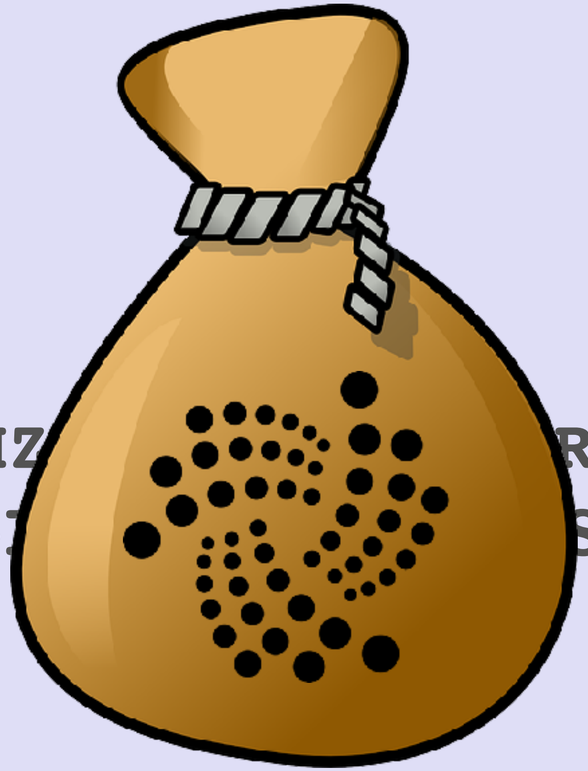
IOTA TUTORIAL I5

BundleHash



NBBKCKPFECRKCDIBKSTHZ RK9FECFK

PTTSWTLUWG SABRD9KDVFJ9GT9CKA

INTRO

- In this video I will explain how the bundleHash is created.

BUNDLEHASH

- How the bundleHash is created, see:
  https://github.com/iotaledger/iota.lib.js/blob/v0.4.7/lib/crypto/bundle/bundle.js
  `Bundle.prototype.finalize = function()`

- To create the bundleHash the following transactionObject values are used:
  address, value, obsoleteTag, currentIndex, lastIndex and timestamp.
  See prepareTransfers demo (step 6):
  https://www.mobilefish.com/download/iota/preparetransfers_demo.txt
  Note: The currentIndex and lastIndex are array positions, that is why you will not see these fields in step 6.

- For each transactionObject in the bundle:

BUNDLEHASH

- valueTrits = convertToTrits(value). The length is always 81 trits (padded with 0).
  Note:
  The number of IOTAs created is: 2,779,530,283,277,761
  This number converted to trits:
  1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
  That is 33 trits which means valueTrits can hold more than enough trits.

- timestampTrits = convertToTrits(timestamp).
  This is the number of seconds elapsed since January 1, 1970 00:00:00 UTC.
  The length is always 27 trits (padded with 0).
  Note:
  27 trits (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1) converted to a decimal value is: 3,812,798,742,493

BUNDLEHASH

- currentIndex = bundle array index
  currentIndexTrits = convertToTrits(currentIndex).
  The length is always 27 trits (padded with 0).

- lastIndex = bundle.length - 1
  lastIndexTrits = convertToTrits(lastIndex).
  The length is always 27 trits (padded with 0).

