Schematizing Trust in Named Data Networking

Yingdi Yu[1], Alex Afanasyev[1], David Clark[2],
kc claffy[3], Van Jacobson[1], Lixia Zhang[1]

1. UCLA 2. MIT 3. CAIDA

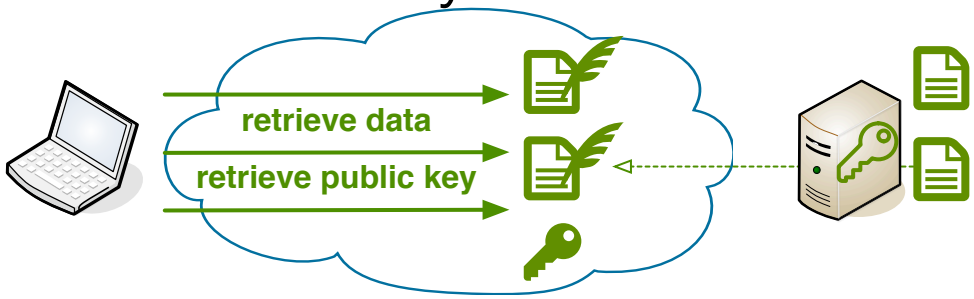
NAMED DATA NETWORKING

Motivation

- Usability is critical to security solutions
- Tool to explicitly express trust model
- Mechanism to automate trust management

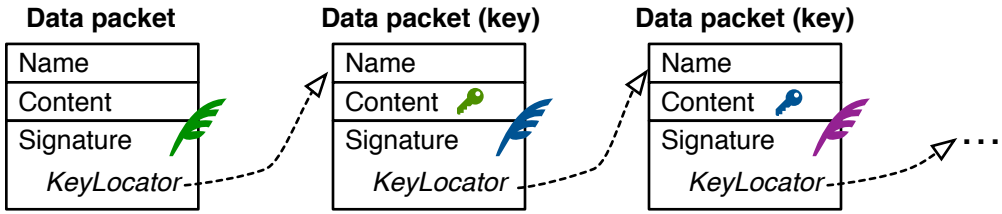
Data Authentication in NDN

- Data-centric authenticity



  - mandate signature on every data packet

- Data authentication needs public key only



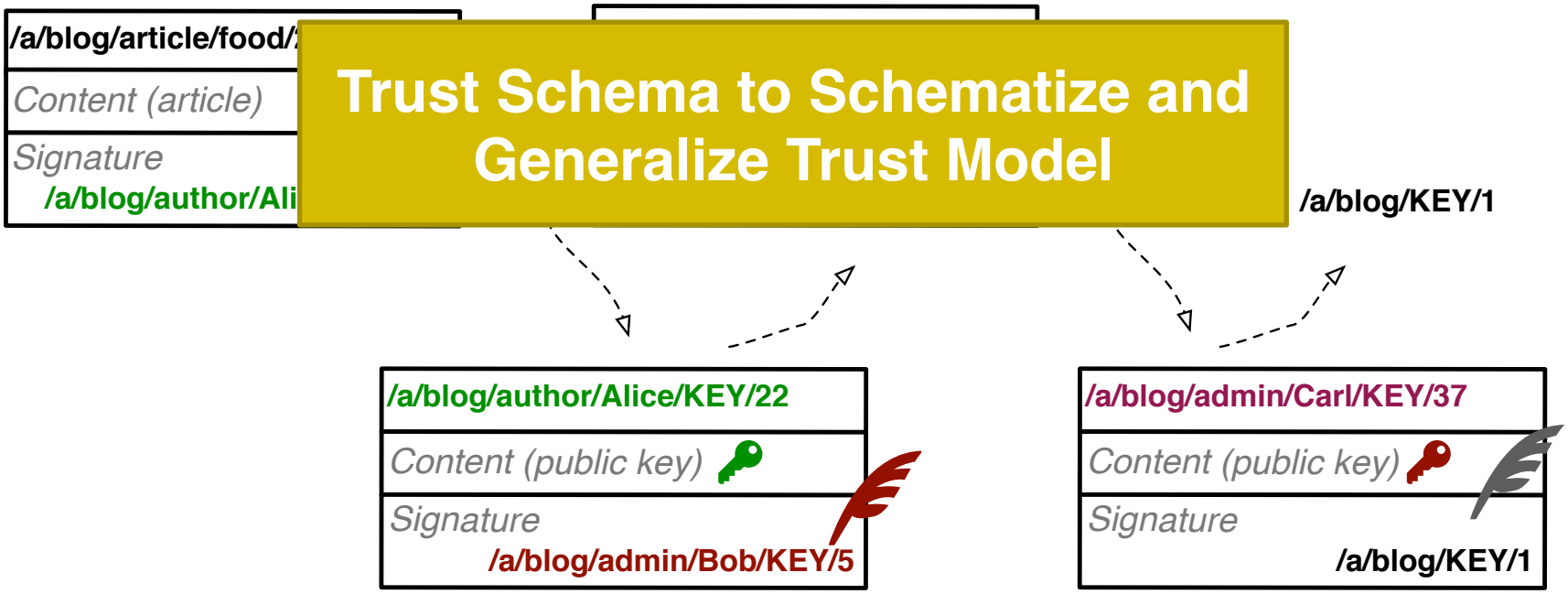
  - independent from where/how data packet is retrieved
  - privilege of online signing key can be restricted

