mobilefish.com

IOTA TUTORIAL I7

Create & validate a signature

KNBJDBIRYCUGVWMSKPVA IRCBYHLMUTLGGAV9LIIPZSBGIENVBQ9NBQWXOXQSJRIRBHYJ9LCTJLISGGBRFRTTWD
ABBYUVKPYFDJWTFLICY KNVMSQERSYDPSSXPCZLVKWYKYZMREAEYZOSPWEJLHHFPYGSNSUYRZXANDNQTTLLZA
MDVJVVJNTNPJBPTTPHL YKEQHDZPAZ9XOXMCJQUUXGTU9NBSUZQUXSSMYQCOLUUEXZZA9SGWDAELZJLNNZVMPT
FTEPNHMKDMXMMHDAJ XHEAEWKQNESBWGM9TULSPXKGSQXVRM9DRFARRGSUVGZNWOFCSSCTDTQORPKSAVTVW9
JIDQGQRJAAICYSWF JIHECEKOKLZ9NCWDENYFGYQLQKIZOPIZMNUOEKHIZBAVOTCT9PXOAKIWXUHFZDTNIGJ
AYQKSTHEHMQZQGB WDRBDOSVVWKWEUEVDYSRMPJXPLDOGDBTXPNKYLBMYUWZFELTRUYKFLUPCVVQJZU9U9IM
MTFKIHHTZJTCQE FWZVAWKEJURHGOUXDMCRRGOFTTHXSIAVXLZZJIDLBVNBSYYZEH99ADGSPVLCVMU
YUZMAJZC9IAGM LJGT9LXMODDJTVIH9MPF9BNSLGXUDG9VURODUSNJSRRYCSRRFJCAYRDUP9KCZCQ
GIKQHQRIWSLHP TJRIQUFPGAGCOJCUUJZVEWICYZZBJUROCRWOLRUOPRMXBCIFFTWHEEJRJNUXMYZW
WZ9ITBLYRXCMS UJPILVFKDZLCJKHWCSMC9WFYSH9JKJGQBHRSKNAONLMPTKPEXKCPLXOSNCRVSCXGR
PZMNALCYLRMM EMEIBNYEHFUKHLOVXLPIUUQPLAUZNDRMTXMZUQABEM9MNV9ILLMSMFHHJCDBJ9FXDS
LWIJYFKHKEEQ VHWIUGDEXSYWXXGSUHHJNJBFHAXGSQMSOCUZZQW9JNJVTYDJAM9YWNDAOLZPPWAFOY
NJPIDRSJVQTMTB 9IPP9QHOZIFIVBRPXPBBWOIOJKXY9ZMZEOXTB9NVTHUWGSEIJEIMTXDAHEABRKWRLN
KNBJDBIRYCUGVWMSK KIRCBYHLMUTLGGAV9LIIPZSBGIENVBQ9NBQWXOXQSJRIRBHYJ9LCTJLISGGBRFRTTWD

v1.0.0

INTRO

- In this video I will explain how a signature is created and validated.

HOW AN ADDRESS IS CALCULATED

- Before I explain how a signature in a bundle is created and validated, it is important to understand how IOTA addresses are calculated.

- In IOTA tutorial 9.1, I have explained in detail how the key, digests and address are created.

- Here is a short explanation:

