# Name-Data Integrity

- In ICN:
  - Old security model with secure channel to trusted servers is awkward with promiscuous caching
  - Would like a naming scheme that can support security based on name-data integrity so we don't need to care which copy we get back
- In IETF:
  - A number of WGs needs to name data objects instead of hosts, e.g.: core, decade,..
  - Advantages with a common naming scheme for data objects within the IETF includes
  - Application independent naming of data objects makes it easier to create hybrid' applications
- In Both:
  - Name-data integrity implies some use of crypto related to names

# Cryptography in ICN Names

- Name-data integrity: Given a name and data object allows you to check that the latter matches the former
  - (To me) this is a core ICN service, since it means you don't care from whom or where you got the data object
- If we can use the same names for objects and preserve name-data integrity regardless of ICN protocol, then that seems like a possibly major benefit (esp. if ICN gets deployed in reality)
- “ni” URI scheme, is aimed at that:
  - draft-farrell-decade-ni, resolving IESG Evaluation comments
  - e.g. “ni:///sha-256-32;20W-LA?ct=text/plain”
  - Other formats defined too, e.g. binary, .well-know/ni URL

# Crappy or Happy Crypto?

- Hashes are handy but nasty as names
- Do symmetric schemes suck?
- Signature schemes have rubbish revocation
- Canonicalization (c14n) is continually crappy
  - Probably implies some form of “alias” is needed in all ICN schemes that support name-data integrity
- (Possible) conclusion:
  - Try find common ground for naming at least static objects and do experiments to figure out how to handle other things later
    - Do not try to boil ocean and provide name-data integrity for everything at once
  - I suggest “ni” URIs (but then I would, wouldn't I:-)

# Hash-based Names

- ni URIs are an example
- Not human-friendly, nor aggregatable
  - But search is needed for human friendly (and not just text based search)
  - Aggregates will scatter over caches anyway unless we want to benefit large providers
- Major plus: no keys => no key management

# Any role for Symmetric Schemes?

- Seems a bit DRM-like
- Scaling symmetric key management can be hard and lead to bottlenecks
- But maybe there's some scope here...

# Signature Schemes

- Name includes hash of public key
  - Usual problems with hashes
- Name free-form but data has wrapper with signature over name and content (and other stuff)
  - Wrapping content some new way is a pain
- Don't get name-data integrity quite as before
  - Name-public-key-signature-data integrity
  - Anyone with that private key can fool the holder of the name
- Private keys do leak out
- Revocation scheme problem
- Basically the same as any other PKI

# Flakier Crypto...

- IBE, Group Signatures, …
- Suggestion: Ignore these

# A not-so-modest Suggestion

- Try find common ground for naming at least static objects
  - Would allow better protocol comparisons e.g. against a “standard” message corpus
  - Doesn't mean all schemes need to use same names but that mappings should be defined
- Figure out how to handle naming other things later
  - Dynamic objects, etc. etc.
- I suggest URIs ad specifically “ni” URIs
  - (but then I would, wouldn't I:-)
  - Model there would be to define mappings between ni URIs and other name forms as used by other schemes
  - Mappings could be ICNRG documents