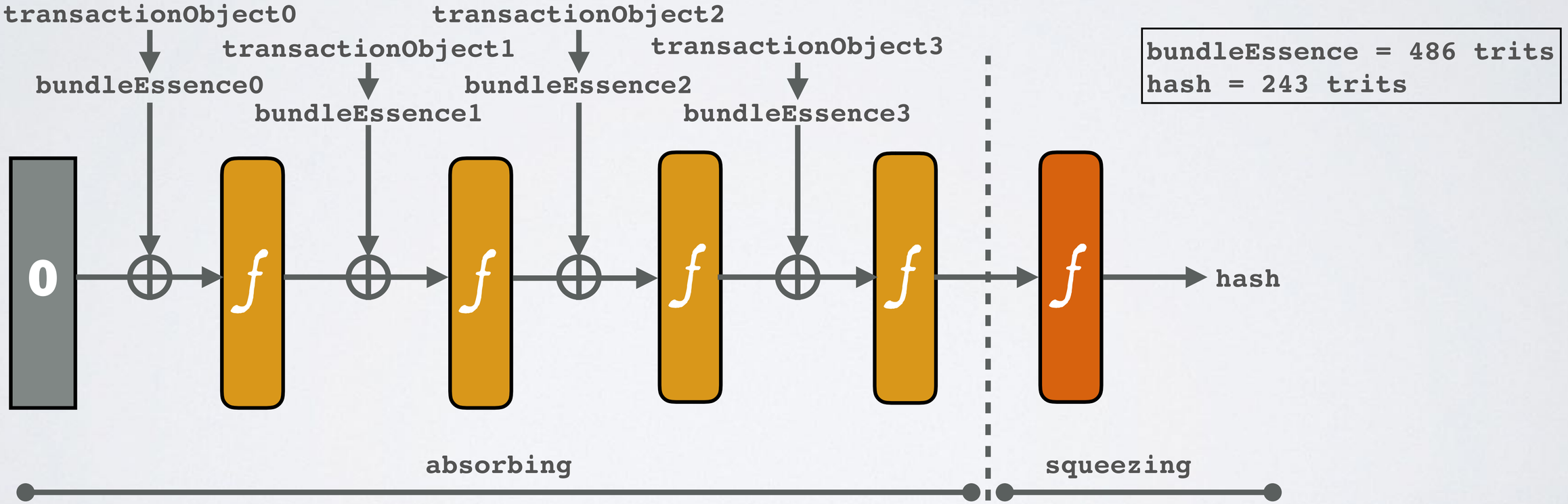
- bundleEssence = convertToTrits(**address** + convertToTrytes(**valueTrits**) + **obsoleteTag** + convertToTrytes(**timestampTrits**) + convertToTrytes(**currentIndexTrits**) + convertToTrytes(**lastIndexTrits**))
  The bundleEssence length is always 81 + 27 + 27 + 9 + 9 + 9 = 162 trytes = 162 x 3 = 486 trits

BUNDLEHASH

- Use the cryptographic sponge construction to absorb the bundleEssences and squeeze the hash.

```
bundle = [transactionObject0, transactionObject1, transactionObject2, transactionObject3]

transactionObjectN = {address, value, obsoleteTag, timestamp, currentIndex, lastIndex}
```



BUNDLEHASH

- Convert the hash to trytes:
  bundleHash = convertToTrytes(hash)
  The bundleHash contains 243 / 3 = 81 trytes. For example:
  `bundleHash = NBBKCKPFECRKCDIBKSTHZYZKSXFEUPTIJRK9FECFKPTTSWTLUWGIFS9AHSDT9LASABRD9KDVFJ9GT9CKA`

- Add the bundleHash for each transactionObject in the bundle (fieldname is bundle).
  The bundleHash identifies which transactions belongs to the same bundle.
  See prepareTransfers demo (step 7):
  https://www.mobilefish.com/download/iota/preparetransfers_demo.txt

- After the bundleHash is added in the transactionObjects, the normalizedBundleHash is calculated. The normalizedBundleHash is explained in IOTA tutorial 16.
  The normalizedBundleHash is used to create or validate a signature.

BUNDLEHASH

- If the normalizedBundleHash has tryte value M:

  - Update the obsoleteTag in the **FIRST** transactionObject in the bundle by converting the obsoleteTag to trits, add value 1 and convert the result back to trytes.

  - After the obsoleteTag is updated:
    - Calculate the bundleEssences.
    - Apply the cryptographic sponge construction.
    - Calculate the bundleHash.
    - Calculate the normalizedBundleHash.

  - Repeat these steps until the normalizedBundleHash has no tryte value M.

BUNDLEHASH

- To check for tryte value M in the normalizedBundleHash, see:
  https://github.com/iotaledger/iota.lib.js/blob/v0.4.7/lib/crypto/bundle/bundle.js
  `Bundle.prototype.finalize = function()`

- See example where the obsoleteTag was updated multiple times.
  https://www.mobilefish.com/download/iota/data_transaction_example2.txt

BUNDLEHASH

- Remarks:
  - The normalizedBundleHash is not stored in the bundle but it is calculated using the bundleHash which is stored in the bundle.
  - The **obsoleteTag** plays an important role in the creation of the bundleHash.
    https://github.com/iotaledger/iota.lib.js/blob/v0.4.7/lib/crypto/bundle/bundle.js

```
Bundle.prototype.finalize = function()
```