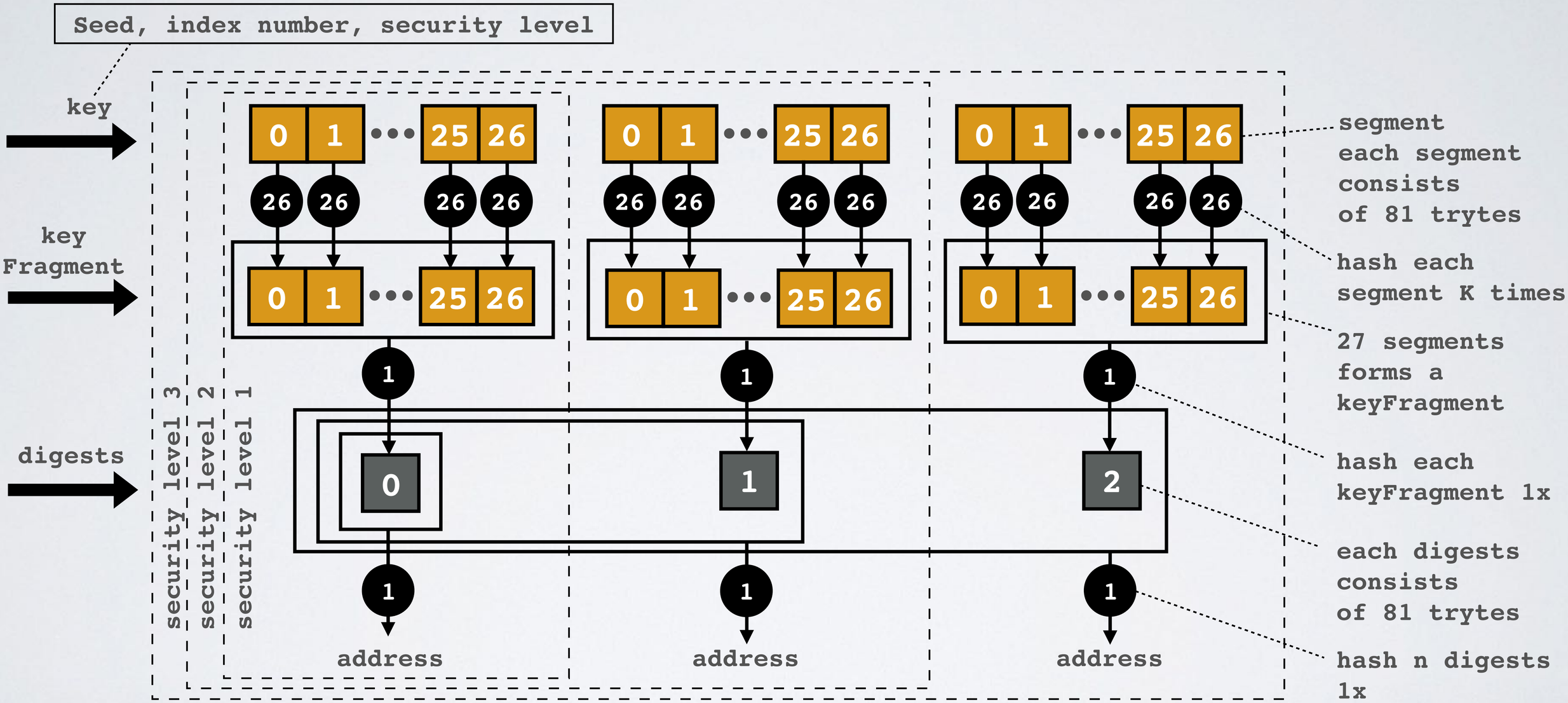
  - A key (= private key) is generated using the seed, index number and security level.

  - The index number is an integer (0,1,2 …. 9007199254740991) and every address has a corresponding index number.

HOW AN ADDRESS IS CALCULATED

- The generated key is divided in segments of 81 trytes each.
- Depending on the selected security level (1,2 or 3) you will have (1x27=) 27, (2x27=) 54 or (3x27=) 81 segments.
- Each key segment is hashed 26 times.
- 27 key segments together forms a keyFragment and each keyFragment is hashed one time to create a digests.
- Depending on the selected security level, 1, 2 or 3 digests are combined together.
- The combined digests are hashed one time to create the address.

