FLIC Manifests
Goals

• Copy UNIX inode concept
  • Index tables and memory pointers

• Adaption to ICN
  • Hash values instead of block numbers

• Advantages
  • Single root manifest signature that covers all children nodes
  • No need for explicit chunking
  • Supports block-level de-duplication
Limitations

- All data leaves must be present at creation time
FLIC Overview

root manifest

optional name:
   /icn/name/of/this/flic

HashGroup:
   optional metadata:
      block size, total tree size, overall digest, locator, etc.
      hash-valued data pointer -----------> | data |
      ...                                   | sub manifest
      hash-valued manifest pointer ----------->  

optional additional HashGroups ..

optional signature
FLIC Grammar

ManifestMsg := Name? HashGroup+
HashGroup := MetaData? (DataPointer | ManifestPtr)+
DataPointer := HashValue
ManifestPtr := HashValue
HashValue := OCTET[32]

MetaData := Property*
Property := Locator | DataBlockSize | OverallDataSize | OverallDataSHA256Digest | ...
FLIC Trees

- Skewed (similar to a list)
  \[ \text{DDD} \rightarrow \text{DDD} \rightarrow \ldots \rightarrow \text{DDD} \rightarrow \text{DDD} \]

- Balanced

  root manifest = \[ \text{DDD} \]

  \[ \text{DDD} \rightarrow \text{DDD} \rightarrow \text{DDD} \rightarrow \text{DDD} \rightarrow \text{DDD} \]

  \[ \text{DDD} \rightarrow \text{DDD} \rightarrow \text{DDD} \rightarrow \text{DDD} \rightarrow \text{DDD} \]
Use Cases

- Block-level reduplication
- Growing ICN collections
- Republishing under a new name
- Data chunks of variable size
Deduplication

old_mfst -> h1 --> oldData1 <-- h1 < - - - new_mfst
\ - > h2 --> oldData2 <-- h2 < - - - /
\ | replace3 <-- h5 < - - - /
\- > h3 --> oldData3        /
\ > h4 --> oldData4 <-- h4 < - - /
Growing Collections

old data < - - - - mfst_old <-- h_old - - mfst_new
/ 
new data1 <-- h_1 - - - - - - - - - - - /
new data2
...
new dataN <-- h_N - - - - - - - - - - - /
Republishing

- Create nameless manifest trees.
- Use “named manifest” that points to the nameless root.

```r
data <- nameless_mfst()
```
```r
h <- mfst(/com/parc/east/the/flic)
```
```r
<- mfst(/com/parc/west/old/the/flic)
```
```r
<- mfst(/internet/archive/flic234)
```
Variable Size Blocks

M = (manifest
    (hashgroup((metadata(blocksize=12)) (dataptr=h1))
    (hashgroup((metadata(blocksize=1)) (dataptr=h2))
    ...
    )
)
Some Questions

• Should each pointer have an associated size?

• What are the mechanics for handling manifests with multiple hash groups?

• How should encryption be handled?