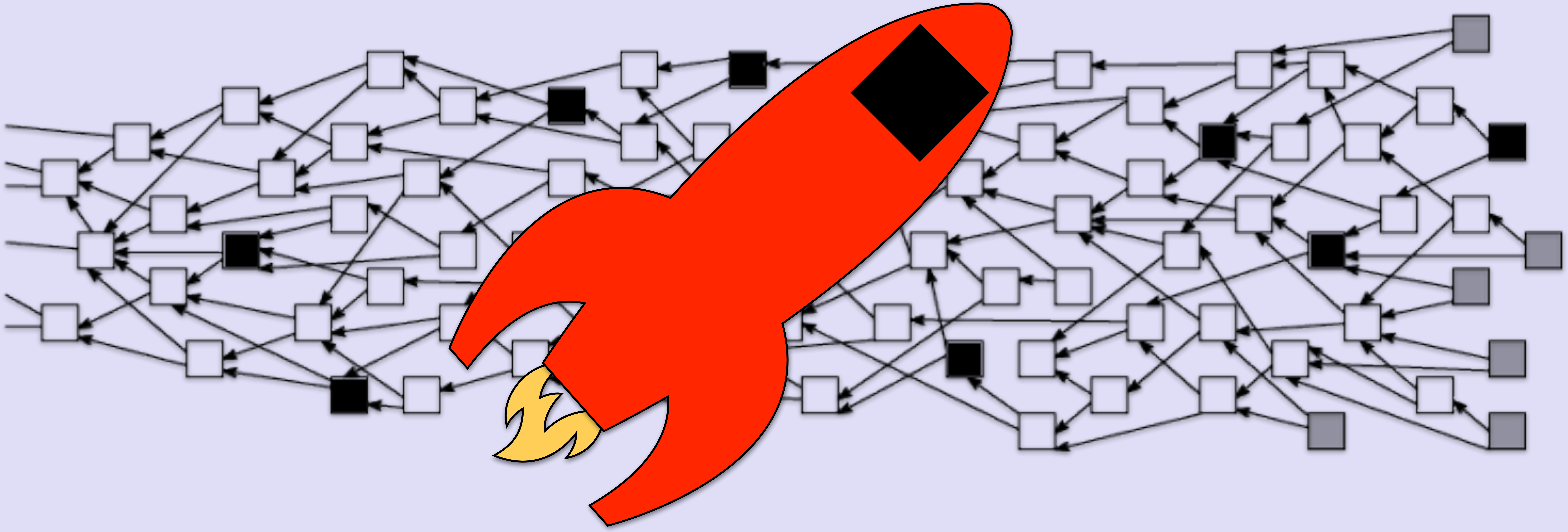
mobilefish.com

IOTA TUTORIAL 20

Masked Authenticated Messaging Payload



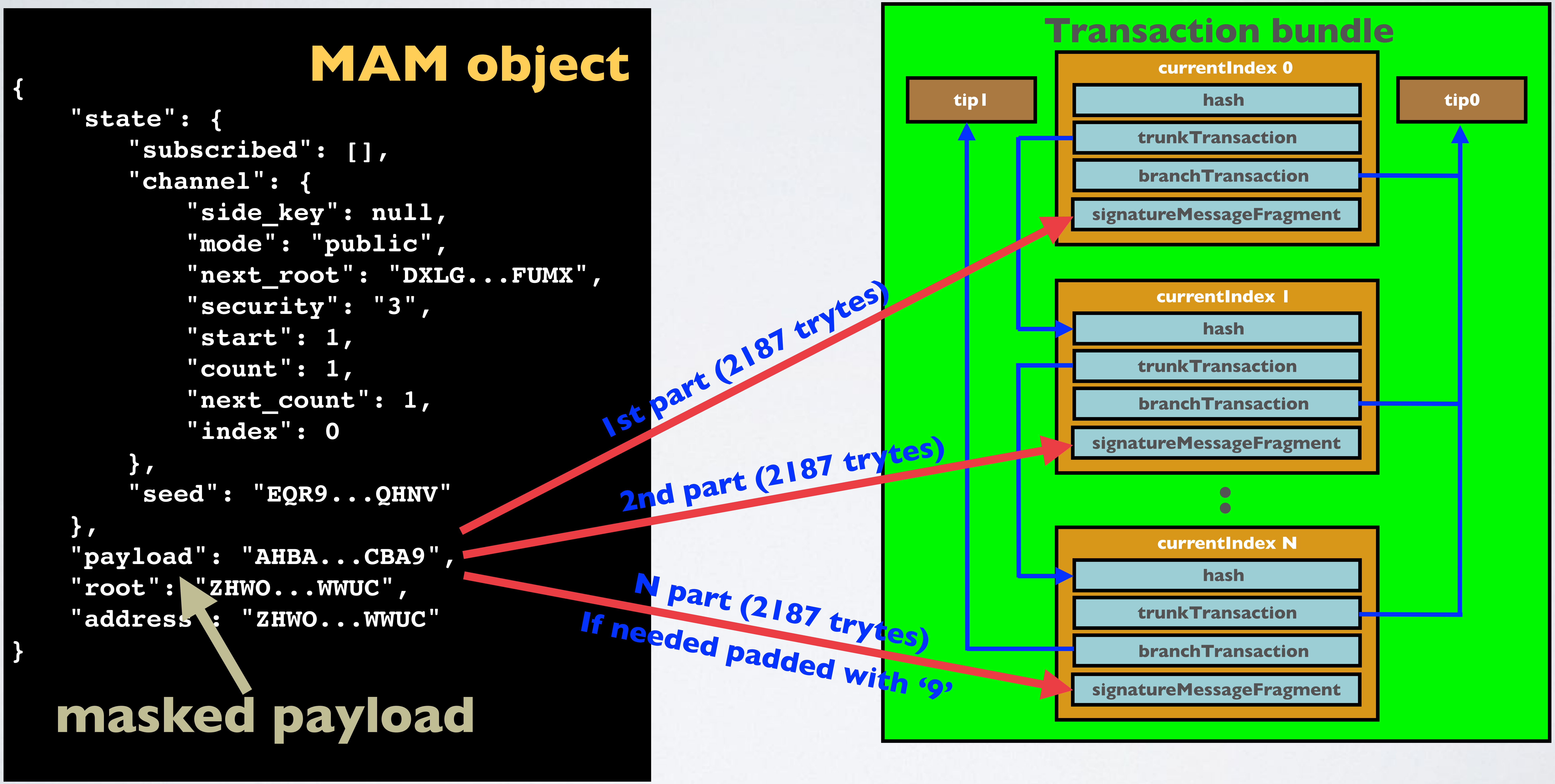
v1.0.0

INTRO

- In this video I will explain how the Masked Authenticated Messaging payload is created and also how it is parsed.

- If you have not watched IOTA tutorial 19, please watch that video first.

MAM OBJECT AND TRANSACTION BUNDLE

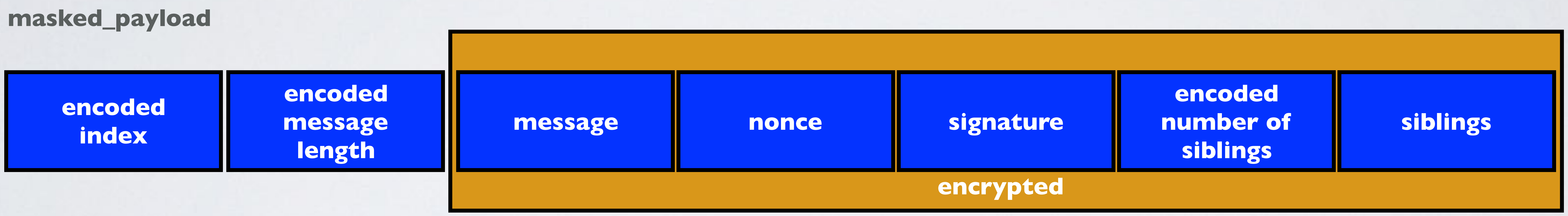


MAM object

```
{
    "state": {
        "subscribed": [],
        "channel": {
            "side_key": null,
            "mode": "public",
            "next_root": "DXLG...FUMX",
            "security": "3",
            "start": 1,
            "count": 1,
            "next_count": 1,
            "index": 0
        },
        "seed": "EQR9...QHNV"
    },
    "payload": "AHBA...CBA9",
    "root": "ZHWO...WWUC",
    "address": "ZHWO...WWUC"
}
```

MASKED PAYLOAD

- The masked payload contains the following information.



