Multiple Signature

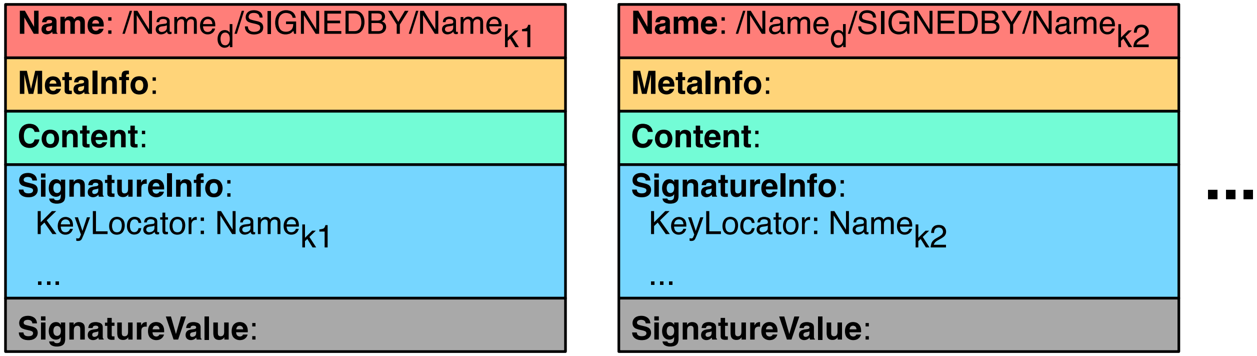
Why Multiple Signature?

- Certificates: the same <name, key> pair may be certified by different parties
    - /alice’s key could be asserted by both /bob and / cathy

- Signature agility: different signing algorithms & key size
    - a RSA signature, a ECDSA signature
    - a signature generated with 2048-bit RSA key, a signature generated with 4096-bit RSA key

Design Options

- Option 1: different signature, different data packet
  - naming convention to distinguish packets with different signature
    - append signing key name to data name: /<data_name>/SIGNEDBY/<key_name>



Name: /Name$_d$/SIGNEDBY/Name$_{k1}$
MetaInfo:
Content:
SignatureInfo:
KeyLocator: Name$_{k1}$
...
SignatureValue:

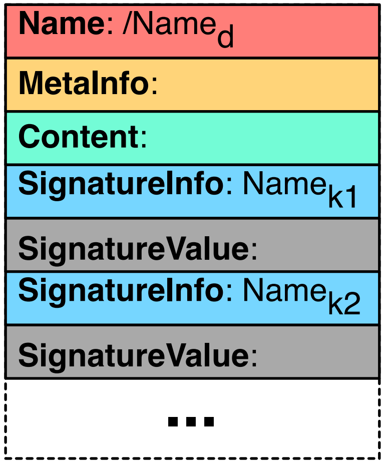
Name: /Name$_d$/SIGNEDBY/Name$_{k2}$
MetaInfo:
Content:
SignatureInfo:
KeyLocator: Name$_{k2}$
...
SignatureValue:

...

- Pros
  - no need to extend packet format
- Cons
  - complexity in collecting signatures
    - discover signatures, one interest/data exchange for each signature

Design Options

- Option 2: extend data packet format to carry multiple signatures
  - [Name, Metainfo, Content, SigInfo1, SigVal1, SigInfo2, SigVal2, …]



Name: /Name$_d$

MetaInfo:

Content:

SignatureInfo: Name$_{k1}$

SignatureValue:

SignatureInfo: Name$_{k2}$

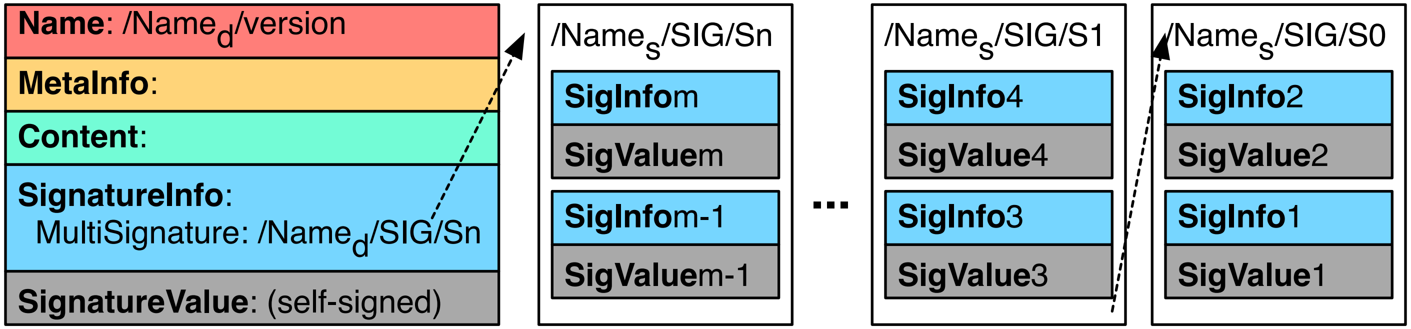
SignatureValue:

...

- Pros:
  - single data retrieval can bring back multiple signatures
- Cons:
  - packet size limit: cannot carry arbitrary number of signatures
  - increase complexity of packet parsing

Design Options

- Option 3: signature bundles
  - group signatures into a separate data packet **signature bundle**
  - put the sig bundle name in SigInfo
  - when bundle involves more than one packet, chain them together



- Pros
  - no need to extend packet format
  - easy to retrieve all the signatures
- Cons
  - key owner is responsible of collecting signatures and making bundle

MultiSignature Extension

- Signature bundle
  - naming convention
    - / <data_name> /SIG/[seqNo]
    - /alice/ksk-123/KEY/SIG/2
    - once a user learns the data name, can pre-fetch bundle packets
  - bundle chain
    - bundle packets are sequentially chained
    - full name of n-th bundle packet is put into (n+1)-th bundle packet.
    - with the n-th bundle packet, one can retrieve all the previous n-1 bundle packets

- MultiSignature Extension
  - a list of seqNo of signature bundles and their implicit digest
  - [[7, 4f3a9d…], …, [1, 75df2a…], [0, b34a34]]
  - a user can construct the full name of each bundle packet in the extension
  - when number of bundle packets exceeds the limit *m*, carry the most recent m seqNos and their digests only
    - the rest can be retrieved through hash chain