Trust Model

- Data signing and verification require a trust model
  - one or more pre-trusted keys
  - which key is authorized to sign/verify which data
    - key is just another type of data
  - defines strict authentication path for each data
- Trust model is application specific
  - keys may have different privileges
- Trust may go across different namespaces

NDN Insight

- Trust model can be defined in a set of relationships between data names and key names

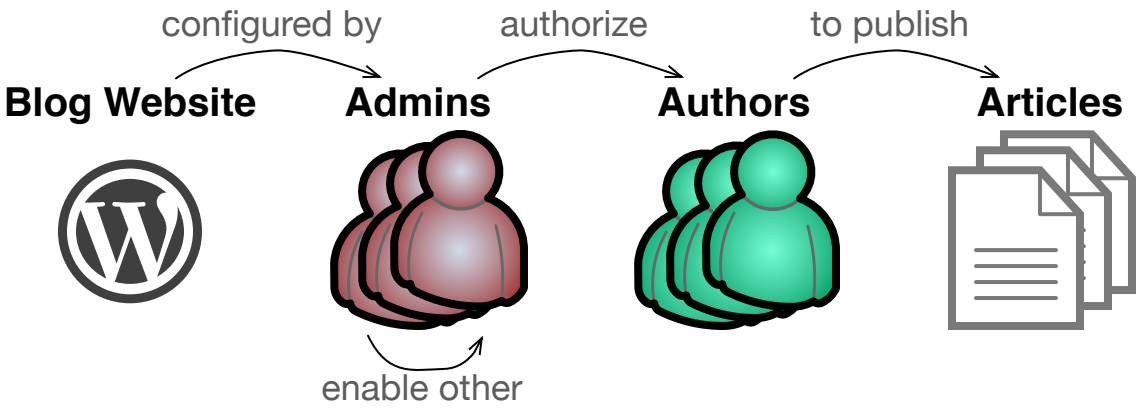


Usable Security

- Need to be easily expressible
  - trust model is application specific
  - given a trust schema, anyone can authenticate data
    - consumers, dedicated storages, routers, …
  - help producers to sign data
- Need to be automated
  - otherwise developers will “temporarily” disable security
    - fake signature, no authentication
- Better to be re-usable
  - applications may share the same trust model

Trust Between Entities

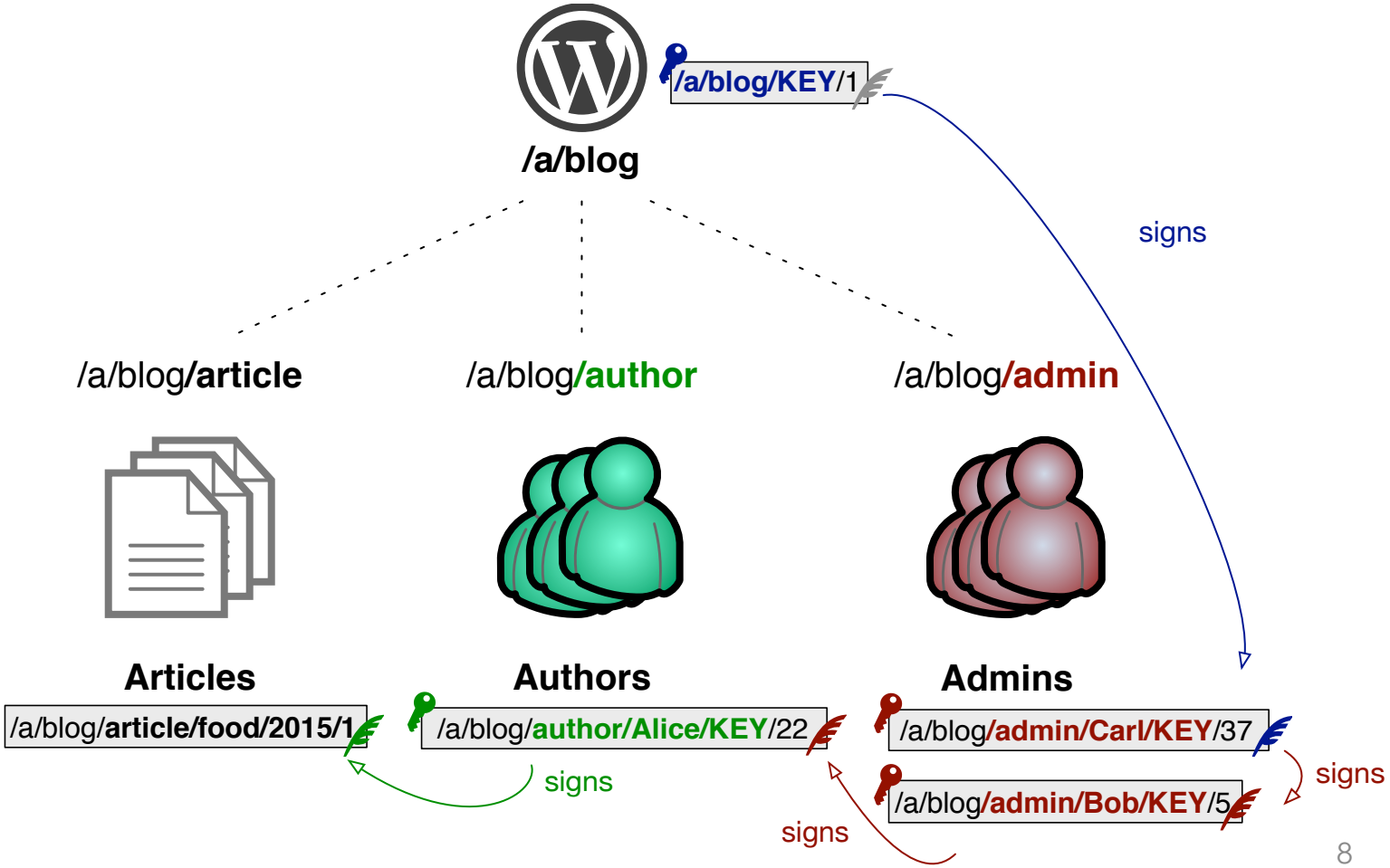
- Blog website framework
  - used by many people to set their own website



