

Abelian Improvement Proposal 0015: Revisit: Mnemonic Codes for Generating Deterministic Accounts *

Abelian

November 5, 2025

Abstract. This AIP is a revisited version for AIP0011. To be self-completed, this AIP adds a standalone section based on the AIP0011. In the added section, this AIP describes how an additional root seed, say `CoinVKRootSeedAut`, as a part of Account-Root-Seeds, is derived from the Entropy-Seed.

This `CoinVKRootSeedAut` is added to separate the root seed for value-hidden key for CTAUT from that for full-private Abelian coin, to provide more flexible extension in the future.

*Finalized on 2024.11.21.

Contents

1	Mapping Between Entropy-Seeds and Mnemonics	3
1.1	Sample An Entropy-Seed	3
1.2	From Entropy-Seeds To Mnemonics	3
1.3	From Mnemonics To Entropy-Seeds	4
1.4	Test vectors	5
2	Derivation of Account-Root-Seeds from Entropy-Seeds	5
2.1	Preliminaries on Abelian Wallet Account	5
2.2	Derivation From Entropy-Seeds to Master-Seeds	7
2.3	Derivation From Master-Seed to Account-Root-Seeds	8
2.4	Deterministic Public Rands	9
2.5	Test Vectors	10
3	Compatibility	10
3.1	A Previous Version	10
3.2	Compatibility	11
4	Revisit: Derivation From Master-Seed to Account-Root-Seeds	12
4.1	Revisit: Derivation From Master-Seed to Account-Root-Seeds . . .	12
4.2	Compatibility	14
A	WordList	15
B	Test vectors For Mapping Between Mnemonics and Entropy-Seeds	21
C	Test vectors For Derivation of Master-Seeds and Account-Root-Seeds	23

1 Mapping Between Entropy-Seeds and Mnemonics

1.1 Sample An Entropy-Seed

This proposal uses an entropy of 256 bits as the random bit-string seed for a deterministic wallet account, referred to as **Entropy-Seed**.

Note that it is required that **Entropy-Seed** is sampled randomly and uniformly from $\{0, 1\}^{256}$, as shown in Algorithm 1.

Algorithm 1 *SampleEntropySeed()*

```
1: entropyseed  $\xleftarrow{\$} \{0, 1\}^{256}$   
2: return entropyseed
```

1.2 From Entropy-Seeds To Mnemonics

For a given **Entropy-Seed** *entropyseed*, the corresponding mnemonic is obtained by the following Algorithm 2.

Algorithm 2 *EntropySeedToMnemonic(entropyseed, wordlist)*

```
1: cs  $\leftarrow$  the first 8 bits of SHA256(entropyseed)  
2: ext  $\leftarrow$  entropyseed || cs  
3: ms23 || ms22 || ... || ms0  $\leftarrow$  ext  
4: for t = 0 to 23 step 1 do  
5:   it  $\leftarrow$  BinaryToInt11(mst)  
6:   mnemonic[t]  $\leftarrow$  wordlist[it]  
7: end for  
8: return mnemonic
```

Remark:

- *entropyseed* is an **Entropy-Seed** sampled as in Section 1.1.
- Here SHA256 is described in the standard of SHA2 [4].

- *ext* consists of $256+8 = 264$ bits.
- *ext* is split into 24 groups of bits, say $ms_0, ms_2, \dots, ms_{23}$, each consisting of 11 bits, such that $ms_{23} \parallel ms_{22} \parallel \dots \parallel ms_0 = ext$.
- `BinaryToInt11()` is the **standard** algorithm that converts binary-string in $\{0,1\}^{11}$ to the corresponding decimal integer in $[0, 2047]$. In particular, `BinaryToInt11(000,0000,0001) = 1`, `BinaryToInt11(000,0000,1000) = 8`, `BinaryToInt11(100,0000,0000) = 1024`, and so on.
- *wordlist* is the commonly used wordlist with 2048 words as in BIP0039 [5], as shown in Appendix A.
- As a result, the output *mnemonic* consists of 24 words in *wordlist*.

1.3 From Mnemonics To Entropy-Seeds

The mapping from Mnemonics to Entropy-Seeds is just the inverse procedure as shown in the following Algorithm 3.

Remark:

- *mnemonic* consists of 24 words.
- *wordlist* is the commonly used wordlist with 2048 words as in BIP0039, as shown in Appendix A.
- `LookupIndex(word, wordlist)` finds the index of *word* in *wordlist*. Note that if the output index is not in the scope $[0, 2047]$, it implies that an illegal word is used and FAIL is returned.
- `IntToBinary11()` is the inverse of `BinaryToInt11()`, converting an integer in $[0, 2047]$ to a binary-string in $\{0,1\}^{11}$.
- The output *entropyseed* is a 256-bit string in $\{0,1\}^{256}$.

Algorithm 3 *MnemonicToEntropySeed(mnemonic, wordlist)*

```
1: for  $t = 0$  to 23 step 1 do
2:    $i_t \leftarrow \text{LookupIndex}(\text{mnemonic}[t], \text{wordlist})$ 
3:   if  $i_t \notin [0, 2047]$  then
4:     return FAIL
5:   end if
6:    $ms_t \leftarrow \text{IntToBinary11}(i_t)$ 
7: end for
8:  $ext \leftarrow ms_{23} \parallel ms_{22} \parallel \dots \parallel ms_0$ 
9:  $\text{entropyseed} \leftarrow \text{the first 256 bits of } ext$ 
10:  $cs \leftarrow \text{the last 8 bits of } ext$ 
11:  $cs' \leftarrow \text{the first 8 bits of } \text{SHA256}(\text{entropyseed})$ 
12: if  $cs' \neq cs$  then
13:   return FAIL
14: end if
15: return  $\text{entropyseed}$ 
```

1.4 Test vectors

The test vectors for the mapping between Mnemonics and Entropy-Seeds are given in Appendix B.

