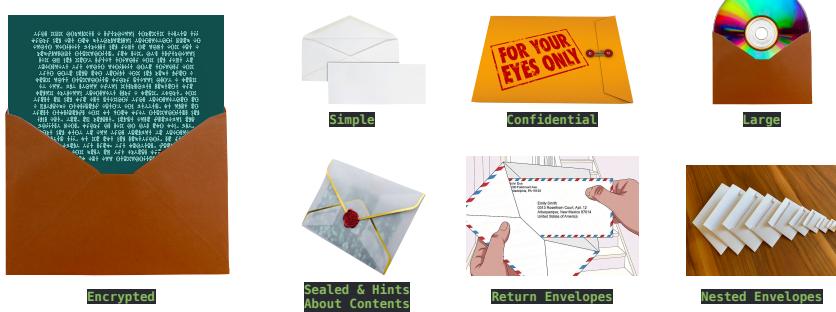




YOU CAN DO A LOT WITH ENVELOPES!

Different Features at Each Level



Envelopes All the Way Down

```
struct Envelope {
    let subject: Envelope
    let assertions: [Assertion]
}

struct Assertion {
    let predicate
    let object: Envelope
}
```



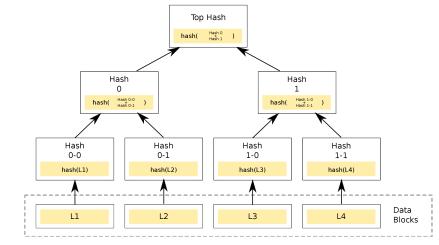
ELISION

Act or instance of omitting something;
or to suppress, ignore or pass over

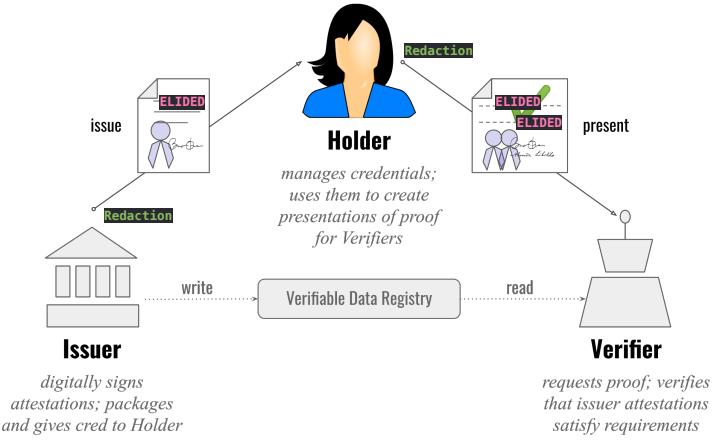
In an Envelope, elided elements are replaced by their Merkle-Tree digest, therefore allowing the same digests to be calculated for the entire tree despite those absences.

We can also increase the noncorrelation of hash values¹ by leveraging random salts.

1: in particular for well-known predicates



Elision by either Issuer or Holder or both



Elision¹ possible for all parts of Semantic Triple

```
; Nothing elided
"Alice" [
    "knows": "Bob"
]

; SUBJECT elided
ELIDED [
    "knows": "Bob"
]

; PREDICATE elided
"Alice" [
    ELIDED: "Bob"
]

; OBJECT elided
"Alice" [
    "knows": ELIDED
]

; ASSERTION elided
ELIDED

; ENVELOPE elided
ELIDED
```

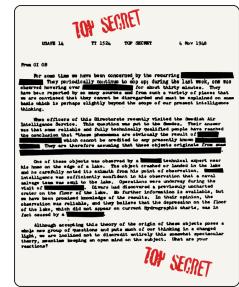
1: Anywhere you can elide, you could also encrypt without changing the merkle tree

REDACTION

Elision for purpose of withholding

Goal is to omit information that is not needed for the business purposes of the parties that the 'envelope' is being shared with.

The **Redacted** digests do not change, so all signatures still verify.

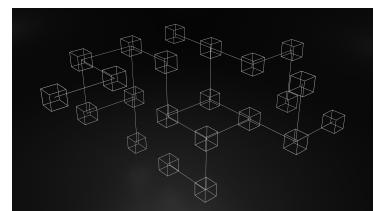


REFERENCE or ENCRYPTION

Elision that affords future retrieval
reference to external objects or encrypted internal objects

Goals to allow for omission for purposes of limiting size (such as images), additional authentication (permissions to read or logging access), or remote lookup.

Reference can also serve Proof-of-Existence and Proof-of-Publication. **Encryption** can serve Proof-of-Escrow.



OR CONTACT:
ChristopherA@LifeWithAlacrity.com
Blockchain Commons