  - authors can publish articles
  - admins can create author account
  - blog configuration and admins can designate other admins

Name-based Trust

- Blog framework namespaces

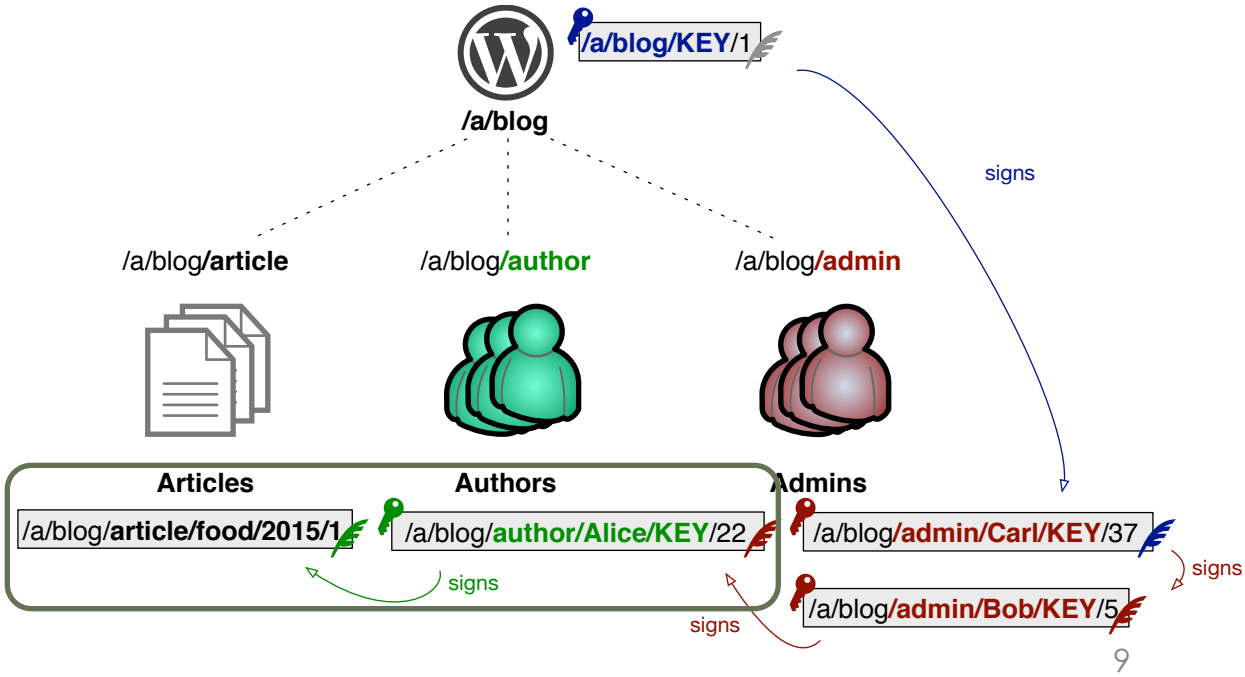


Generalize Trust Relationship

- Relationship between data and key names

/a/blog/article/food/2015/3 ← /a/blog/author/Alice/KEY/22

/a/blog/article/drink/2014/9 ← /a/blog/author/Zach/KEY/5



Generalize Trust Relationship