2 Derivation of Account-Root-Seeds from Entropy-Seeds

2.1 Preliminaries on Abelian Wallet Account

Account and Account-Root-Seeds. In Abelian, as shown in Fig. 1, each **account** consists of a set of **root seeds/keys**, referred to as **Account-Root-Seeds**, say $(\text{CoinSpKeyRootSeed}, \text{CoinSnKeyRootSeed}, \text{CoinDetectorRootKey}, \text{CoinVKeyRootSeed})$, where CoinSnKeyRootSeed and CoinVKeyRootSeed are optional. In particular, for an account which will generate only pseudo-private addresses, the CoinSnKeyRootSeed and CoinVKeyRootSeed are null. Note that CoinSpKeyRootSeed , CoinSnKeyRootSeed ,

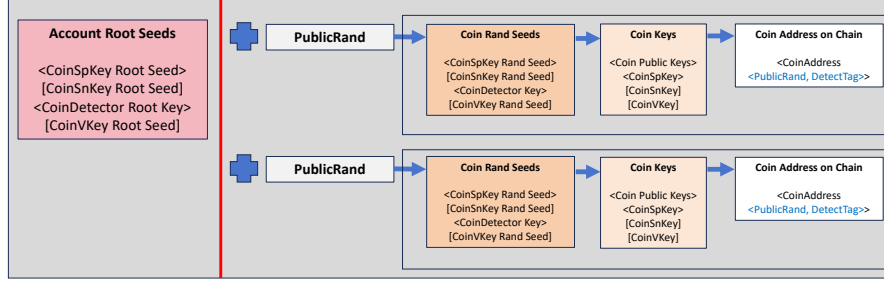


Figure 1: Account and (Address, key)

CoinDetectorRootKey, and CoinVKeyRootSeed are all 512-bit long.

Public Rand and (Address, Key). As shown in Fig. 1, to generate an (Address, Key) pair under an account, a **Public Rand** (with 512 bits) needs to be introduced. In particular, for each given Public Rand, a corresponding (Address, Key) will be *deterministically* generated from the Account-Root-Seeds. Note that there are two ways to generate an (address, key) pair under an account, namely (1) given only the Account-Root-Seeds: sample a random Public Rand and generate the (address, key) pair from the given Account-Root-Seeds and sampled Public Rand, or (2) given the Account-Root-Seeds and a well-form Public Rand: generate the (address, key) pair from the given Account-Root-Seeds and Public Rand.

Derivation of Account-Root-Seeds. For such an account, it is ideal that CoinSpKeyRootSeed, CoinSnKeyRootSeed, CoinDetectorRootKey, and CoinVKeyRootSeed are independent entropies. However, from the view of practice, it is desired that they are derived from an entropy, referred to as **Master-Seed**, as shown in Fig. 2. As a response to such a desire, to provide good user-friendliness (say, using mnemonic), this proposal derives a **Master-Seed** from **Entropy-Seed**, and then derives **Account-Root-Seeds** from the **Master-Seed**.

Note that this proposal does not directly use **Entropy-Seed** as **Master-Seed**.

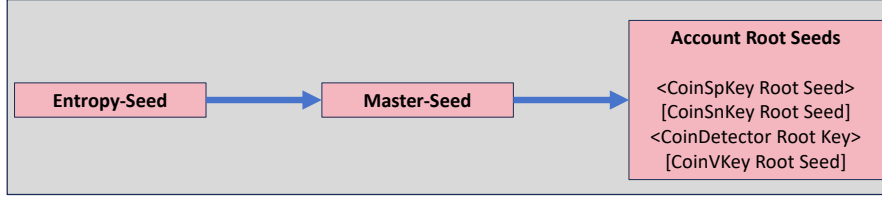


Figure 2: Derivation of Account-Root-Seeds

This is to provide flexibility to potential extension, for example, deriving multiple accounts from one Entropy-Seed.

Notations. Below we define a PRF (Pseudo Random Function)

$$\text{PRF}(key, input) := \text{KMAC256}(key, input, 512, \text{"ABELIANPRF"})$$

where KMAC256 serves as a PRF [2] (approved by NIST) to generate 512-bit output, *input* serves as the context, and “ABELIANPRF” specifies the Domain Separation Customization String. Note that here with a 512-bit (or 256-bit) key and a 512-bit output, the above PRF provides 256-bit security.¹

2.2 Derivation From Entropy-Seeds to Master-Seeds

Given an Entropy-Seed, a corresponding Master-Seed is deterministically derived as shown in the following Algorithm 4. Note that with the security of PRF, it can be deduced that given a master-seed, it is infeasible to distinguish the Entropy-Seed from a random $x \in \{0, 1\}^{256}$.

¹Why use KMAC256 with a 256-bit input and a 512-bit output rather than HMAC-SHA512? To be safe, we want to use SHA3 rather than SHA2 (i.e., SHA512), and KMAC is the only known (variable-length message) authentication code algorithm based on KECCAK which is the underlying function of the SHA3 standard. Although it is claimed that HMAC can be based on any Hash function, we are not sure whether the claim is applicable to SHA3, since HMAC was proposed before SHA3.

Algorithm 4 *EntropySeedToMasterSeed(entropyseed, customizationContext)*

1: $masterseed \leftarrow \text{PRF}(entropyseed, \text{"AccountMasterSeed"} \parallel customizationContext)$

2: **return** $masterseed$

Remark:

- $entropyseed$ is 256-bit long, determined by the 24-word mnemonic rule.
- $entropyseed$ serves as the key.
- $\text{"AccountMasterSeed"} \parallel customizationContext$ serves as the input, where different applications may use different customizationContext (which is "" by default). This also allows to support the case of generating multiple accounts from one mnemonic.
- $masterseed$ is 512-bit long.

Note that this derivation is very different from that of BIP0039.

2.3 Derivation From Master-Seed to Account-Root-Seeds

Given a Master-Seed, the corresponding Account-Root-Seeds are deterministically derived as shown in the following Algorithm 5.

Algorithm 5 *MasterSeedToAccountRootSeeds(masterseed)*

- 1: $coinSpKeyRootSeed \leftarrow \text{PRF}(masterseed, \text{"CoinSpendKeyRootSeed"})$
 - 2: $coinSnKeyRootSeed \leftarrow \text{PRF}(masterseed, \text{"CoinSerialNumberKeyRootSeed"})$
 - 3: $coinDetectorRootKey \leftarrow \text{PRF}(masterseed, \text{"CoinDetectorRootKey"})$
 - 4: $coinVKeyRootSeed \leftarrow \text{PRF}(masterseed, \text{"CoinValueKeyRootSeed"})$
 - 5: **return** $(coinSpKeyRootSeed, coinSnKeyRootSeed, coinDetectorRootKey, coinVKeyRootSeed)$
-

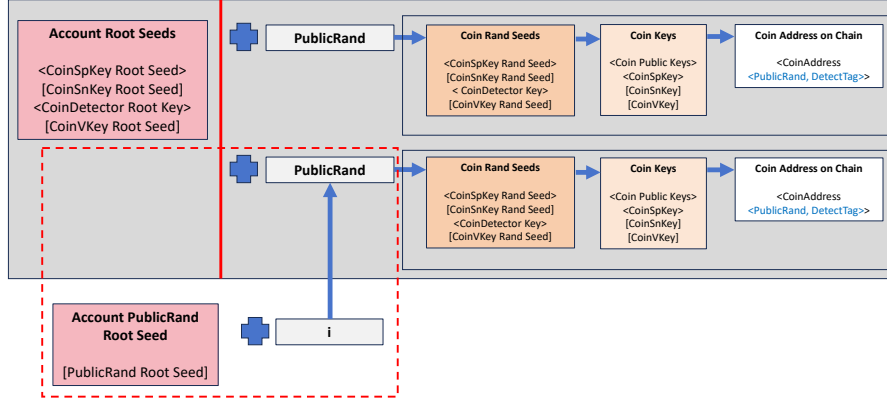


Figure 3: Account with Deterministic Public Rands

2.4 Deterministic Public Rands

In some scenarios, it is desired that deterministic Public Rands are used. To support such scenarios, this proposal defines a rule to deterministically generate Public Rands from integers in $[0, 2^{32} - 1]$, referred to as “Sequence Numbers”. In particular, this proposal derives a `PublicRandRootSeed` for an account from a given `Master-Seed`, and then derives Public Rands from given sequence numbers when needed, as shown in Fig. 3, Algorithm 6, and Algorithm 7.