CALCULATE ADDRESS



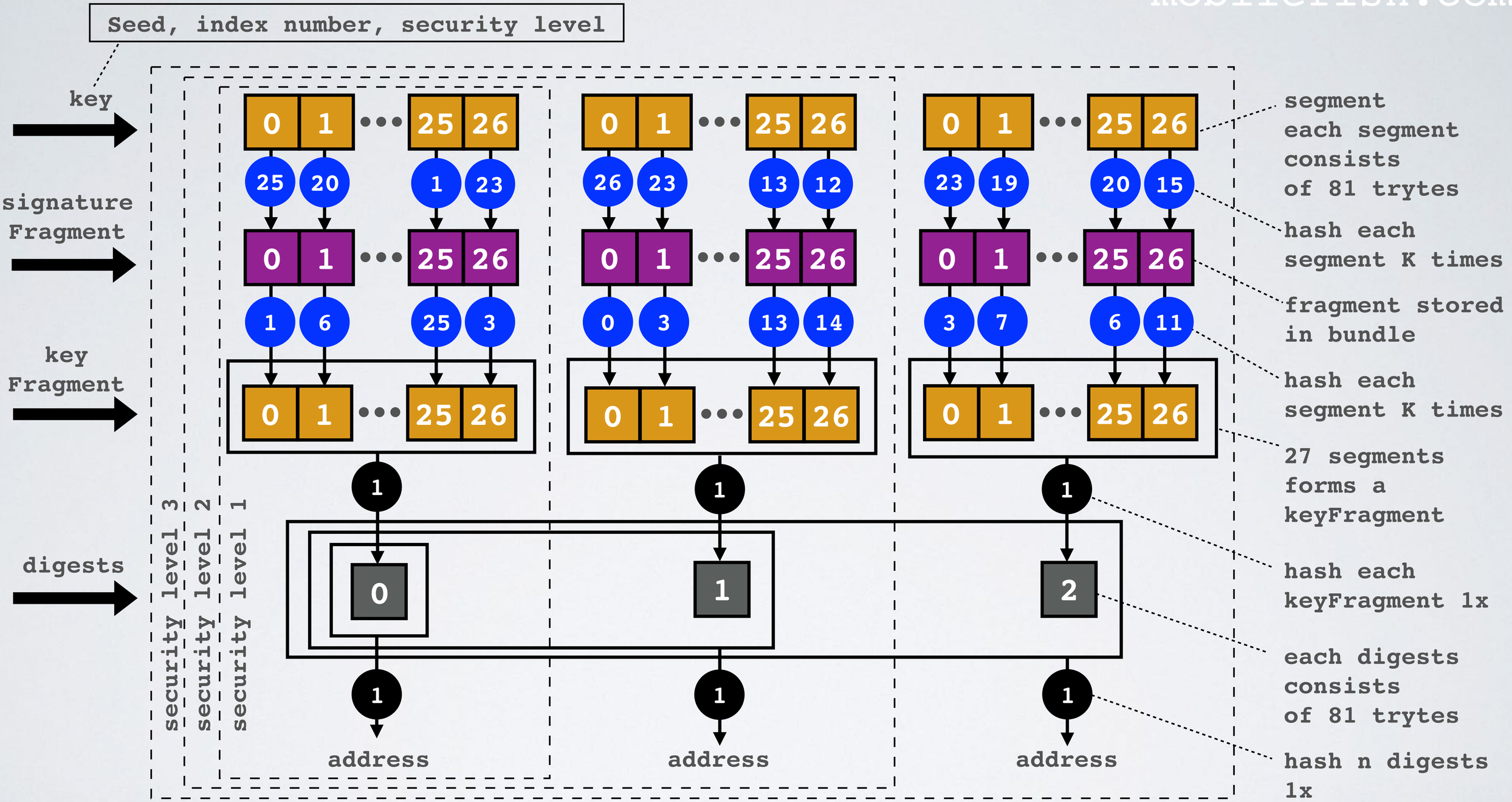
HOW A SIGNATURE IS CREATED

- The generated key is divided in segments of 81 trytes each.
- Depending on the selected security level (1,2 or 3) you will have (1x27=) 27, (2x27=) 54 or (3x27=) 81 segments.
- To create a signature:
  - Calculate the normalizedBundleHash = normalizedBundle(bundleHash)
    The bundleHash is explained in IOTA tutorial 15.
    The normalizedBundleHash is explained in IOTA tutorial 16.
  - Convert each tryte in the normalizedBundleHash to its decimal value.