- Relationship between data and key names

  /a/blog/article/food/2015/3 ← /a/blog/author/Alice/KEY/22

  /a/blog/article/drink/2014/9 ← /a/blog/author/Zach/KEY/5

- Generalize relationship

  blog_prefix + "blog" + "article" + category + misc_info

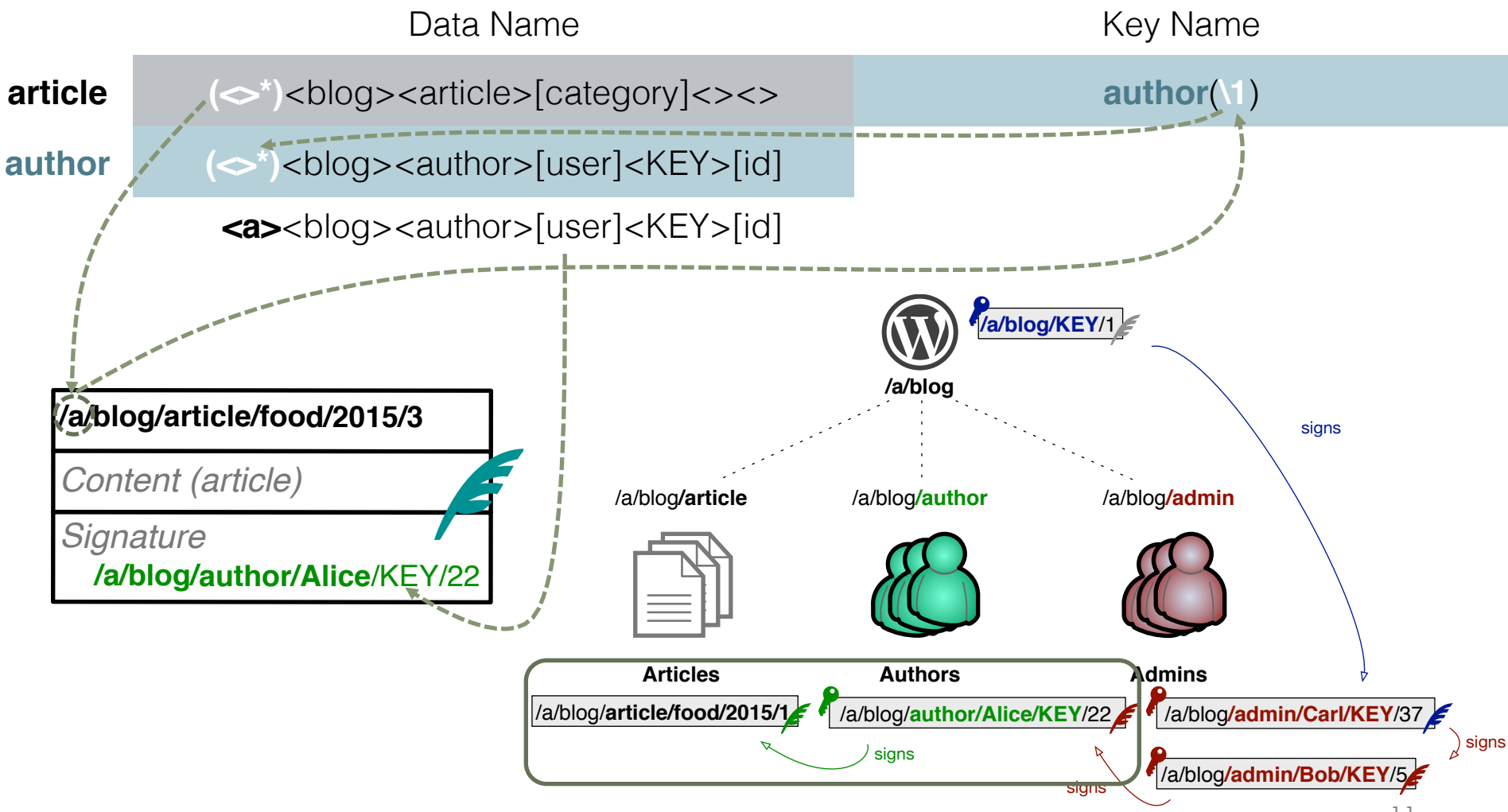
  blog_prefix + "blog" + "author" + name + "KEY" + key_id

- Regex-based syntax

  (<>*)<blog><article>[category]<><>

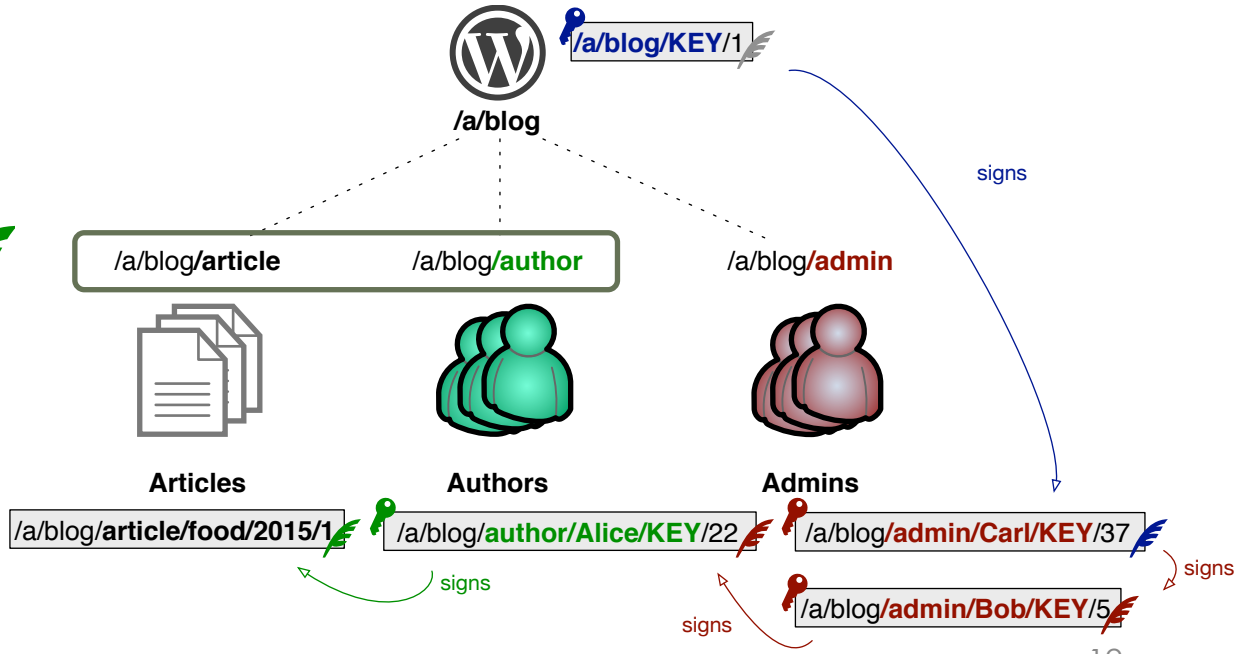
  \1<blog><author>[user]<KEY>[Id]