Algorithm 6 *MasterSeedToAccountPublicRandRootSeed(masterseed)*

- 1: `publicRandRootSeed` \leftarrow `PRF(masterseed, “PublicRandRootSeed”)`
 - 2: **return** `publicRandRootSeed`
-

Algorithm 7 *DerivePublicRand(publicRandRootSeed, i)*

- 1: `seqNo` \leftarrow `EncodeSeqNo(i)`
 - 2: `publicRand` \leftarrow `PRF(publicRandRootSeed, seqNo)`
 - 3: **return** `publicRand`
-

Remark:

- `publicRandRootSeed` output by *MasterSeedToAccountPublicRandRootSeed()* is 512-bit long.

- i is an integer in $[0, 2^{32} - 1]$.
- `EncodeSeqNo(i)` encodes $i \in [0, 2^{32} - 1]$ to a hex-string of length 8 with lower case, say 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f . In particular, `EncodeSeqNo(1) = 00000001`, `EncodeSeqNo(10) = 0000000a`, `EncodeSeqNo(31) = 0000001f`.
- The output `publicRand` is 512-bit long.

2.5 Test Vectors

The test vectors for the derivation from Entropy-Seeds to Master-Seeds and the derivation from Master-Seeds to Account-Root-Seeds are given in Appendix C.

3 Compatibility

3.1 A Previous Version

At *height* = 300,000, we released packages of `AbewalletMLP-v1.0.1`, which follow BIP0039, namely

- The mapping between Entropy-Seeds and Mnemonics is the same as Algorithm 2 and Algorithm 3.
- The derivation from Entropy-Seeds to Master-Seeds is shown as the following Algorithm 8.
- The derivation from Master-Seeds to Account-Root-Seeds is shown as the following Algorithm 9.

Remark:

Algorithm 8 *EntropySeedToMasterSeed_{BIP39}(entropyseed)*

- 1: $words \leftarrow \text{EntropySeedToMnemonic}(entropyseed)$
 - 2: $key \leftarrow \text{Use whitespace to splice words}$
 - 3: $masterseed \leftarrow \text{PBKDF2}(key, \text{"mnemonic"}, 2048, 64, \text{HMAC-SHA512})$
 - 4: **return** $masterseed$
-

Algorithm 9 *MasterSeedToAccountRootSeeds_{old}(masterseed)*

- 1: $\text{coinSpKeyRootSeed} \leftarrow \text{PRFOLD}(masterseed, \text{"spendkey"})$
 - 2: $\text{coinSnKeyRootSeed} \leftarrow \text{PRFOLD}(masterseed, \text{"serialnumberkey"})$
 - 3: $\text{coinDetectorRootKey} \leftarrow \text{PRFOLD}(masterseed, \text{"valuekey"})$
 - 4: $\text{coinVKeyRootSeed} \leftarrow \text{PRFOLD}(masterseed, \text{"detectorkey"})$
 - 5: **return** $(\text{coinSpKeyRootSeed}, \text{coinSnKeyRootSeed}, \text{coinDetectorRootKey}, \text{coinVKeyRootSeed})$
-

- PBKDF2 [1] applies a pseudorandom function (such as HMAC-SHA512) repeatedly to the salt and password to generate the key. The NIST Recommendation [6] approved PBKDF2 as the PBKDF using HMAC with any approved hash function as the PRF, and decided to revise it [3].
- Here HMAC-SHA512 is used as the pseudo-random function [5].
- PRFOLD is defined by :

$\text{PRFOLD}(key, input) := \text{KMAC256}(key, input, 512, \text{"PQABELIAN-WALLET"})$

where KMAC256 serves as a PRF [2] to generate 512-bit output, $input$ serves as the context, and "PQABELIAN-WALLET" specifies the Domain Separation Customization String.

3.2 Compatibility

Note that there may be only a few of users that created mnemonics by using AbewalletMLP-v1.0.1, and that a compatibility solution transparent to users may be pretty complicated and may cause huge development effort and inefficiency, we could use a simple way to address the compatibility.

In particular, at the UI-layer of a wallet, the user is noticed that if his mnemonic was generated by **AbewalletMLP-v1.0.1**, he should tell the system (for example by a check box) and the system will call the above Algorithm 8 and Algorithm 9 accordingly.

4 Revisit: Derivation From Master-Seed to Account-Root-Seeds

4.1 Revisit: Derivation From Master-Seed to Account-Root-Seeds

Given a Master-Seed, the corresponding Account-Root-Seeds are deterministically derived as shown in the following Algorithm 10.

Algorithm 10 *MasterSeedToAccountRootSeeds(masterseed)*

```

1: coinSpKeyRootSeed  $\leftarrow$  PRF(masterseed, "CoinSpendKeyRootSeed")
2: coinSnKeyRootSeed  $\leftarrow$  PRF(masterseed, "CoinSerialNumberKeyRootSeed")
3: coinDetectorRootKey  $\leftarrow$  PRF(masterseed, "CoinDetectorRootKey")
4: coinVKeyRootSeed  $\leftarrow$  PRF(masterseed, "CoinValueKeyRootSeed")
5: coinVKeyRootSeedAut  $\leftarrow$  PRF(masterseed, "CoinValueKeyRootSeedAut")
6: return (coinSpKeyRootSeed, coinSnKeyRootSeed, coinDetectorRootKey,
    coinVKeyRootSeed, coinVKeyRootSeedAut)

```

Note that, compared with the original Algorithm 5, only a `coinVKeyRootSeedAut` is added. Fig. 4 and Fig. 5 shows the change as well. It is worth noticing that when generating an address, at most one of `{coinVKeyRootSeed, coinVKeyRootSeedAut}` is used. Please refer to the following Sec. 4.2 for the details.