Note: encoded means convert integer value to trits

- message = {"payload":“ODGD..GAQD”, "next_root":“SJLO..RC9T”}

- The payload contains the actual sensor data converted to trytes.
  For example: {“data":40,"dateTime":"23/02/2018 10:54:34”}

MASKED PAYLOAD

- The masked payload is created in file:
  https://github.com/iotaledger/MAM/blob/master/mam/src/mam.rs
  ```
  pub fn create(seed: &[Trit], message: &[Trit], side_key:
  &[Trit], root: &[Trit], siblings: &[Trit], next:
  &[Trit], start: isize, index: usize, security: u8, ..)
  ```

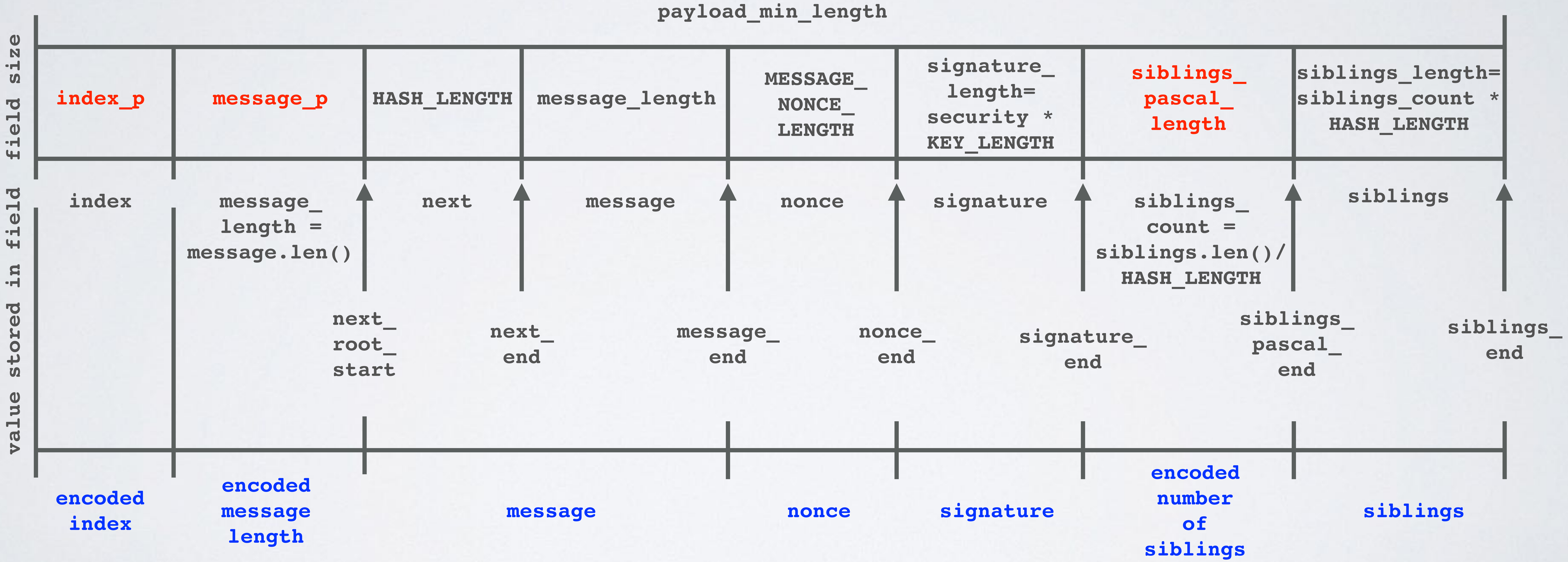
- In the next slides I will explain the basics how the masked payload is created and also how it is parsed by using drawings as a visual aid to help you to understand the concept.

MASKED PAYLOAD FORMAT

https://github.com/iotaledger/MAM/blob/master/mam/src/mam.rs

```
pub fn create(seed: &[Trit], message: &[Trit], side_key: &[Trit], root: &[Trit],
siblings: &[Trit], next: &[Trit], start: isize, index: usize, security: u8, ..)
```

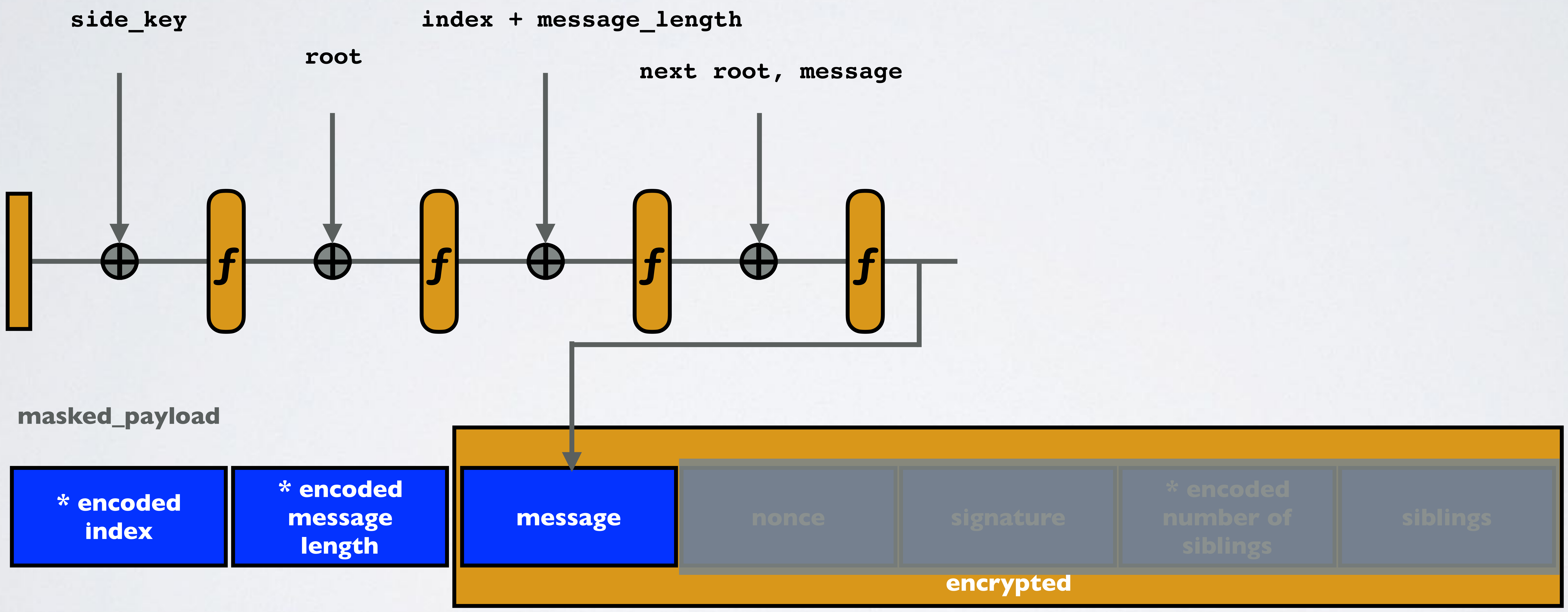
```
HASH_LENGTH = 243 trits
KEY_LENGTH = 27 x 243 = 6561 trits
MESSAGE_NONCE_LENGTH = 81 trits
```



MESSAGE MASKED

https://github.com/iotaledger/MAM/blob/master/mam/src/mam.rs

```
pub fn create(seed: &[Trit], message: &[Trit], side_key: &[Trit], root: &[Trit],
siblings: &[Trit], next: &[Trit], start: isize, index: usize, security: u8, ..)
```



*) Decimal value converted to trits

NONCE

- https://github.com/iotaledger/iota.rs/blob/master/curl-cpu/src/ham.rs
```
fn search<C: Curl<Trit>, CB: Curl<BCTrit>>(security: u8,
offset: usize, length: usize,..)
```

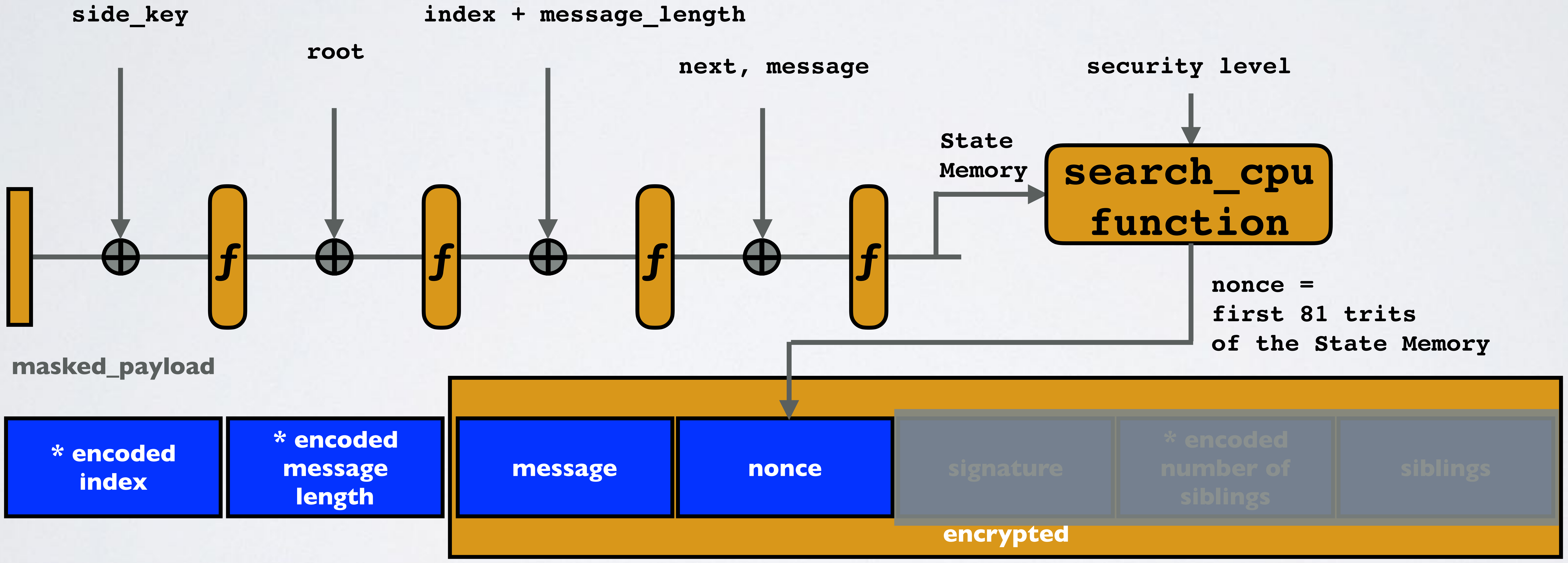
- Nonce is a number based on the side key, root, index, message length, next root and message. Lets call these values "masked bundle values".