Key Name Pattern Derivation



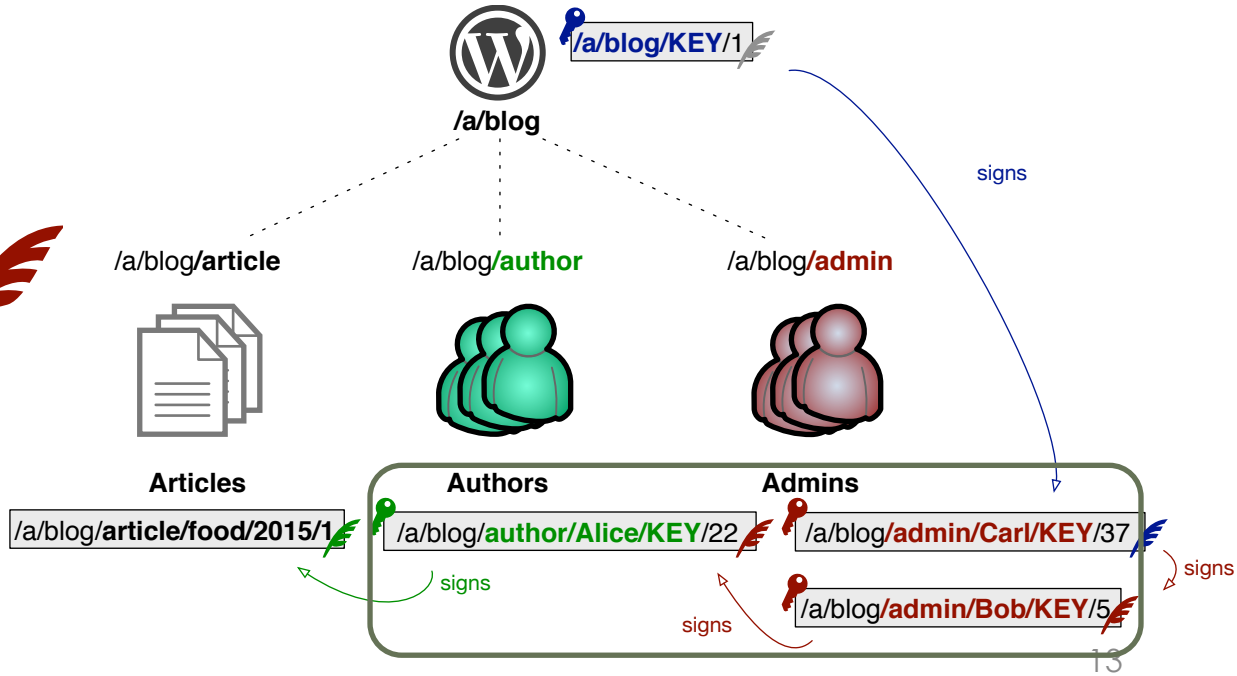
Enforce Least Privilege

	Data Name	Key Name
article	(<>*)<blog><article>[category]<><>	**author**(\1)
author	(<>*)<blog><author>[user]<KEY>[id]	