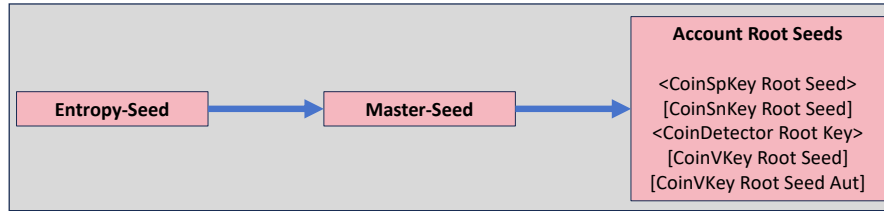


Figure 4: Revisit: Derivation of Account-Root-Seeds

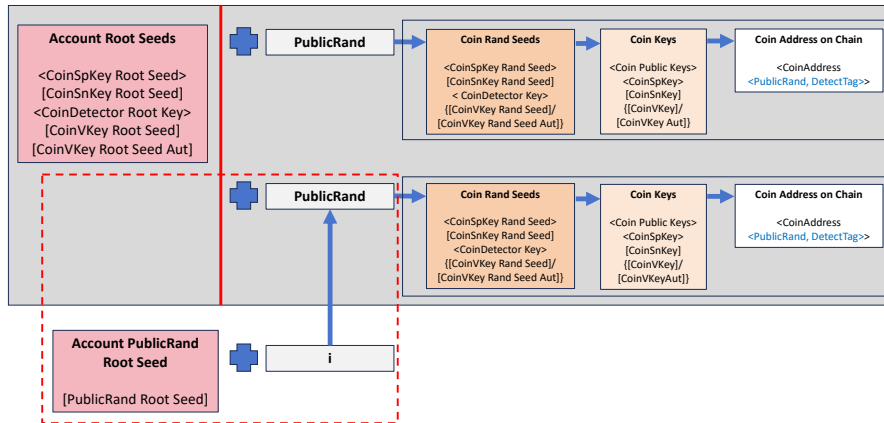


Figure 5: Revisit: Account with Deterministic Public Rands

4.2 Compatibility

Before the Aconcagua Fork, the chain supports three types of address corresponding to three privacy levels, say “Type 0” for “Full-Privacy-Legacy”, “Type 1” for “Full-Privacy”, “Type 2” for “Pseudonym-Privacy”. Type 0 address is not the scope of this document.

From the Aconcagua Fork, an additional type of address is added, say “Type 3” for “Pseudonym-CT-Privacy” where “CT” is for “Confidential Transaction” of Abelian User Token (AUT). The Type 3 address is used to replace the Type 2 address and Type 2 address is deprecated but still supported. In particular, for these addresses, the rules are as below:

- To create a Type 1 address,
(CoinSpKeyRootSeed, CoinSnKeyRootSeed, CoinDetectorRootKey, CoinVKeyRootSeed) is used.
- To create Type 2 address,
(CoinSpKeyRootSeed, CoinDetectorRootKey) is used.
- To create Type 3 address,
(CoinSpKeyRootSeed, CoinDetectorRootKey, CoinVKeyRootSeedAut) is used.

References

- [1] B. Kaliski. PKCS #5: Password-Based Cryptography Specification Version 2.0. RFC 2898, Sept. 2000.
- [2] J. Kelsey, S. jen Chang, and R. Perlner. Sha-3 derived functions: cshake, kmac, tuplehash and parallelhash, 2016-12-22 00:12:00 2016. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=922422.

- [3] NIST. Nist to revise sp 800-132, recommendation for password-based key derivation – part 1: Storage applications. Website, 2023. <https://csrc.nist.gov/News/2023/decision-to-revise-nist-sp-800-132>.
- [4] N. I. of Standards, T. (NIST), and Q. Dang. Secure hash standard (shs), 2012-03-06 00:03:00 2012.
- [5] M. Palatinus, P. Rusnak, A. Voisine, and S. Bowe. Mnemonic code for generating deterministic keys. <https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki>. Accessed 10 October 2024.
- [6] M. S. Turan, E. B. Barker, W. E. Burr, and L. Chen. Recommendation for password-based key derivation ::part 1: storage applications, 2010-01-01 05:01:00 2010.

A WordList

The wordlist is the same as that of BIP0039 at <https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt>. In particular, all words in sequence as following:

abandon ability able about above absent absorb abstract absurd abuse access accident account accuse achieve acid acoustic acquire across act action actor actress actual adapt add addict address adjust admit adult advance advice aerobic affair afford afraid again age agent agree ahead aim air airport aisle alarm album alcohol alert alien all alley allow almost alone alpha already also alter always amateur amazing among amount amused analyst anchor ancient anger angle angry animal ankle announce annual another answer antenna antique anxiety any apart apology appear apple approve april arch arctic area arena argue arm armed armor army around arrange arrest arrive arrow art artefact artist artwork ask aspect assault asset assist assume asthma athlete atom attack attend attitude attract auction audit august aunt author auto autumn average avocado avoid awake aware away

awesome awful awkward axis baby bachelor bacon badge bag balance balcony ball
bamboo banana banner bar barely bargain barrel base basic basket battle beach
bean beauty because become beef before begin behave behind believe below belt
bench benefit best betray better between beyond bicycle bid bike bind biology bird
birth bitter black blade blame blanket blast bleak bless blind blood blossom blouse
blue blur blush board boat body boil bomb bone bonus book boost border boring
borrow boss bottom bounce box boy bracket brain brand brass brave bread breeze
brick bridge brief bright bring brisk broccoli broken bronze broom brother brown
brush bubble buddy budget buffalo build bulb bulk bullet bundle bunker burden
burger burst bus business busy butter buyer buzz cabbage cabin cable cactus cage
cake call calm camera camp can canal cancel candy cannon canoe canvas canyon
capable capital captain car carbon card cargo carpet carry cart case cash casino
castle casual cat catalog catch category cattle caught cause caution cave ceiling
celery cement census century cereal certain chair chalk champion change chaos
chapter charge chase chat cheap check cheese chef cherry chest chicken chief child
chimney choice choose chronic chuckle chunk churn cigar cinnamon circle citizen
city civil claim clap clarify claw clay clean clerk clever click client cliff climb clinic
clip clock clog close cloth cloud clown club clump cluster clutch coach coast coconut
code coffee coil coin collect color column combine come comfort comic common
company concert conduct confirm congress connect consider control convince cook
cool copper copy coral core corn correct cost cotton couch country couple course
cousin cover coyote crack cradle craft cram crane crash crater crawl crazy cream
credit creek crew cricket crime crisp critic crop cross crouch crowd crucial cruel
cruise crumble crunch crush cry crystal cube culture cup cupboard curious current
curtain curve cushion custom cute cycle dad damage damp dance danger daring
dash daughter dawn day deal debate debris decade december decide decline deco-
rate decrease deer defense define defy degree delay deliver demand demise denial