- Nonce can be interpreted as a value created by scrambling these "masked bundle values" in a specific way using the security level.

NONCE MASKED

- https://github.com/iotaledger/iota.rs/blob/master/curl-cpu/src/ham.rs

```
fn search<C: Curl<Trit>, CB: Curl<BCTrit>>(security: u8,
offset: usize, length: usize,..)
```



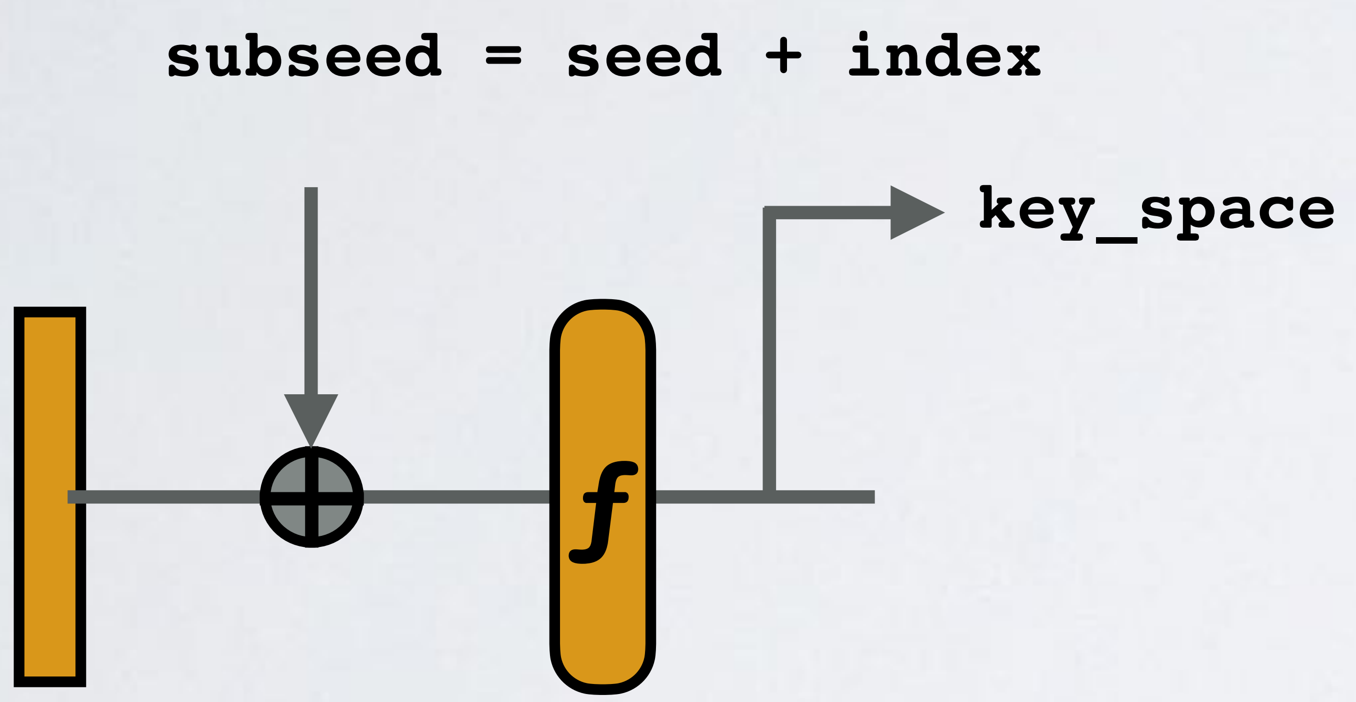
*) Decimal value converted to trits

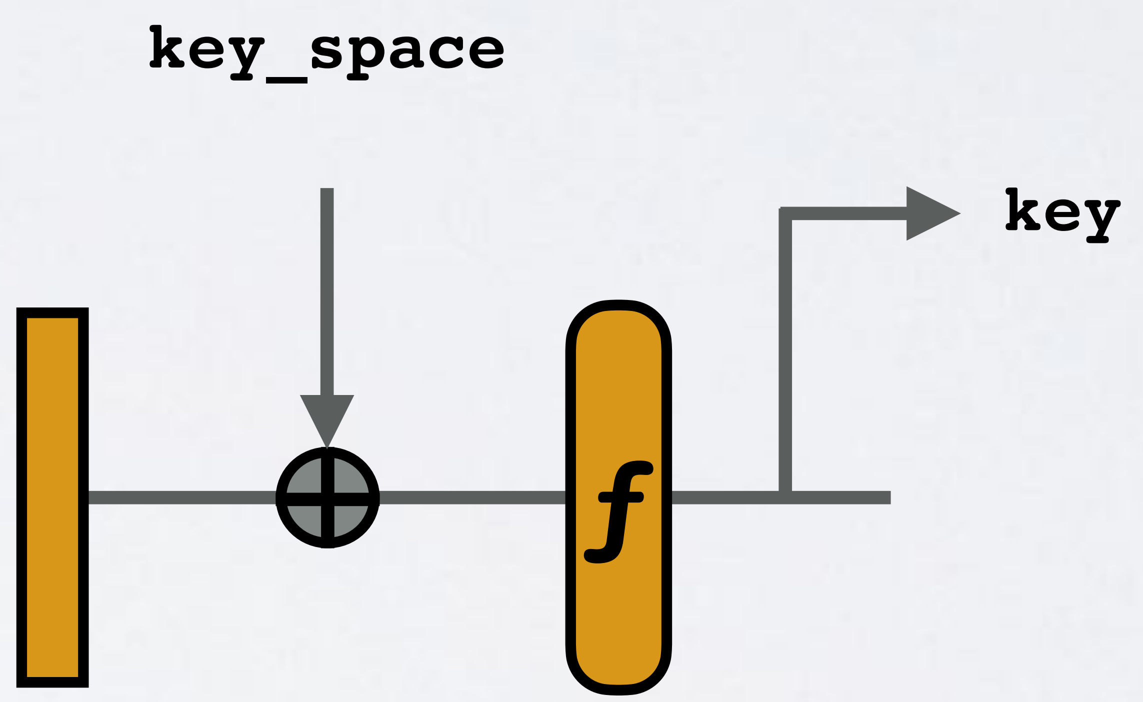
SUBSEED AND KEY

https://github.com/iotaledger/iota.rs/blob/master/sign/src/iss.rs

```
pub fn subseed<C>(seed: &[Trit], index: isize,..)
```

```
pub fn key<T, C>(key_space: &mut [T], security: usize, ..)
```