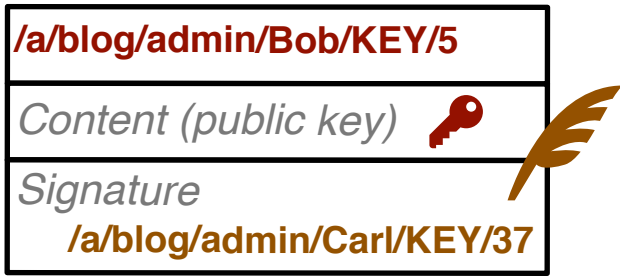
Link Trust Relationship

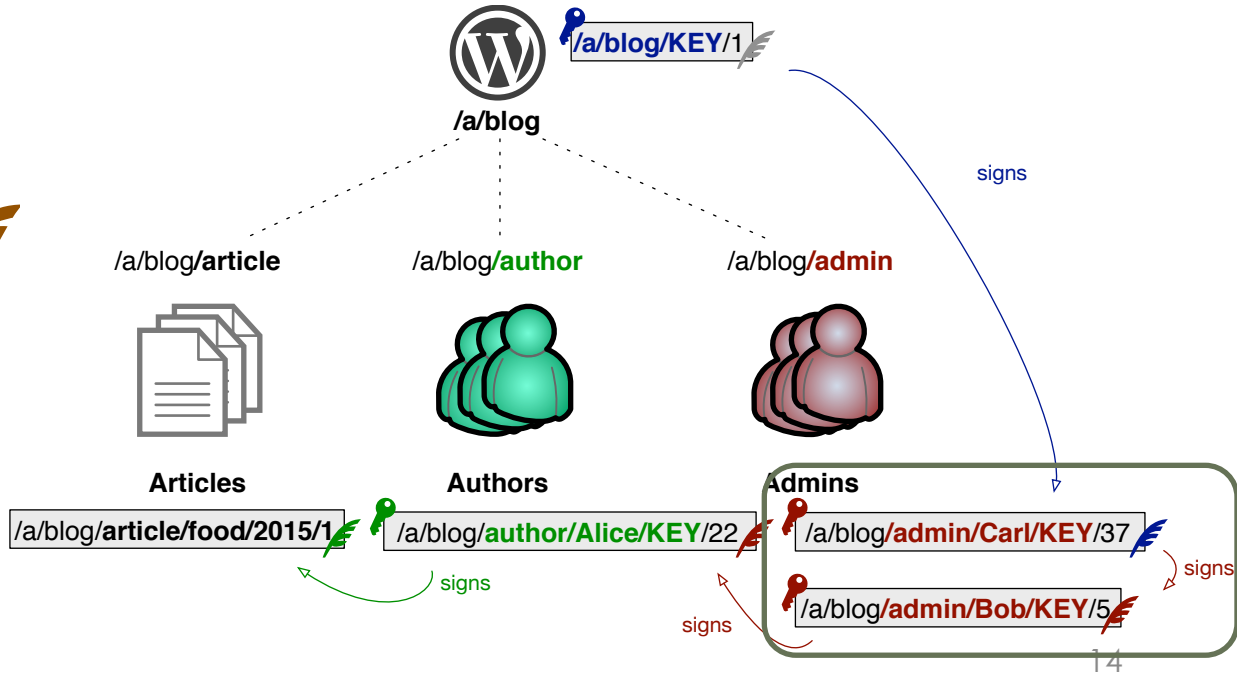
	Data Name	Key Name
article	(<>*)<blog><article>[category]<><>	**author**(\1)
author	(<>*)<blog><author>[user]<KEY>[id]	**admin**(\1)
admin	(<>*)<blog><admin>[user]<KEY>[id]	



Multiple Trusted Signers

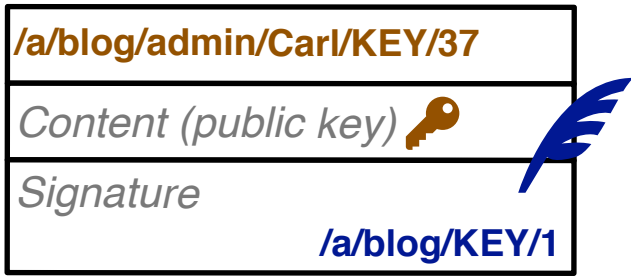
	Data Name	Key Name
article	(<>*)<blog><article>[category]<><>	**author**(\1)
author	(<>*)<blog><author>[user]<KEY>[id]	**admin**(\1)
admin	(<>*)<blog><admin>[user]<KEY>[id]	**admin**(\1)

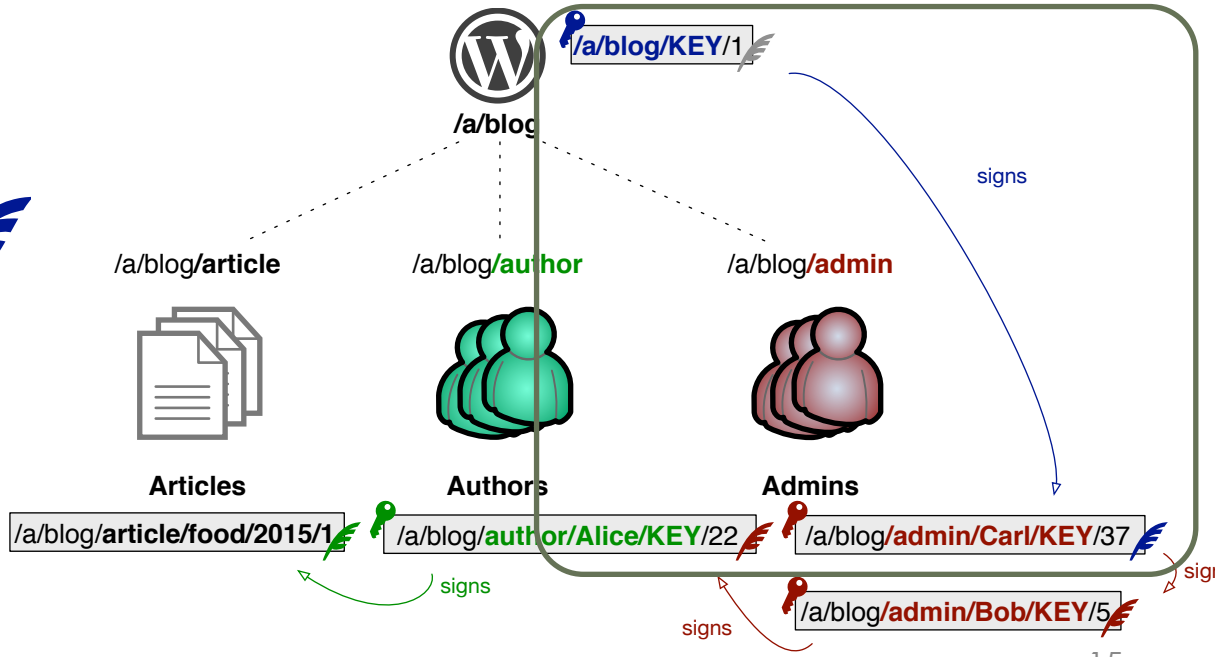




Link Trust Anchor

	Data Name	Key Name
article	(<>*)<blog><article>[category]<><>	**author**(\1)
author	(<>*)<blog><author>[user]<KEY>[id]	**admin**(\1)
admin	(<>*)<blog><admin>[user]<KEY>[id]	**admin**(\1)