dentist deny depart depend deposit depth deputy derive describe desert design
desk despair destroy detail detect develop device devote diagram dial diamond
diary dice diesel diet differ digital dignity dilemma dinner dinosaur direct dirt dis-
agree discover disease dish dismiss disorder display distance divert divide divorce
dizzy doctor document dog doll dolphin domain donate donkey donor door dose
double dove draft dragon drama drastic draw dream dress drift drill drink drip
drive drop drum dry duck dumb dune during dust dutch duty dwarf dynamic ea-
ger eagle early earn earth easily east easy echo ecology economy edge edit educate
effort egg eight either elbow elder electric elegant element elephant elevator elite
else embark embody embrace emerge emotion employ empower empty enable en-
act end endless endorse enemy energy enforce engage engine enhance enjoy enlist
enough enrich enroll ensure enter entire entry envelope episode equal equip era
erase erode erosion error erupt escape essay essence estate eternal ethics evidence
evil evoke evolve exact example excess exchange excite exclude excuse execute ex-
ercise exhaust exhibit exile exist exit exotic expand expect expire explain expose
express extend extra eye eyebrow fabric face faculty fade faint faith fall false fame
family famous fan fancy fantasy farm fashion fat fatal father fatigue fault favorite
feature february federal fee feed feel female fence festival fetch fever few fiber fic-
tion field figure file film filter final find fine finger finish fire firm first fiscal fish fit
fitness fix flag flame flash flat flavor flee flight flip float flock floor flower fluid flush
fly foam focus fog foil fold follow food foot force forest forget fork fortune forum
forward fossil foster found fox fragile frame frequent fresh friend fringe frog front
frost frown frozen fruit fuel fun funny furnace fury future gadget gain galaxy gallery
game gap garage garbage garden garlic garment gas gasp gate gather gauge gaze
general genius genre gentle genuine gesture ghost giant gift giggle ginger giraffe girl
give glad glance glare glass glide glimpse globe gloom glory glove glow glue goat
goddess gold good goose gorilla gospel gossip govern gown grab grace grain grant

grape grass gravity great green grid grief grit grocery group grow grunt guard
guess guide guilt guitar gun gym habit hair half hammer hamster hand happy
harbor hard harsh harvest hat have hawk hazard head health heart heavy hedge-
hog height hello helmet help hen hero hidden high hill hint hip hire history hobby
hockey hold hole holiday hollow home honey hood hope horn horror horse hospi-
tal host hotel hour hover hub huge human humble humor hundred hungry hunt
hurdle hurry hurt husband hybrid ice icon idea identify idle ignore ill illegal ill-
ness image imitate immense immune impact impose improve impulse inch include
income increase index indicate indoor industry infant inflict inform inhale inherit
initial inject injury inmate inner innocent input inquiry insane insect inside inspire
install intact interest into invest invite involve iron island isolate issue item ivory
jacket jaguar jar jazz jealous jeans jelly jewel job join joke journey joy judge juice
jump jungle junior junk just kangaroo keen keep ketchup key kick kid kidney kind
kingdom kiss kit kitchen kite kitten kiwi knee knife knock know lab label labor
ladder lady lake lamp language laptop large later latin laugh laundry lava law lawn
lawsuit layer lazy leader leaf learn leave lecture left leg legal legend leisure lemon
lend length lens leopard lesson letter level liar liberty library license life lift light
like limb limit link lion liquid list little live lizard load loan lobster local lock logic
lonely long loop lottery loud lounge love loyal lucky luggage lumber lunar lunch
luxury lyrics machine mad magic magnet maid mail main major make mammal
man manage mandate mango mansion manual maple marble march margin marine
market marriage mask mass master match material math matrix matter maximum
maze meadow mean measure meat mechanic medal media melody melt member
memory mention menu mercy merge merit merry mesh message metal method
middle midnight milk million mimic mind minimum minor minute miracle mirror
misery miss mistake mix mixed mixture mobile model modify mom moment mon-
itor monkey monster month moon moral more morning mosquito mother motion

motor mountain mouse move movie much muffin mule multiply muscle museum
mushroom music must mutual myself mystery myth naive name napkin narrow
nasty nation nature near neck need negative neglect neither nephew nerve nest
net network neutral never news next nice night noble noise nominee noodle nor-
mal north nose notable note nothing notice novel now nuclear number nurse nut
oak obey object oblige obscure observe obtain obvious occur ocean october odor
off offer office often oil okay old olive olympic omit once one onion online only open
opera opinion oppose option orange orbit orchard order ordinary organ orient orig-
inal orphan ostrich other outdoor outer output outside oval oven over own owner
oxygen oyster ozone pact paddle page pair palace palm panda panel panic pan-
ther paper parade parent park parrot party pass patch path patient patrol pattern
pause pave payment peace peanut pear peasant pelican pen penalty pencil people
pepper perfect permit person pet phone photo phrase physical piano picnic picture
piece pig pigeon pill pilot pink pioneer pipe pistol pitch pizza place planet plastic
plate play please pledge pluck plug plunge poem poet point polar pole police pond
pony pool popular portion position possible post potato pottery poverty powder
power practice praise predict prefer prepare present pretty prevent price pride pri-
mary print priority prison private prize problem process produce profit program
project promote proof property prosper protect proud provide public pudding pull
pulp pulse pumpkin punch pupil puppy purchase purity purpose purse push put
puzzle pyramid quality quantum quarter question quick quit quiz quote rabbit
raccoon race rack radar radio rail rain raise rally ramp ranch random range rapid
rare rate rather raven raw razor ready real reason rebel rebuild recall receive recipe
record recycle reduce reflect reform refuse region regret regular reject relax release
relief rely remain remember remind remove render renew rent reopen repair re-
peat replace report require rescue resemble resist resource response result retire
retreat return reunion reveal review reward rhythm rib ribbon rice rich ride ridge