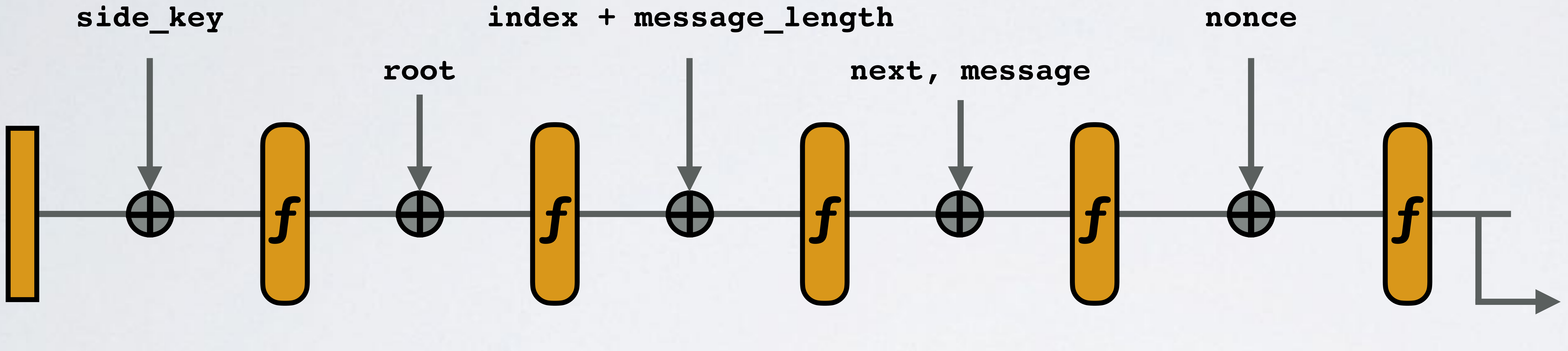
```
subseed size   = HASH_LENGTH
seed size      = HASH_LENGTH
key_space size = HASH_LENGTH
key size       = security x KEY_LENGTH

HASH_LENGTH    = 243 trits
KEY_LENGTH     = 27 x 243 = 6561 trits
```

SIGNATURE MASKED

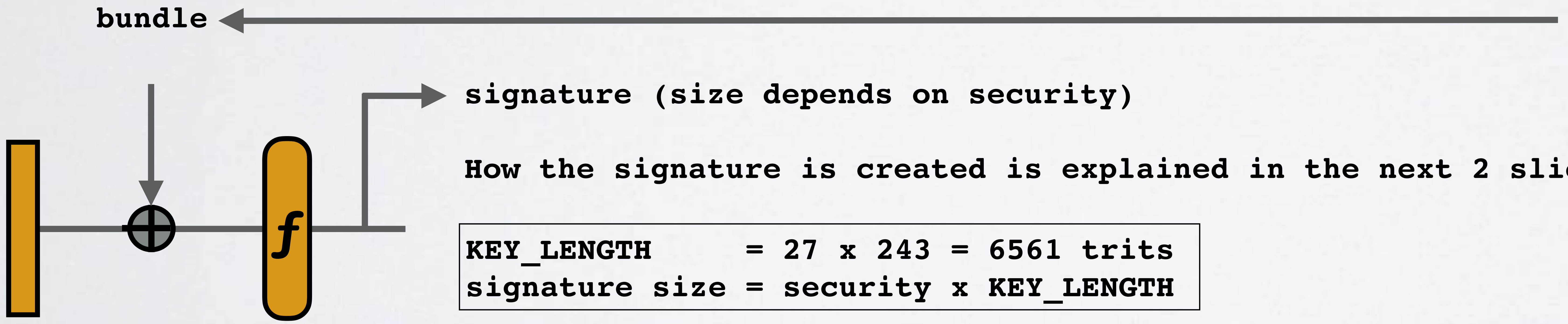
https://github.com/iotaledger/iota.rs/blob/master/sign/src/iss.rs

```
pub fn create(seed: &[Trit], message: &[Trit], side_key: &[Trit], root: &[Trit],
siblings: &[Trit], next: &[Trit], start: isize, index: usize, security: u8, ..)
```



bundle = First 243 trits of the State Memory.

```
pub fn signature<C>(bundle: &[Trit], key_signature: &mut [Trit],..)
```



bundle is also known as bundleHash

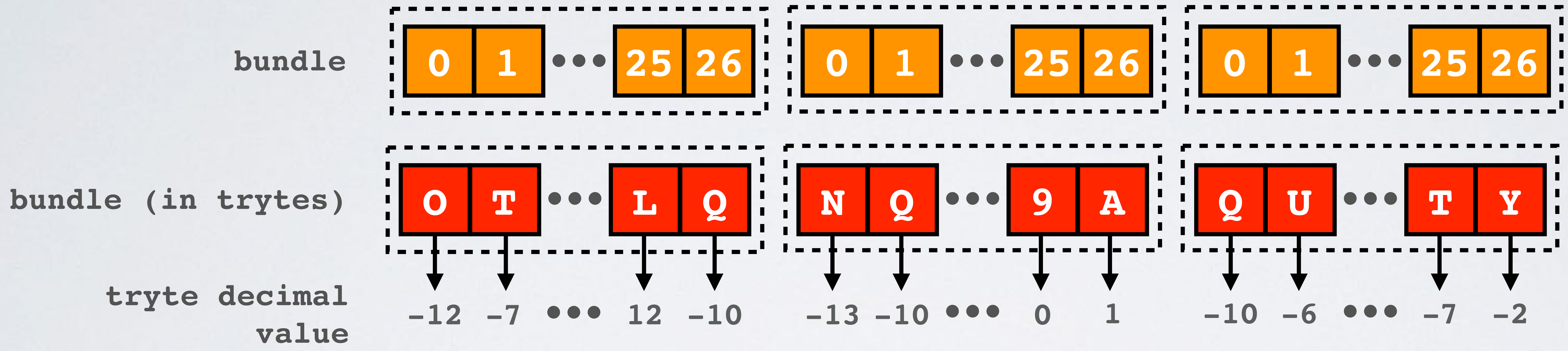
signature (size depends on security)

How the signature is created is explained in the next 2 slides.

```
KEY_LENGTH     = 27 x 243 = 6561 trits
signature size = security x KEY_LENGTH
```

CALCULATE NUMBER OF HASHES

bundle size = 243 trits = 3 x 27 trytes



https://github.com/iotaledger/iota.rs/blob/master/sign/src/iss.rs

```
pub fn signature<C>(bundle: &[Trit], key_signature: &mut [Trit],..)
```

K = 13 - decimal: 25 20 ••• 1 23 | 26 23 ••• 13 12 | 23 19 ••• 20 15

K times to hash each segment to CREATE signature

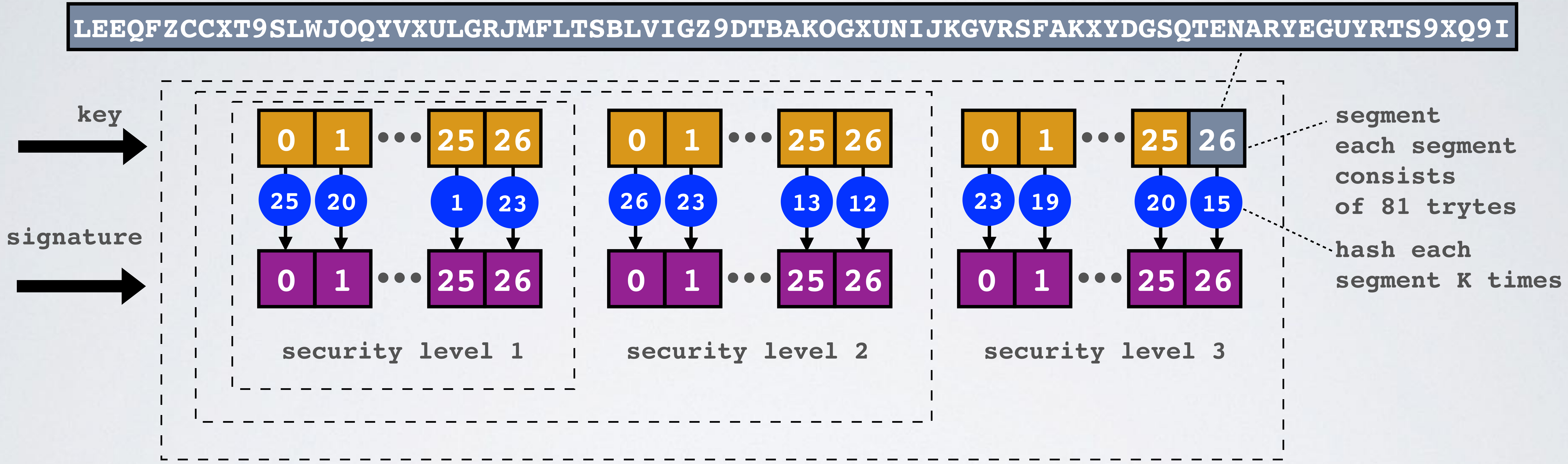
https://github.com/iotaledger/iota.rs/blob/master/sign/src/iss.rs

```
pub fn digest_bundle_signature<C>(bundle: &[Trit], signature: &mut [Trit],..)
```