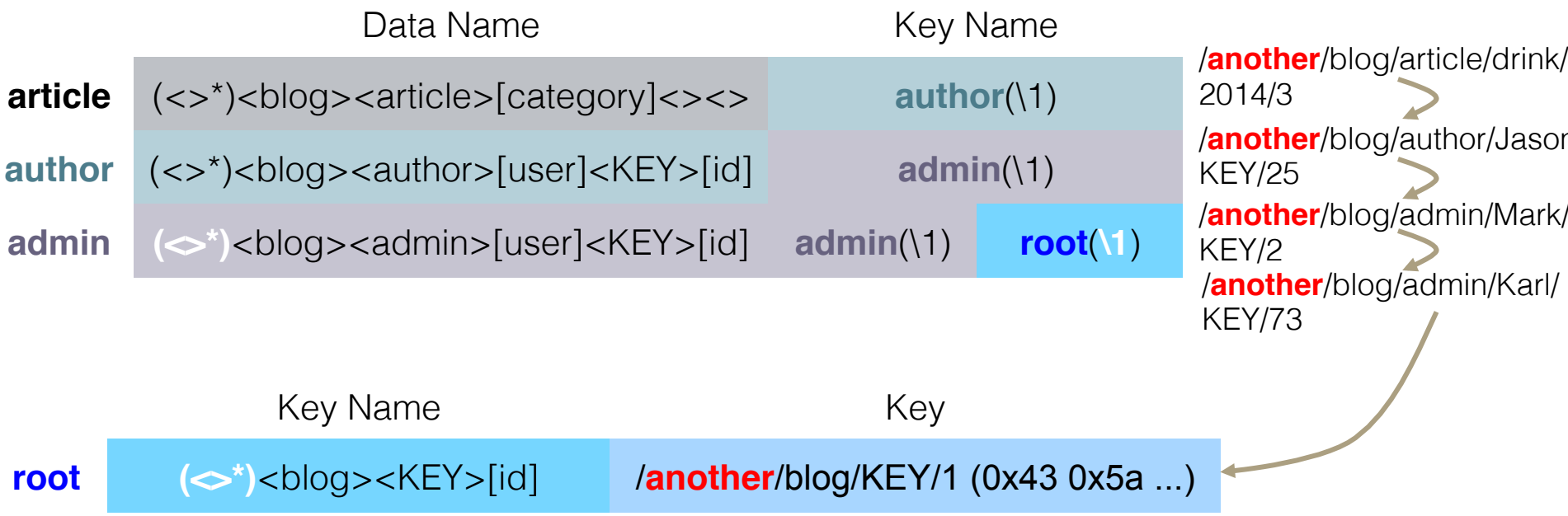
Trust Schema

	Data Name	Key Name	
article	(<>*)<blog><article>[category]<><>	**author**(\1)	
author	(<>*)<blog><author>[user]<KEY>[id]	**admin**(\1)	
admin	(<>*)<blog><admin>[user]<KEY>[id]	**admin**(\1)	**root**(\1)

	Key Name	Key
root	(<>*)<blog><KEY>[id]	/a/blog/KEY/1 (0x30 0x82 ...)

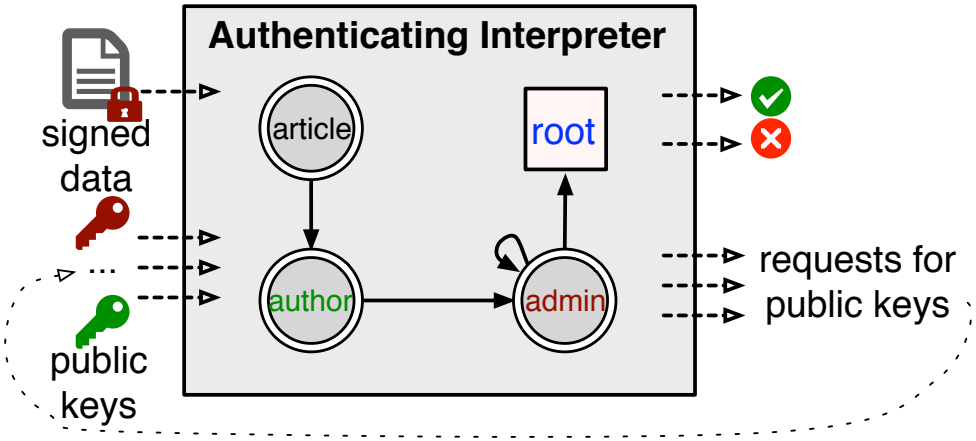
Different trust anchor for different blog website