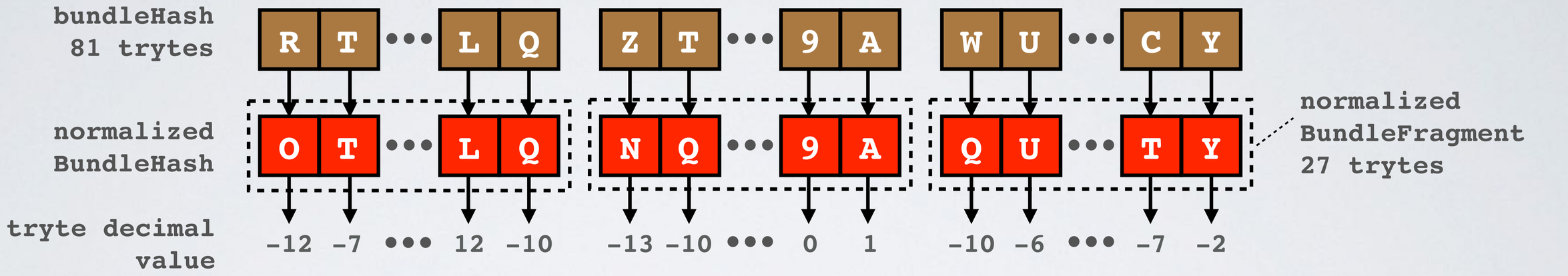
HOW A SIGNATURE IS CREATED

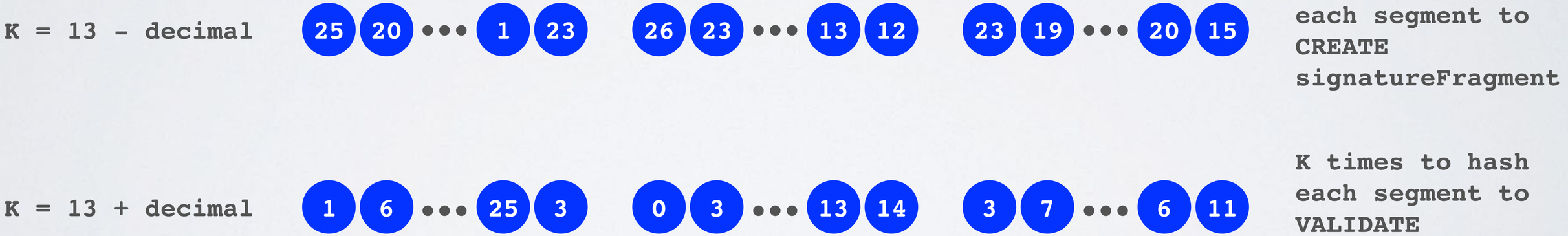
- Apply for each decimal value the following calculation: K = 13 - decimal value.
- Each key segment must be hashed K times.
- Combine the 27 hashed key segments to a signatureFragment.
  A signatureFragment has (27x81=) 2187 trytes.
- Depending on the selected security level you will have 1, 2 or 3 signatureFragments.
- A transactionObject stores a signatureFragment in the signatureMessageFragment field. This field size is 2187 trytes. Additional transactionObjects are needed to store the remaining signatureFragments.
- Note: signatureMessageFragments is the same as signatureFragments.

mobilefish.com



CALCULATE NUMBER OF HASHES





HOW A SIGNATURE IS VALIDATED

- To validate a signature in a transaction bundle:
  - Extract all corresponding signatureMessageFragments from the bundle. Depending on the selected security level you will have 1, 2 or 3 signatureMessageFragments.
  - The combined signatureMessageFragments is called the signature.
  - Extract the bundlehash from the bundle.
    Note: The bundleHash value is the same within the bundle.

TRANSACTIONOBJECT EXAMPLE PART 1 OF 2

- A transactionObject in a bundle (security level 2).


```
{
    "hash": "YDDQ...A9999",
    "signatureMessageFragment": "JHAK...MVGY",
    "address": "HRKD...XKHX",
    "value": -3,
    "obsoleteTag": "999999999999999999999999999",
    "timestamp": 1515494426,
    "currentIndex": 1,
    "lastIndex": 2,
    "bundle": "RTGX...LQCY",
    "trunkTransaction": "WVCLP...99999",
    "branchTransaction": "DOXV...X999",
    "tag": "999999999999999999999999999",
    "attachmentTimestamp": 1515496571334,
    "attachmentTimestampLowerBound": 0,
    "attachmentTimestampUpperBound": 3812798742493,
    "nonce": "AZ999IOB99999999999999999999",
    "persistence": true
},
```