K = 13 + decimal: 1 6 ••• 25 3 | 0 3 ••• 13 14 | 3 7 ••• 6 11

K times to hash each segment to VALIDATE signature

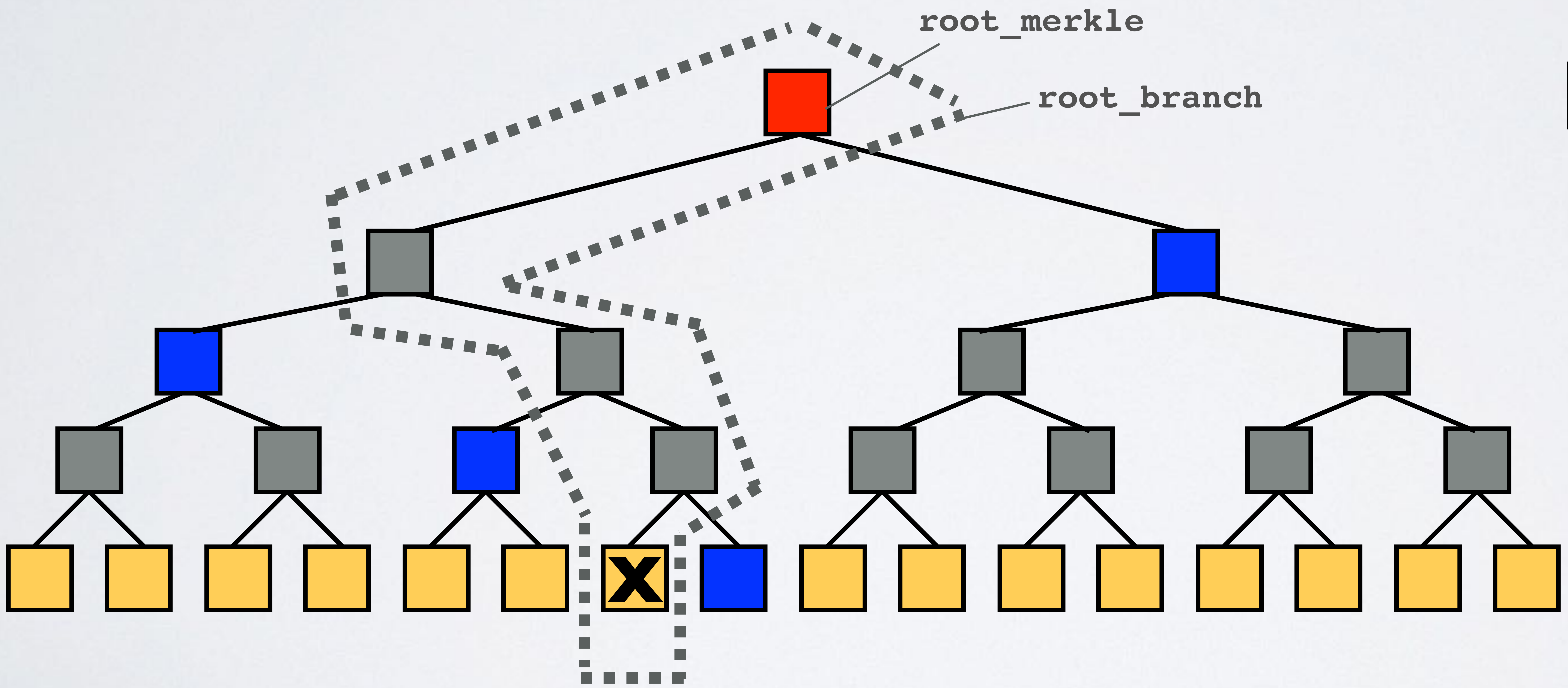
CALCULATE SIGNATURE


LEEQFZCCXT9SLWJOQYVXULGRJMFLTSBLVIGZ9DTBAKOGXUNIJKGVRSFAKXYDGSQTENARYEGUYRTS9XQ9I
key
signature
0
1
25
26
25
20
1
23
0
1
25
26
security level 1
0
1
25
26
26
23
13
12
0
1
25
26
security level 2
0
1
25
26
23
19
20
15
0
1
25
26
security level 3
segment
each segment consists of 81 trytes
hash each segment K times

```
KEY_LENGTH       = 27 x 243 = 6561 trits
key size         = security x KEY_LENGTH
signature size = security x KEY_LENGTH
```

CALCULATE NUMBER OF SIBLINGS AND SIBLINGS

- Setup Merke tree:
  https://github.com/iotaledger/mam.client.js/blob/master/lib/mam.web.js
  `function createMessage(SEED, MESSAGE, SIDE_KEY, CHANNEL)`

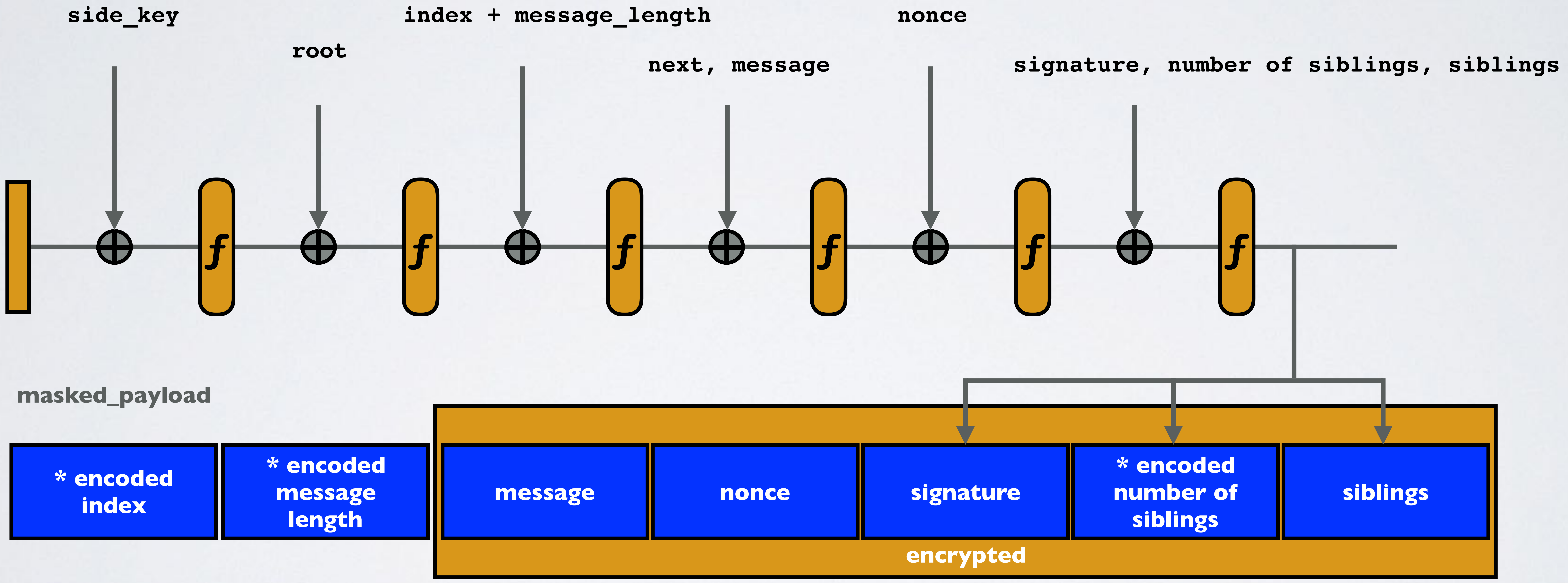


🟦 `root_siblings belonging to the root_branch`

SIGNATURE, NUMBER OF SIBLINGS, SIBLINGS MASKED

https://github.com/iotaledger/MAM/blob/master/mam/src/mam.rs

```
pub fn create(seed: &[Trit], message: &[Trit], side_key: &[Trit], root: &[Trit],
siblings: &[Trit], next: &[Trit], start: isize, index: usize, security: u8, ..)
```



*) Decimal value converted to trits

PARSE MASKED PAYLOAD

- When consuming a MAM stream, the signature is first validated by verifying if the signature belongs to one of the tree's leaves.
  If the signature verification fails, the entire message is deemed invalid.
  If the signature verification is valid, the message is unmasked.