rifle right rigid ring riot ripple risk ritual rival river road roast robot robust rocket
romance roof rookie room rose rotate rough round route royal rubber rude rug rule
run runway rural sad saddle sadness safe sail salad salmon salon salt salute same
sample sand satisfy satoshi sauce sausage save say scale scan scare scatter scene
scheme school science scissors scorpion scout scrap screen script scrub sea search
season seat second secret section security seed seek segment select sell seminar
senior sense sentence series service session settle setup seven shadow shaft shallow
share shed shell sheriff shield shift shine ship shiver shock shoe shoot shop short
shoulder shove shrimp shrug shuffle shy sibling sick side siege sight sign silent silk
silly silver similar simple since sing siren sister situate six size skate sketch ski
skill skin skirt skull slab slam sleep slender slice slide slight slim slogan slot slow
slush small smart smile smoke smooth snack snake snap sniff snow soap soccer
social sock soda soft solar soldier solid solution solve someone song soon sorry sort
soul sound soup source south space spare spatial spawn speak special speed spell
spend sphere spice spider spike spin spirit split spoil sponsor spoon sport spot
spray spread spring spy square squeeze squirrel stable stadium staff stage stairs
stamp stand start state stay steak steel stem step stereo stick still sting stock
stomach stone stool story stove strategy street strike strong struggle student stuff
stumble style subject submit subway success such sudden suffer sugar suggest suit
summer sun sunny sunset super supply supreme sure surface surge surprise sur-
round survey suspect sustain swallow swamp swap swarm swear sweet swift swim
swing switch sword symbol symptom syrup system table tackle tag tail talent
talk tank tape target task taste tattoo taxi teach team tell ten tenant tennis tent
term test text thank that theme then theory there they thing this thought three
thrive throw thumb thunder ticket tide tiger tilt timber time tiny tip tired tissue
title toast tobacco today toddler toe together toilet token tomato tomorrow tone
tongue tonight tool tooth top topic topple torch tornado tortoise toss total tourist

toward tower town toy track trade traffic tragic train transfer trap trash travel tray
 treat tree trend trial tribe trick trigger trim trip trophy trouble truck true truly
 trumpet trust truth try tube tuition tumble tuna tunnel turkey turn turtle twelve
 twenty twice twin twist two type typical ugly umbrella unable unaware uncle un-
 cover under undo unfair unfold unhappy uniform unique unit universe unknown
 unlock until unusual unveil update upgrade uphold upon upper upset urban urge
 usage use used useful useless usual utility vacant vacuum vague valid valley valve
 van vanish vapor various vast vault vehicle velvet vendor venture venue verb verify
 version very vessel veteran viable vibrant vicious victory video view village vintage
 violin virtual virus visa visit visual vital vivid vocal voice void volcano volume vote
 voyage wage wagon wait walk wall walnut want warfare warm warrior wash wasp
 waste water wave way wealth weapon wear weasel weather web wedding weekend
 weird welcome west wet whale what wheat wheel when where whip whisper wide
 width wife wild will win window wine wing wink winner winter wire wisdom wise
 wish witness wolf woman wonder wood wool word work world worry worth wrap
 wreck wrestle wrist write wrong yard year yellow you young youth zebra zero zone
 zoo

B Test vectors For Mapping Between Mnemonics and Entropy-Seeds

The test vectors include Entropy-Seeds (using hexadecimal) and corresponding mnemonics.

- Entropy-Seed:

00

Mnemonic:

abandon abandon abandon abandon abandon abandon abandon abandon

9f6a2878b2520799a44ef18bc7df394e7061a224d2c33cd015b157d746869863

Mnemonic:

panda eyebrow bullet gorilla call smoke muffin taste mesh discover soft ostrich alcohol speed nation flash devote level hobby quick inner drive ghost inside

- Entropy-Seed:

066dca1a2bb7e8a1db2832148ce9933eea0f3ac9548d793112d9a95c9407efad

Mnemonic:

all hour make first leader extend hole alien behind guard gospel lava path output census museum junior mass reopen famous sing advance salt reform

- Entropy-Seed:

f585c11aec520db57dd353c69554b21a89b20fb0650966fa0a9d6f74fd989d8f

Mnemonic:

void come effort suffer camp survey warrior heavy shoot primary clutch crush open amazing screen patrol group space point ten exist slush involve unfold

C Test vectors For Derivation of Master-Seeds and Account-Root-Seeds

ToDo: add `CoinVKRootSeedAut`

The test vectors include Entropy-Seeds shown in Appendix B and corresponding Master-Seeds. In addition, Public Rands for some randomly chosen Sequence Numbers are given. All item are using hexadecimal.

- Entropy Seed:

00

Master Seed:

d4fde696ab58de7b5097a1ec017e59d1f440342c1e278ea092d2c88cec6cc147
836245c200218a02dc0d00f8b875d19c7c5d89fdb2f85e085cdc7d45e7a94300

coinSpKeyRootSeed:

48d41357356d035064d5abb63b166413d362ef61f43bf333fffc5351dbbe977f
b383e743727f0360b5e16cd1e5fe7fbda816d20e15b869e1bc90acee79962673

coinSnKeyRootSeed:

56c658bad0035e16677d561b22ffb56e194f7160e40eb37466a8afbde5cb7bd7
6d4061b38a67f32b63b6e03f2b6b3e49a55671170990ee01be672bd4e0356632

coinDetectorRootKey:

bddd5c6a3049d96124666f009c0fb3af2b695fd28dd567b5eb25130b24788d20
21edeae091cc10ae5154878dc94098c22e60f4a663efaa98c6916d055f7802ca

coinVKRootSeed:

a2f568513bcd1610d4cca7f9770b84f7815c16a011a4fd27387f5e9067191f4f
6b823290c0d8a394a412b5d068d3385e5a71cefb541c529e74c7215058f0ba0d

publicRandRootSeed:

e3a7966dd6e343509b682be724be4035ff8ac480775f4dd72559aa63f6124e56
9d5341321c43a68eec7c1cb9232cae7bd643da63c9306cdba7e3415315bd48b2

Public Rand for Sequence Number 887699001:

d832f4df91d59bf2c26c3dc75975c1a53b484089726bd4a82f10b46cbb6f56af
094374b37d0953350af428153e99873a10265269db2b9ce4b30b0c292f7a3136

Public Rand for Sequence Number 2172391158:

dfa3ecb99d8c6ba43cd0fa4d778f910333125ab47833dc83279785dc87ee634f
dd0562d6aa43004f6db3f94efcbdae4ace5f1b4b738de815c299e68aacf06928

Public Rand for Sequence Number 2474215398:

10d5530ffa173f24bdc2a2970e42403d55a676d2d29649f56f9e47458ff04681
c1cbc08d6f15bb6fa34dea9d273b50e38da3047c8652dabe65d3f050e8ec6224

coinValueKeyRootSeedAut:

d8e0a5709e36a1b2fdb18b3c592859f23ca9743911396231be1b97ca6f85b8b1
24b30013773b5b0e44ba65dd566a81f6697e52e2897b0375f6761519e19da4fa

Public Rand for Sequence Number 2400773142:

1823c3c895c419aa32b993d788e87a722358d24dd57e05f0ab0c749e39a5be48
e0075fdc07038eb9e1adbc274dd7cfa3d313ec97fcdefaff56aa9770d6046fd6

Public Rand for Sequence Number 94650305:

d2e984e12d818ba294d0571722d51d300ce32549016f0bc93cb1850b811d9b61
9f34bf3aa909c7d4a65253202fe11148cefd57a6b42dcb93891d94123e1dc474