2187 trytes

81 trytes

TRANSACTIONOBJECT EXAMPLE PART 2 OF 2

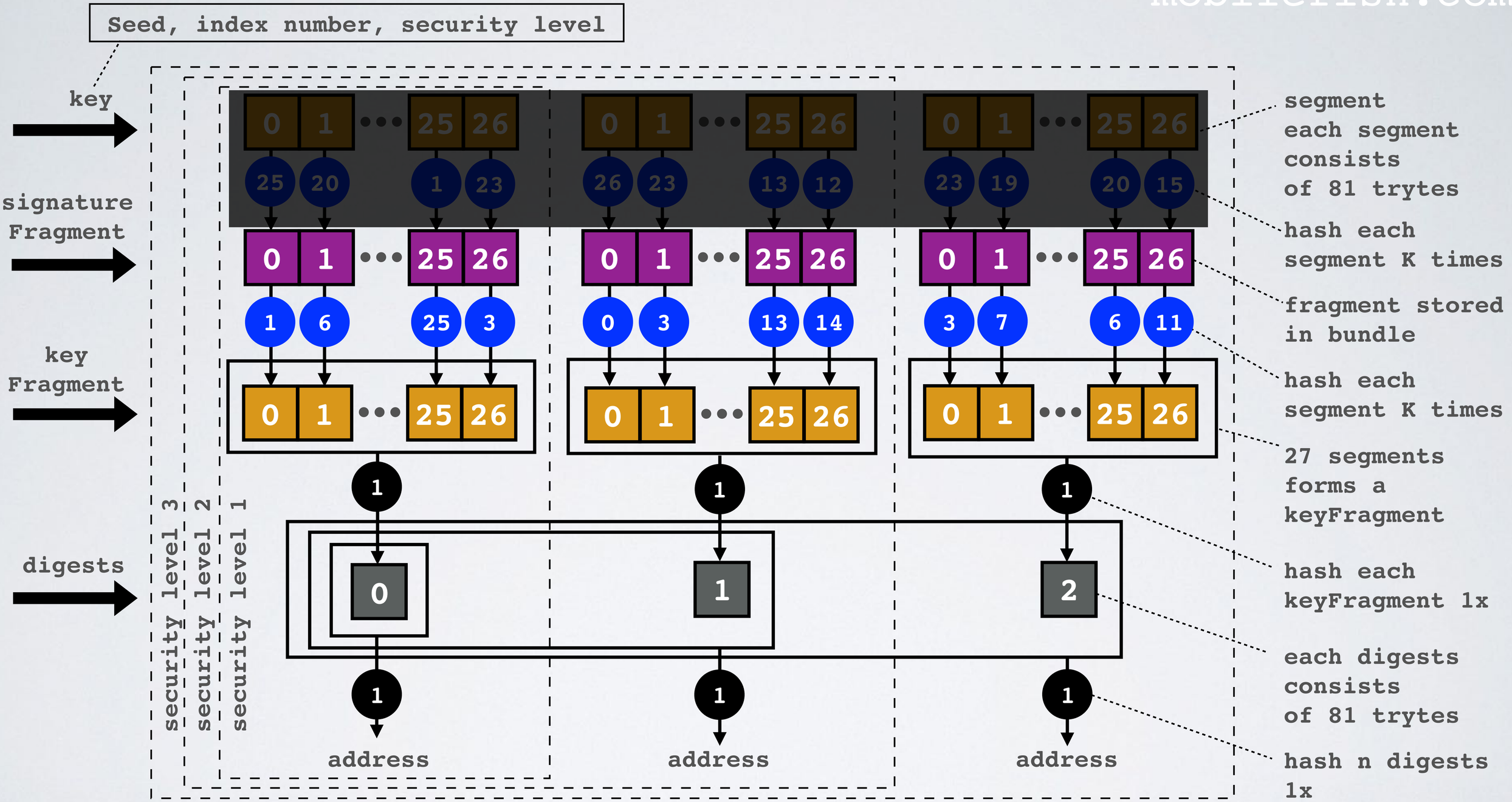
- A transactionObject in a bundle (security level 2).


```
{
    "hash": "WVCL...OUCPM",
    "signatureMessageFragment": "ZRUE...AWFY",
    "address": "HRKD...XKHX",
    "value": 0,
    "obsoleteTag": "999999999999999999999999999",
    "timestamp": 1515494426,
    "currentIndex": 2,
    "lastIndex": 2,
    "bundle": "RTGX...LQCY",
    "trunkTransaction": "WVCLP...99999",
    "branchTransaction": "DOXV...X999",
    "tag": "999999999999999999999999999",
    "attachmentTimestamp": 1515496561769,
    "attachmentTimestampLowerBound": 0,
    "attachmentTimestampUpperBound": 3812798742493,
    "nonce": "JA999ISA99999999999999999999",
    "persistence": true
},
```


2187 trytes

81 trytes

mobilefish.com



HOW A SIGNATURE IS VALIDATED

- Validate a signature.
  https://github.com/iotaledger/iota.lib.js/blob/v0.4.7/lib/crypto/signing/signing.js

```
var validateSignatures = function(expectedAddress,
signatureFragments, bundleHash)
```