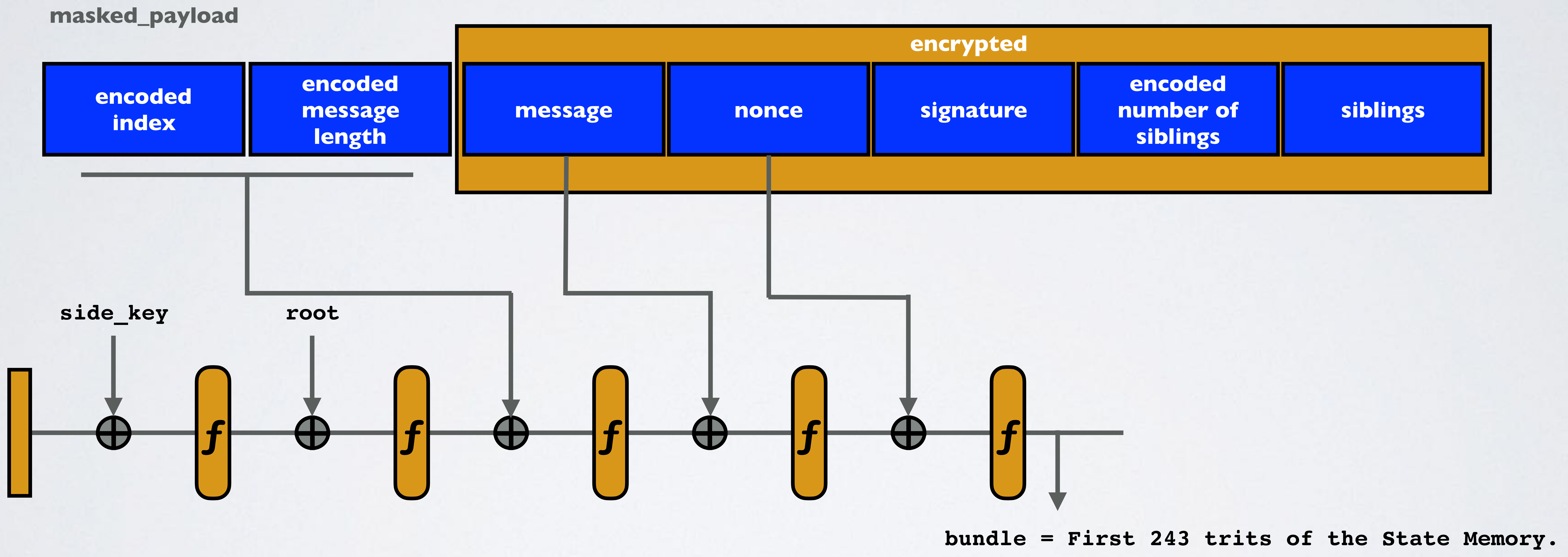
- The masked payload is parsed in file:
  https://github.com/iotaledger/MAM/blob/master/mam/src/mam.rs

```
pub fn parse<C>(payload: &mut [Trit], side_key: &[Trit],
root: &[Trit], ..)
```

EXTRACT BUNDLE

https://github.com/iotaledger/MAM/blob/master/mam/src/mam.rs

```
pub fn parse<C>(payload: &mut [Trit], side_key: &[Trit], root: &[Trit], ..)
```

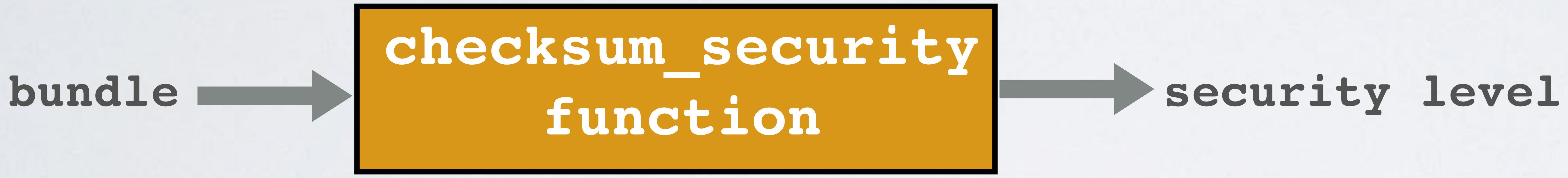


```
bundle = First 243 trits of the State Memory.
```

EXTRACT SECURITY LEVEL

- https://github.com/iotaledger/iota.rs/blob/master/sign/src/iss.rs

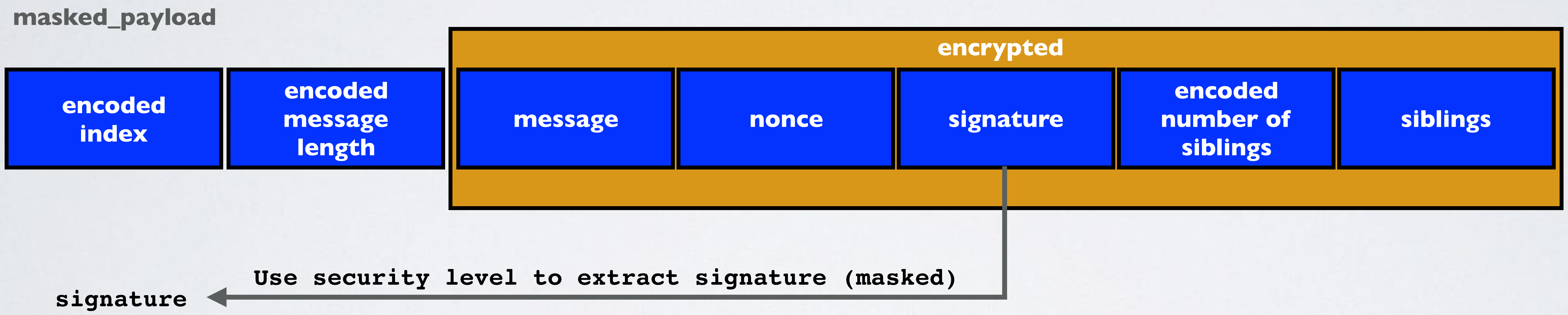
```
pub fn checksum_security(hash: &[Trit])
```

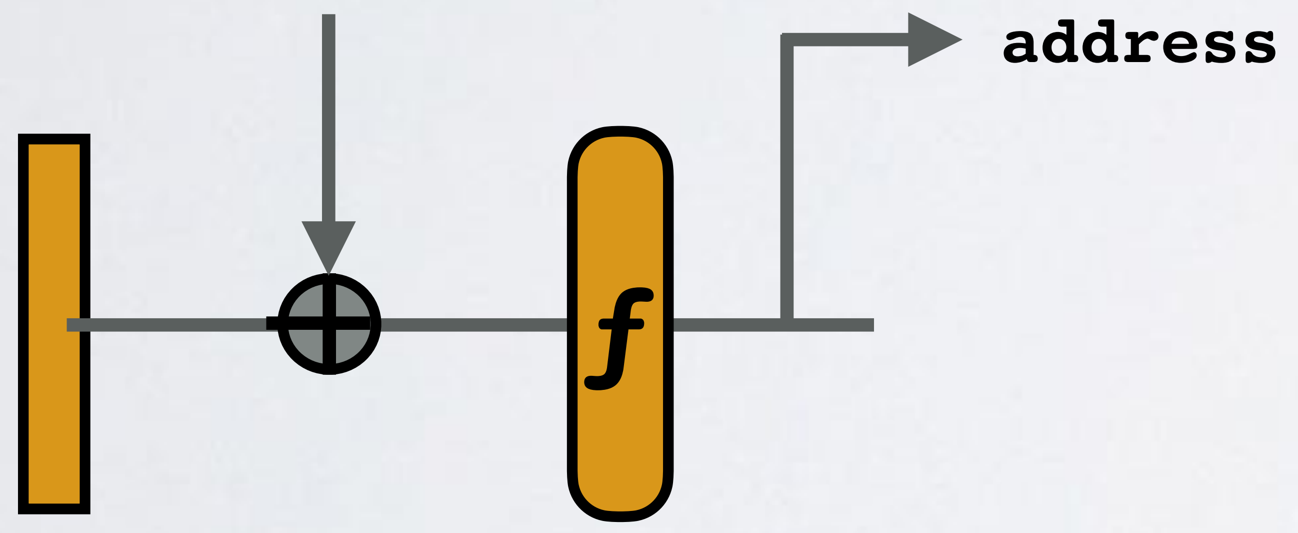


EXTRACT SIGNATURE AND ADDRESS

https://github.com/iotaledger/iota.rs/blob/master/sign/src/iss.rs

```
pub fn digest_bundle_signature<C>(bundle: &[Trit], signature: &mut [Trit],..)
```

