Air-gapped QR Code Data

Between

Cold Storage Wallet and Online App

t.chen@ellipal.com

R1.18

The goal of ELLIPAL's secure crypto wallet is to keep the security feature of cold storage and benefit from convenience of the mobile app in a single solution. We called it "The Cold Wallet 2.0" which provides users next generation cold storage wallet comparing paper wallet and offline computers, and also is easy to use with a companion mobile App.

The solution comes to well-designed hardware with the camera and LCD screen only, no any other interfaces like USB, Bluetooth or WIFI. The data between the wallet and the phone app is air-gapped QR Code which is visual and also simple to be verified by the user.

This document shows ELLIPAL's air-gapped data format inside the QR Code. You may find the source code of decoding/encoding demonstration and libs in Javascript on https://github.com/ELLIPAL/js-ellipal.

The data is in generic URI format:

elp://OPTIONS@ACTION/ACTIONDATA

OPTIONS can be format version and page indicator of QR Code data

ACTION defines the intent of this URI: synchronizing the address, transferring tx to be signed or signed.

OPTIONS

This part is normally empty. If existing, it separates the latter part by the char '@'. Different items in OPTIONS are divided by the char '|'.

The table shows some examples:

Content	Remark
2:8	this URI is the 2nd-page ACTIONDATA of the whole 8 pages of the ACTION. page indicator is empty if there is one page only.
V2	V2 of air-gapped data format. V1 is as empty by default to shorten total URI length.
V2|2:8	V2 of air-gapped data format and 2nd of 8 pages data.

ACTION and ACTIONDATA

ACTION LIST

ACTION	direction	function
sync	Wallet to App	Synchronizing the single account. deprecated.
sync2	Wallet to App	Synchronizing an account of one or multi-cryptocurrency from Wallet to App
tosign	App to Wallet	App shows the TX data to be signed by the Wallet
signed	Wallet to App	Wallet shows signed data to App
eosnamesync	App to Wallet	EOS employs EOS account name rather than an address in the user's activities. When a user created an EOS on App via the public key, we could sync it back to the Wallet. The name should be checked by the user.

- **sync**

elp://sync/accountname/CRYPTOTYPE/address/pubkey/legacyaddress

accountname: user set account name on Wallet.
CRYPTOTYPE: the coin or token symbol like BTC, ETH, LTC, BNB, and others.
address: Cryptocurrency address.
pubkey: Only needed for BTC and its forks like BCH, LTC or DGB.
legacyaddress: Only needed for BCH for App use

- **sync2**

elp://2:16@sync2/walletrelease/deviceid/accountname/accountdata/indexbtcaddress

sync2 is employed to synchronize ELLIPAL multi-chain, multi-cryptocurrency integration account from hardware wallet to App. One account can be multi-cryptocurrency group derived from a set of mnemonics words, or single coin address imported from a private key. The user can sync all crypto addresses or any single one in the whole group. App distinguishes the account by the BTC address in the group. However, when syncing a single address in the group or one private key imported address, the indexbtcaddress is set as the string "0". The only address in the accountdata then is the mark for the account. sync2 is also used to update the account name if the user changes it.

walletrelease: the hardware wallet release.
deviceid: the hardware device id.
accountname: user set account name on the wallet.
accountdata: multi-cryptocurrent data is separated by the char ']'. extra data in particular crypto is added by char '['. Following is an example:
BTC[addresss[pubkey]ETH[address]BCH[address[pubkey[legacyaddress]

The list shows the crypto extra data.

```
BTC[address[pubkey
ETH[address
BCH[newaddress[pubkey[legacyaddress
```

DSH[address[pubkey

LTC[address[pubkey

DGB[address[pubkey

XRP[address[pubkey

ETC[address

CMT[address

XLM[address

BTX[address[pubkey

BTG[address[pubkey

BCD[address[pubkey

DOGE[address[pubkey

TRX[address

DCR[address[pubkey

RGS[address[pubkey

ZXC[address[pubkey

EOS[pubkey1[pubkey2

SMART[address[pubkey

ECA[address[pubkey

indexbtcaddress: the BTC address in the group addresses of the account. When sync a single address, this is set as the string "0".

- **tosign**

elp://2:8@tosign/CHAINTYPE/address/tx/tokensymbol/decimal

CHAINTYPE: the native coin symbol of the chain like BTC, ETH or TRX.
address: the crypto address.
tx: unsigned tx encoded in BASE64 modification(char '/' replaced by '_').

tokensymbol: used for token transactions like ERC20.

decimal: used for token transactions to show the amount readable by people.

- **signed**

elp://signed/CHAINTYPE/address/hexdatasigned

CHAINTYPE: the native coin symbol of the chain like BTC, ETH or TRX.

address: the crypto address.

hexdatasigned: combined hex string of signed C S and V(only in ETH like chains).

- **eosnamesync**

elp://eosnamesync/eosaccount/owner_public_key/active_public_key