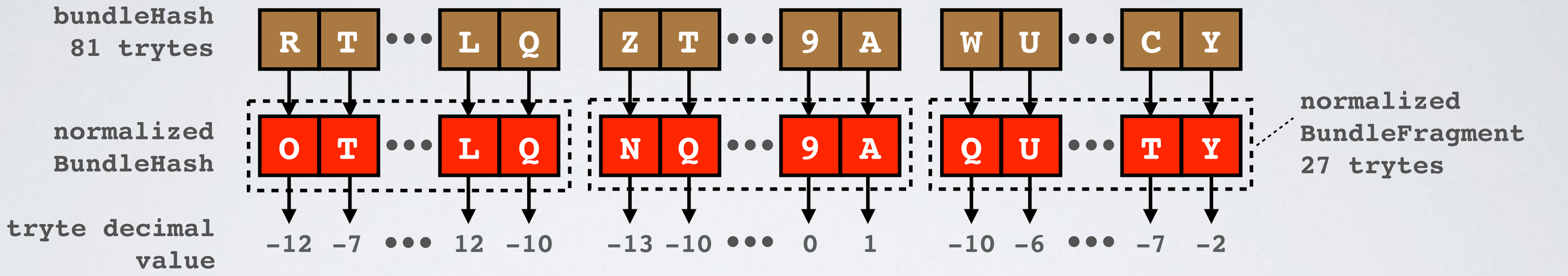
HOW A SIGNATURE IS VALIDATED

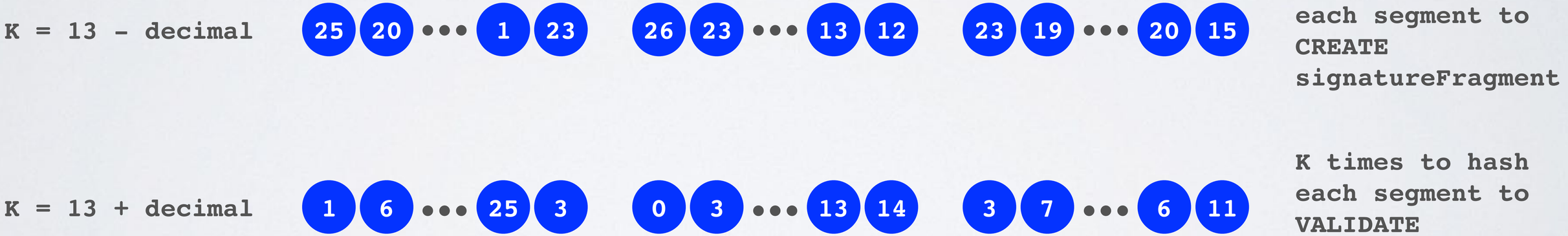
- A signature consists of 1,2 or 3 signatureMessageFragments (= signatureFragments) , depending on the selected security level.
- To validate a signature:
  - Calculate the normalizedBundleHash = normalizedBundle(bundleHash)
  - Convert each tryte in the normalizedBundleHash to its decimal value.
  - Apply for each decimal value the following calculation: K = 13 **+** decimal value.
  - Each signatureFragment segment must be hashed K times.

HOW A SIGNATURE IS VALIDATED

- Combine the 27 hashed signatureFragment segments to a keyFragment.
  A keyFragment has (27x81=) 2187 trytes.
- Each keyFragment is hashed one time to create a digests.
- Depending on the selected security level, 1, 2 or 3 digests are combined together.
- The combined digests are hashed one time to create the address.
- If this calculated address is the same as the address taken from the transactionObject in the bundle, than the signature is valid.

CALCULATE NUMBER OF HASHES





CREATE AND VALIDATE SIGNATUREFRAGMENT

- Create signatureFragment.
  https://github.com/iotaledger/iota.lib.js/blob/v0.4.7/lib/crypto/signing/signing.js

```
var signatureFragment =
function(normalizedBundleFragment, keyFragment)
```

