places.