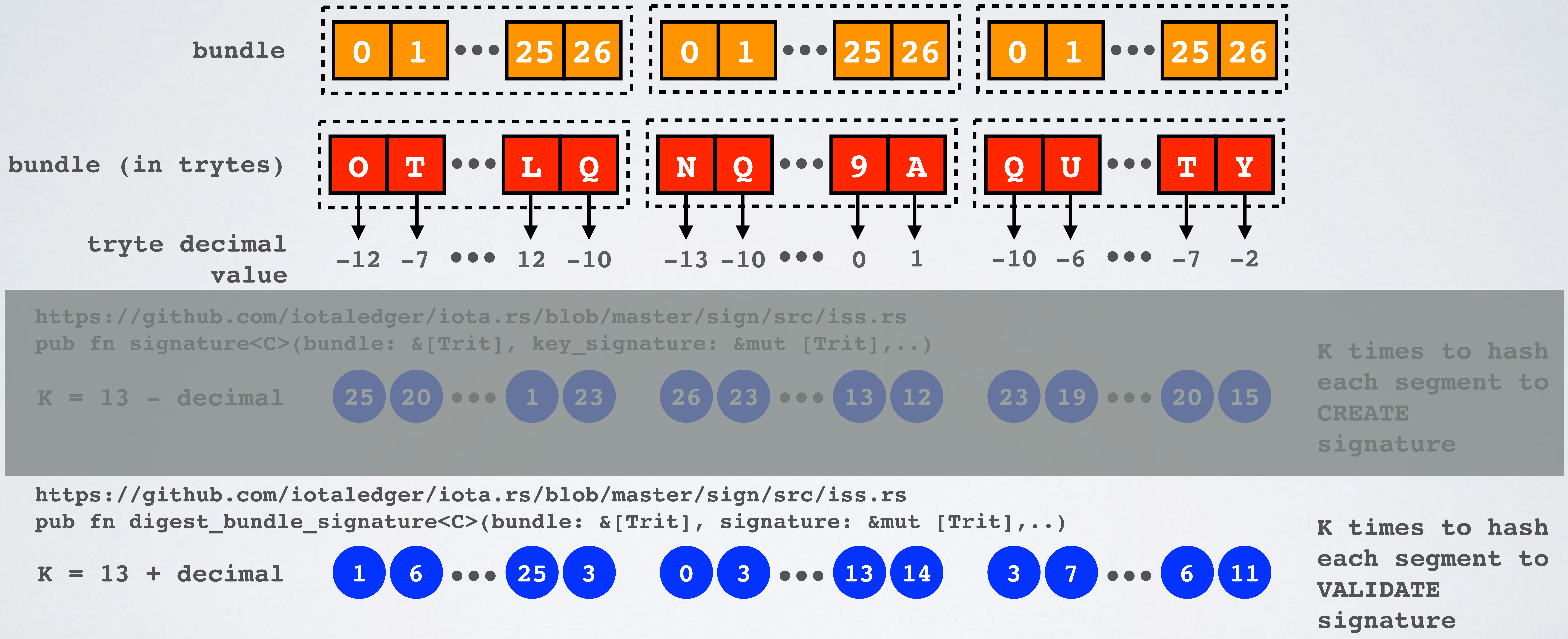




```
pub fn digest_bundle_signature<C>(bundle: &[Trit], signature: &mut [Trit],..) {
  :
  iss::digest_bundle_signature(&hmac, &mut payload[pos..sig_end], curl);
  hmac.clone_from_slice(&curl.rate());        (hmac contains the digest)
  :
  curl.absorb(&hmac);
  hmac.clone_from_slice(curl.rate());         (hmac contains the address)
  :
}
```

CALCULATE NUMBER OF HASHES

bundle size = 243 trits = 3 x 27 trytes

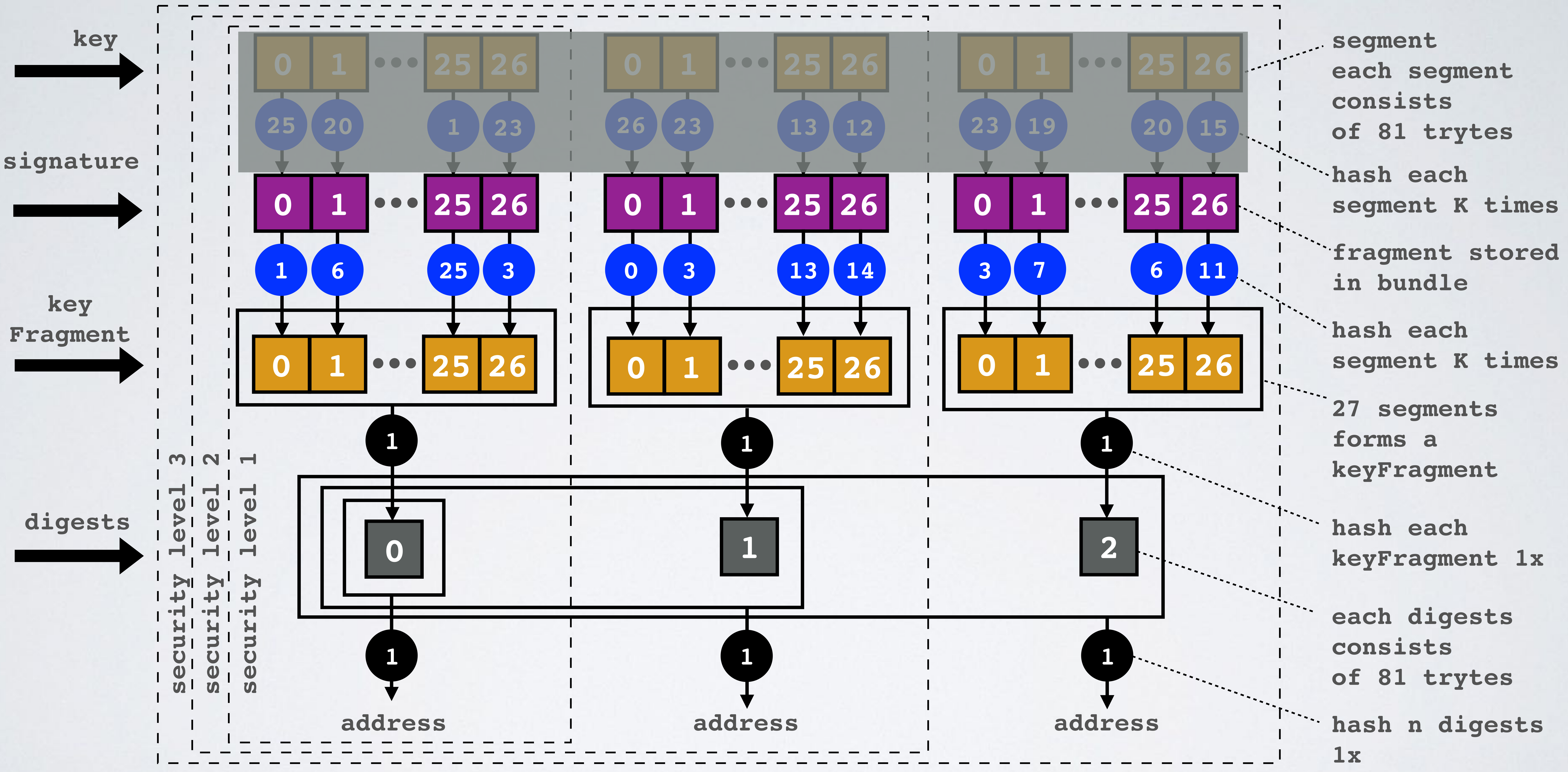


```
https://github.com/iotaledger/iota.rs/blob/master/sign/src/iss.rs
pub fn signature<C>(bundle: &[Trit], key_signature: &mut [Trit],..)
```

K times to hash each segment to CREATE signature

```
https://github.com/iotaledger/iota.rs/blob/master/sign/src/iss.rs
pub fn digest_bundle_signature<C>(bundle: &[Trit], signature: &mut [Trit],..)
```

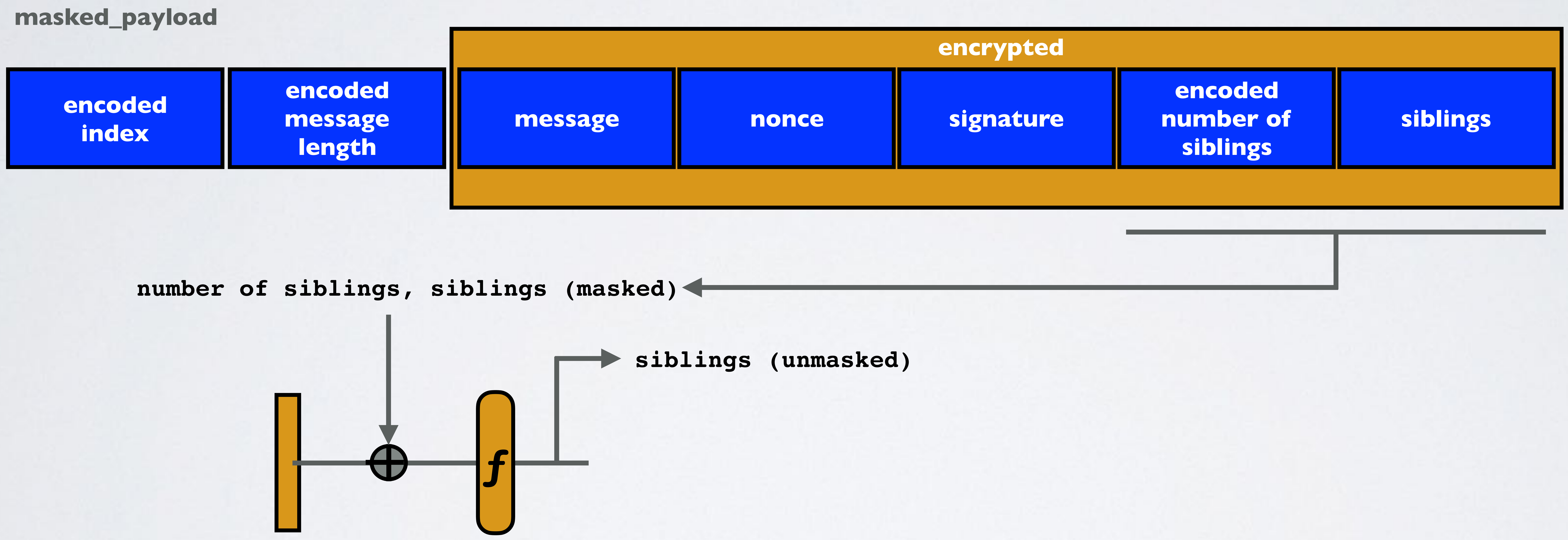
K times to hash each segment to VALIDATE signature


key
signature
key Fragment
digests
security level 3
security level 2
security level 1
0
1
25
26
25
20
1
23
26
23
13
12
23
19
20
15
1
6
25
3
0
3
13
14
3
7
6
11
2
address
address
address
segment
each segment consists of 81 trytes
hash each segment K times
fragment stored in bundle
hash each segment K times
27 segments forms a keyFragment
hash each keyFragment 1x
each digests consists of 81 trytes
hash n digests 1x