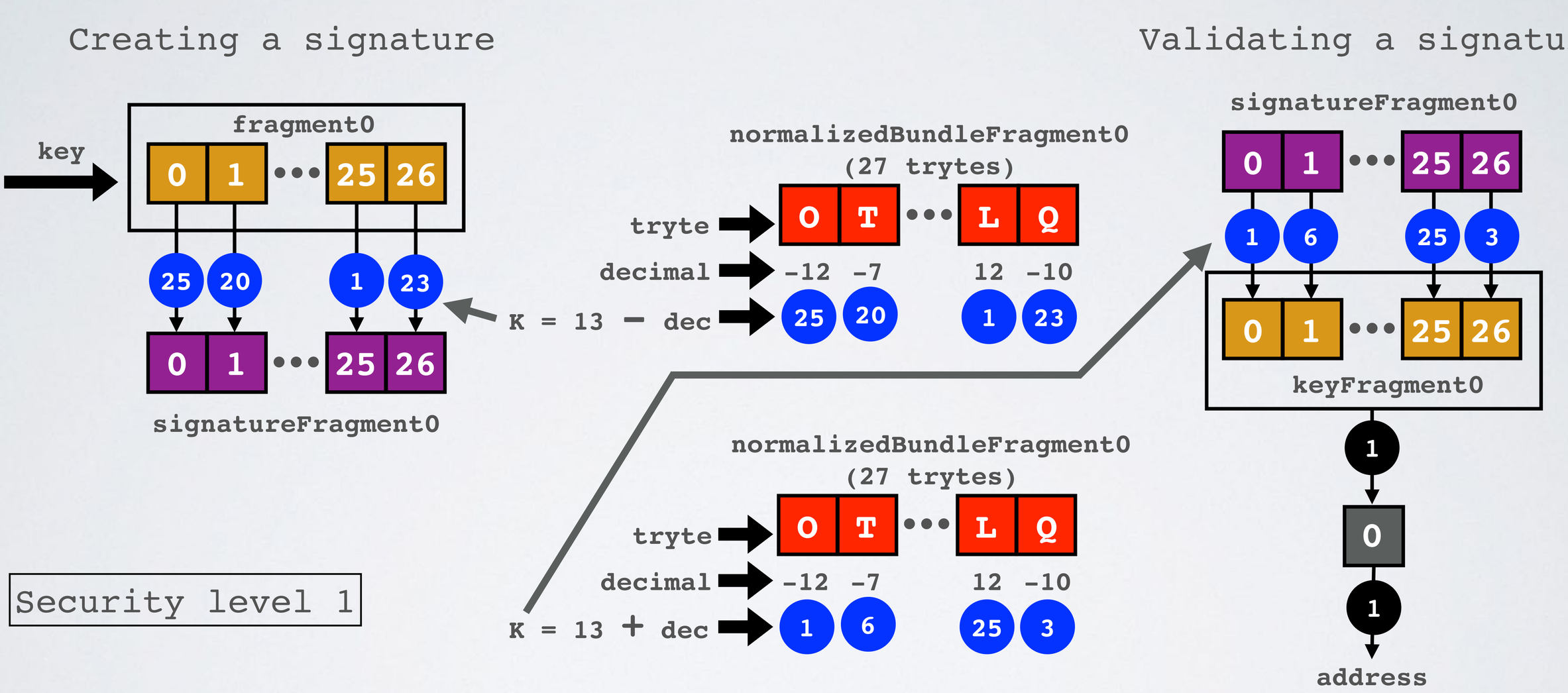
- Validate signatureFragment.
  https://github.com/iotaledger/iota.lib.js/blob/v0.4.7/lib/crypto/signing/signing.js

```
var validateSignatures = function(expectedAddress,
signatureFragments, bundleHash)

var digest = function(normalizedBundleFragment,
signatureFragment)
```

mobilefish.com

CREATE AND VALIDATE A SIGNATURE



Security level 1

IOTA TRYTE ALPHABET

Trits	Dec	Tryte		Trits	Dec	Tryte
0, 0, 0	0	9				
1, 0, 0	1	A		-1,-1,-1	-13	N
-1, 1, 0	2	B		0,-1,-1	-12	O
0, 1, 0	3	C		1,-1,-1	-11	P
1, 1, 0	4	D		-1, 0,-1	-10	Q
-1,-1, 1	5	E		0, 0,-1	-9	R
0,-1, 1	6	F		1, 0,-1	-8	S
1,-1, 1	7	G		-1, 1,-1	-7	T
-1, 0, 1	8	H		0, 1,-1	-6	U
0, 0, 1	9	I		1, 1,-1	-5	V
1, 0, 1	10	J		-1,-1, 0	-4	W
-1, 1, 1	11	K		0,-1, 0	-3	X
0, 1, 1	12	L		1,-1, 0	-2	Y
1, 1, 1	13	M		-1, 0, 0	-1	Z