Public Rand for Sequence Number 3668972267:

bc160f59c1885986b185540d8628485645ece76952bef6afde49ae7a5c1b5d89
f8e3118ecf0d4f3d46ddae7cf95b88c9a9b49fe170a0696a6de9027782bde90e

coinValueKeyRootSeedAut:

81a723c11135ee7c4308ba47149c419ebb069110b44f986b656f648429e7d2fd
d2f3e3cc7855b504bf6d52528e2e5014c9c15375d23172f12eac9ce8a92a39f8

- Entropy Seed:

ff

Master Seed:

4abcc8067efc753596a079a6b2c81bbff92f65c0bae6369f7572f8cfe881161
b7fc009b5591849caf79d7896c1963f2a39cc57049247173ecb037af59f97718

coinSpKeyRootSeed:

5b14a23daa5936f5828e5a7117dc1fd76c4aab6a23a57de8419ee52ca253021a
aa6a4ccd6468e57ebc5ffb2e672171cd0216febd76fe1e8ab07fe16ca2775d71

coinSnKeyRootSeed:

a46e06fa6d94789a147f22d4862e39b40a0d2cc05ca26738705cfe2d399d3c87
b71f8d99dfc3900281a3b569a5ed2f54f0ee4d0d807593599bd2bedc170d4b7c

coinDetectorRootKey:

ae609c4c2ee8d33395452d0e03cb6be58a939ca6f28095235622cf542a51f407
b0a2f31a8bdf957960518be5a1e2b8c0be53d98b75947525754a6820211ce47
coinVKRootSeed:
7c32ecbc44e49533ce29946314848d60b5e432080a3de6db3b0abaab983a4a55
d40744b86a10817347e89588ab622bc9a8369b3958d7f42b49b263992238cc38
publicRandRootSeed:
110851bc774a02d79b8df4896fd4e497befc1c7bb8d447d282fe94bb13574fb4
ad68bca6337dd8575e907d4a94ef4dc2d180c9f4e99d567b19ca2d9e4459a214
Public Rand for Sequence Number 4151069018:
577ca99203857d5b233abafbee3327e7ea2d4af5d0d64014a4ac047a8f2d911c
698bbec62e8d19e5738f615730b594e86f0098f02b29ef2d88ae3a9e92253a7d
Public Rand for Sequence Number 810230981:
1ed8f5a3bbdce1201782376807364ff46424bcebd43a01f6b8074c97c4c4cc57
6b497a8931ae874d08ae5aa04b4420d771e79a2a97a2636ee20d880424df6b86
Public Rand for Sequence Number 1205304842:
ef6df803d5b2ab20050752cdc59f7c4e16f11172e86782092dc15413484026b7
aed522747178ddcc5b1b57f3067636b9392a41117a2975216d7bd34448eb82c5
coinValueKeyRootSeedAut:
59b9400f1448564bccaa1a4c679810844fcaef76a3e6fc6b75ba39e18485025
789b917766cd589e3173e950a9b0bc4db78b9fd8a150bb46be1ca795d4e4e415

- Entropy Seed:

68a79eaca2324873eacc50cb9c6eca8cc68ea5d936f98787c60c7ebc74e6ce7c
Master Seed:
ec243d0407f4731085ecd7bef047db836b675e8622c937454c3313a8129d6733
7f7fb648ed71a8e7214f38779edf6c29bffb13fff3b42b7c8fc64ace75381c7f
coinSpKeyRootSeed:

1a8efaa57603373aa226bd25a734b355570f777dd3a50b113cdea7ce3902537c
a80682f918a4c26796b4f3a2c64ceeab877d091adac6b1083237fdef19a81d5b
coinSnKeyRootSeed:

b3b6835aa6dcbe526ff030bca5e71a825a23ff1605902f39e392971df493c30b
2dd3c74082a97351bc3b0a152e1563f724ddbfbf2adb8c2906e00c77d235b5515
coinDetectorRootKey:

becde156316d7470eacdc12d01452fc56aa27bb6f1a7ace1ba1a26da3752aaba
f819d97609513fc11e4f8e34a631e062d6467626d47917e0438fdf3c3eb59ccb
coinVKRootSeed:

ed54c3155e21d98f316fb521c92b09679e763e9b80695d62ab76cc17c57ece57
86b54d83494f0e81967ff37c772f7dbcb830069665ad2fc95b83acfe6f9d3ea7
publicRandRootSeed:

f4b510fce8ad25bb3e8e2e3a371a813cd8557d1930284c1b54b301b481965232
e4e36a91c19c622eba1625926d4df7654fbe51036d0af31bc6a93e1fd9f09932

Public Rand for Sequence Number 1386746106:

d6ed41486c223a9bc0877b1465d2ea412ecd7dc5d4aa6e7f5caaf65c1265d7f9
30e55c7c626715d7a9c614b00be192fbbae3293f8440e553f079ab35b24510db

Public Rand for Sequence Number 757380707:

2245d0ecf4e2ddffbeb80fbef9b507e35561acce80451c85e548eb431123ecd7
032d34b182a381f5356db1dd5e92d2c0b40a36cb3e3a98b6ecdff81ae65f6e1e5

Public Rand for Sequence Number 1590126875:

f069a5a75a4563e6e190e8f8b1443341ce3272b68cff99ff384ac47478541dc5
b061fcc78fd750fa3beea3a6e86530691cada8af012d1c12b34344bdab0caieb

coinValueKeyRootSeedAut:

a230714e5df98c856dedf7a387ac486b5ffa8fabf562d3563fd45b27ef16322c
d15e2b3726300215b4c597b3bed385060a0a280c009cd5bbbbbefbc041e0f56c1

- Entropy Seed:

9f6a2878b2520799a44ef18bc7df394e7061a224d2c33cd015b157d746869863

Master Seed:

cd33cfeaec5a48f52f485fea5e9637889609aa64cab852d80aa258e989071c60

7ae9c1f3b37bafef1284d6e06811bd4dfaf1e15a2bb9f895ef7a7584d0ac76903

coinSpKeyRootSeed:

88087c0f9482aed16a34017404c45541055be1624993b727d7296be581fa4d11

05044b680ab2f00f8dfdb8f6f7712580082e6196e09179b27b529e9b1310d8fb

coinSnKeyRootSeed:

a02c93c369274f3d866e98cf6980dfbac25587cbbb4c8b5f45c3f62e2e94b214

913dcbe832a2908511b9d1373feb6ef1df42d147bc11a9e199d3d51638ca771f

coinDetectorRootKey:

babca8531d788cbec214e9a3180301479b2df429705aeb8e454630c5d54b0fcc

dc831f5a1c12bc3c8d92ac3ad37e0c195a7ea3041ff39d3bb855f02eadfb57c7

coinVKRootSeed:

b2c572f0108cc12627d17f8b93e7dd86f0545ee568309c2c93ecf4d514b2501b

b1e596e2bd4005d3068fe96f430b5564cd393aabfd792c40554bb9a09c8e1406

publicRandRootSeed:

e02c06fbe97da236dcfb098b256f4edf4727f0b19cf5f0afe7f4ed9ea524dab7

1adbfd0536267e7c8ebe0909d4820de92f5c37d3bfca0aeb1c34a210d9be189d

Public Rand for Sequence Number 1349967783:

1ab49a08f7dc26c37d10e78d9afdd81c08601391b31a81add08ffdf937444953

1e16876deeaac3d1d896af37f26d0e8b4845205e11dc508c2c4789f0c1150aa2

Public Rand for Sequence Number 2895057325:

3eea3875fb600d6adf5134bb61ee794cd250441315dab228fef0ed070dd172e2

9382e62c1e09d1dd4f6b60113effed91d3a9c98c606d05307aeb644fc2c62402

Public Rand for Sequence Number 335187303:

627796c337060b19ec1c7178b4630f08062c6259318ad20b04d7f3308dff00b9
416dfc75652cf3e63d63bec1bdf99599edc22c78003a885d1c6c42a94348d5bf
coinValueKeyRootSeedAut:
3acda1d4b30360b2a200300238638f12a4e037eb2ab86e48b290afbd02bab60b
d0b5e4e7bffa8f24359acd5c5ce53ec1b3328ed3f83ab6a252cab464228ca50f

- Entropy Seed:

066dca1a2bb7e8a1db2832148ce9933eea0f3ac9548d793112d9a95c9407efad
Master Seed:

7242f5fd0c0b8df7429d60a7a757687eb486e4f45f50f675a9d23fa034607f3a
0e3a7c12a6d7eb0270bfc bdea0e045f8562f2a312765cdbdee4a31ee825634e3
coinSpKeyRootSeed:

70f94f5a37f48e64c8ccb91779d131c6fbd25f44e3a463a196f21de7502f2c5f
5aa9461361af03d1fc560df9ccb96c1afaf1758e9112a01d555cdf42ead3b7de
coinSnKeyRootSeed:

c63f57958e0b02af215837e7a9f7da3f6830ca849e9cc586a1016f59d518ff2d
b832d9b3a70fa8f46f91d63af9ecf4681a639683c8b178a3e4cc4d0ad2e2fdab
coinDetectorRootKey:

d1d62e7f9bee336a61e360785bf1fca64dc9b90602b99abeaff44f48c3dfdc74
14d2a23dca36c929cbf093b27e63d6797797ffb8ff2c3862c90d3dbf4fa9a924
coinVKRootSeed:

ae85d5eafd21843dc6f7518861982107fa63a2a95ab4eb09d0c7b50e5d4f3f9a
afed0f23b2a1958c27f1f5d0d27e23f1716a9093abe2430de1f028f51a397b73
publicRandRootSeed:

dff3840573a631233b893c59c33518324f84c6da4e13cc19dff3aaca66285b09
14dee95e6e8b6f2d1cba3d8ac46beebe09f7b7fadfd1a358cdd5bc26aa84a8d
Public Rand for Sequence Number 3037234089:

4fee704d8d687a026cfd98d1a19dc0cc1d7298b350f9115a722e4388d204e903
6c8b698e2870ca8e04d5fc7716764ad16278022179d05bca86e8a5fdb0941b2c

Public Rand for Sequence Number 3952040230:

77a47010286d7b308e0ec22dfac6768f3a44a430ca54e11fcf72be6cf3379b4f
a30f16a246e404166d2b13c02dc16e9300d84baf3986bd10d2eae731d1a43d52

Public Rand for Sequence Number 997927870:

d58a8e1f7e19a4f5ed092636be3477284a24d7b9d4d928776512123d29194579
6f3e286f097bac617faf946d5d55286eb7c670a26f382b0a6e1e27092f36a291

coinValueKeyRootSeedAut:

4b4c30616d2a0b03fbae33ac9e172d4150a236b4ac416b7e11518b4b5bebd600
dd3fbf9fe01759e1e56371f2dc9e14becc7058b58494fa1be57affa6c89af73c

- Entropy Seed:

f585c11aec520db57dd353c69554b21a89b20fb0650966fa0a9d6f74fd989d8f

Master Seed:

e82bbf95ad7518c82df7165f575d07634edb48d9eb63e444a104e625dbca0e16
dfc15e5004985e94597fb67b72fd9c35eb7621a654d76c9189c47b72c33ec58c

coinSpKeyRootSeed:

9f3e551aeafc8304816980d6c1e3fd4c10d24635df07900cf5dddfbe72a1e98d
8308fc4cb7acf827e21136d352b92ba79bf8bdbbca45fc367817f65ec39a0367

coinSnKeyRootSeed:

3f185e15f879378bc7df9c604fe966024f2aa46101eb9aecf873a556126ad924
c9d8f1bce9a28238989404924cf1ee571b8f06015843010a38ed1fca6d9263e2

coinDetectorRootKey:

0ea3fc5da6fff18dfe3dfc9a87a394ac7543fa2f0cf26e5d8b7e2ec7bd3f7ac9
6d063217f60faa92d1de41b2504684219852de154b81ab3a507ba02562957591

coinVKRootSeed:

a808ef8b153c4f53ad1dc6a64c2c2fea2998cdb41c2abdf1e4659bc07a9369a0
048f4b5fd82385f7dc31780e789e6eed020cff9d0eacddd0faefe6d86e61e4ec
publicRandRootSeed:
8c8f88993d54e24c2d676042df3a3839d406d7d0d5166ca6d05a52d1aad0b04d
c539f9194dcf492afce2ce1711b0c8e92e0660a54698c94c86835219304084c4
Public Rand for Sequence Number 2282586443:
13abd9c51cf91e7ffd7f90daf2c1bf3a1210bba87d75e4cfe09fa285a08c4079
aea3efdcaa83f7a83df7a265d709bd6f817670a4791585e7011b461912479788
Public Rand for Sequence Number 2058736541:
2b51f39a3e6a049a5ab3d856962869580f889287758f32ebe7a5185f620da93f
c4df5e62ce2d0b11295e3ad0e707df306c3c21ab45fe718c431e7c6ef8eb6c73
Public Rand for Sequence Number 3650681307:
b3923c0758fe02869413a515dbc8fd129b2b876f472c8db3ec114455fb21d37d
3d97cf54f3f7540e182a610507e04dce7056bf8b294bb2bd887c22b7c9c742ca
coinValueKeyRootSeedAut:
d5d86a1e486ea52013b2a6967a65b16370b1cee6748ec92a6cd5bf45726e2758
e2cf9f8b7935c825c472ee93a5895ba2ce697d021e84f0110f849604b7cc8b7b