SIBLINGS UNMASKED

https://github.com/iotaledger/MAM/blob/master/mam/src/mam.rs

```
pub fn parse<C>(payload: &mut [Trit], side_key: &[Trit], root: &[Trit], ..)
```



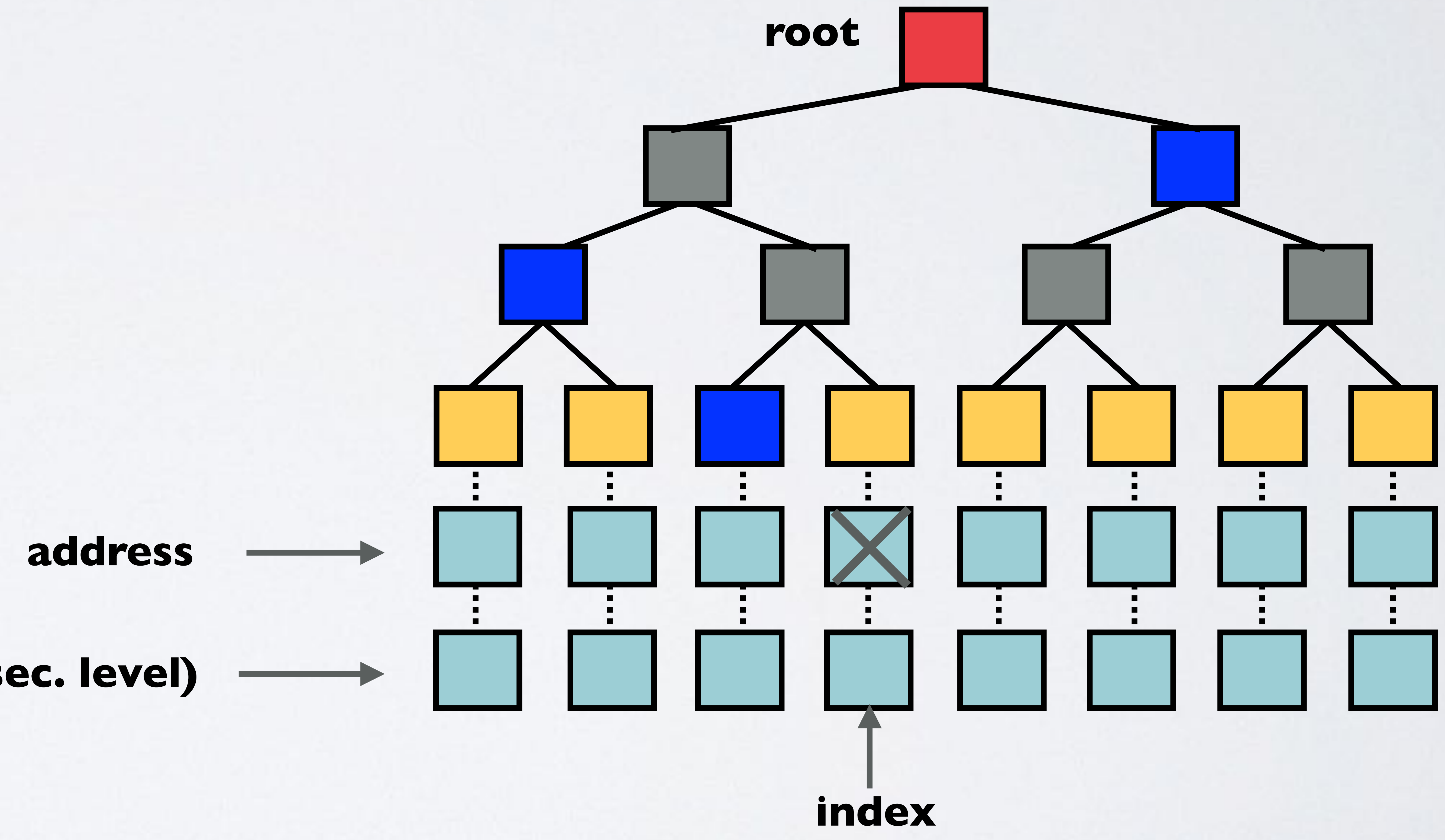
CALCULATE ROOT

https://github.com/iotaledger/iota.rs/blob/master/merkle/src/simple.rs

```
pub fn root<C: Curl<Trit>>(address: &[Trit], hashes: &[Trit], index: usize, ..)
```

The parse function requires a root value when the function is called. This root value must match the calculated root.

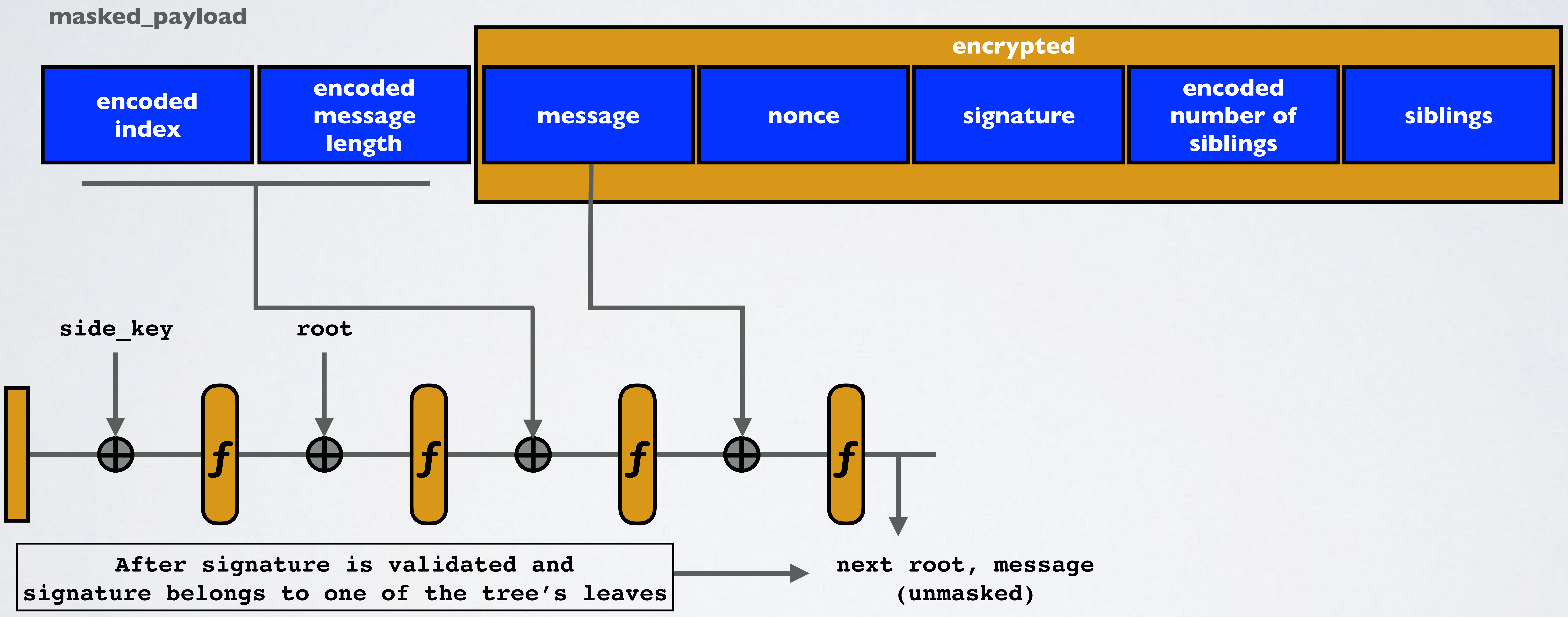
If this is the case, the next root and message are unmasked.



NEXT ROOT AND MESSAGE UNMASKED

https://github.com/iotaledger/MAM/blob/master/mam/src/mam.rs

```
pub fn parse<C>(payload: &mut [Trit], side_key: &[Trit], root: &[Trit], ..)
```


masked_payload
encrypted
encoded index
encoded message length
message
nonce
signature
encoded number of siblings
siblings
side_key
root
f
After signature is validated and signature belongs to one of the tree's leaves
next root, message (unmasked)