Re-usability

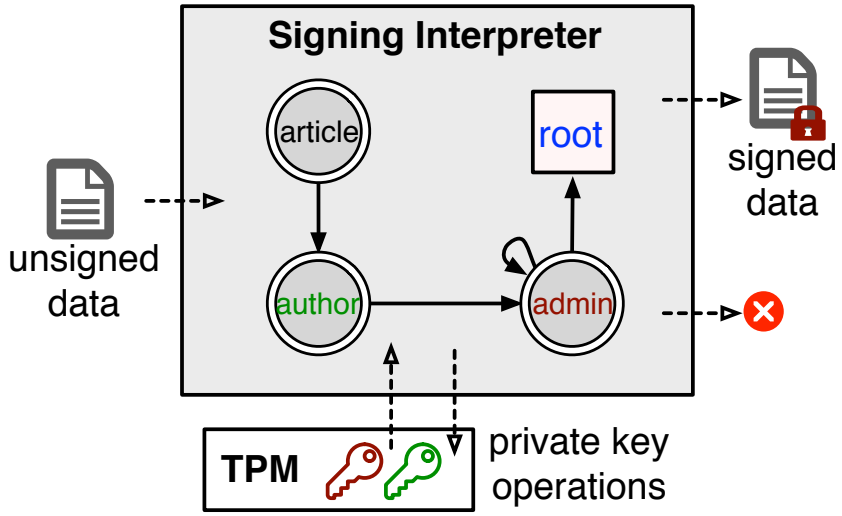


Automation

- Trust schema → FSM

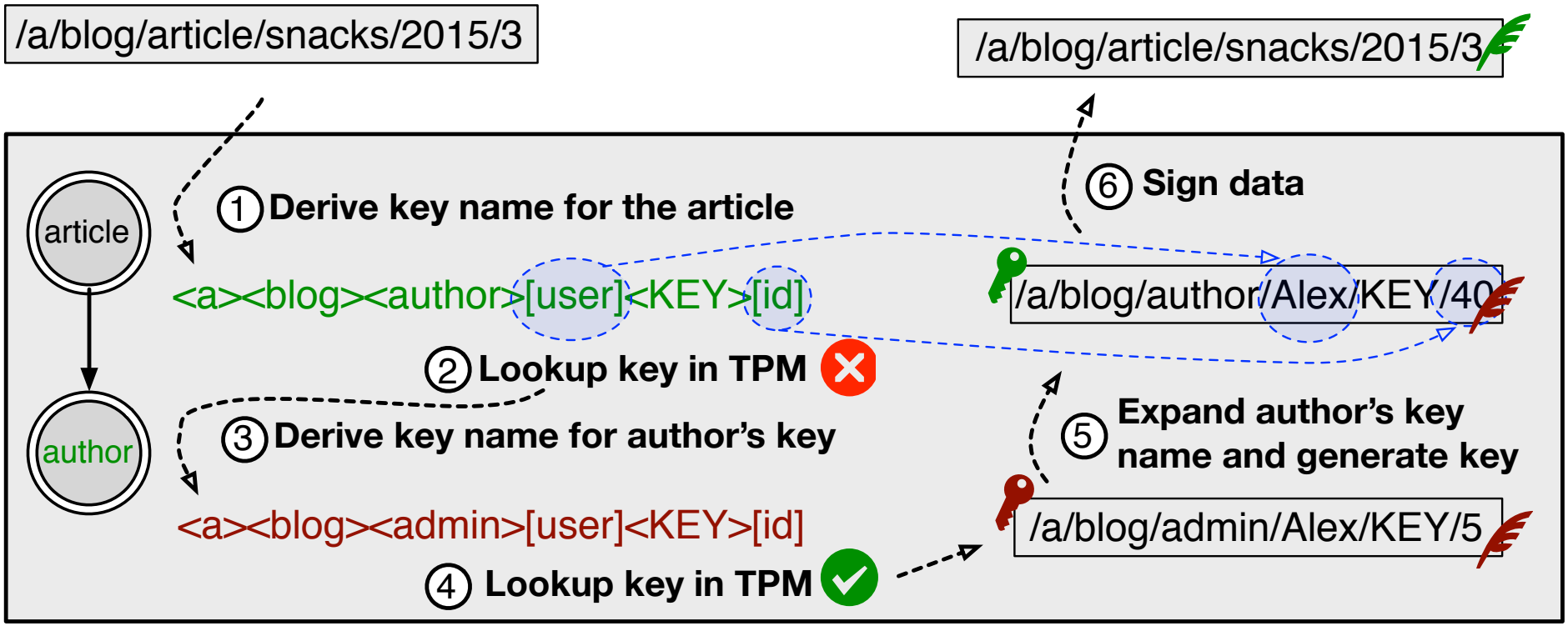


Authentication



Signing

Automated Signing



Trust schema is more than that …

- Universal tool for trust management
- Representable in a data packet
  - can be retrieved and executed by **any** NDN entity
    - end application, dedicated storage, routers, …
  - can be (recursively) authenticated using higher-level schemas
- Security design pattern
  - regulate the behavior of applications
  - a set of common trust models
  - application developer can simply select a pre-defined trust model

Implementation

- Available in all the NDN platform libraries
  - ndn-cxx: http://www.github.com/named-data/ndn-cxx
    - old schema (ValidatorConf)
    - new schema implementation in the upcoming release
  - NDN-CCL: http://named-data.net/codebase/platform/ndn-ccl/
    - NDN-CPP, NDN-JS, PyNDN, jNDN

- Powers data and interest authentication in:
  - NFD: NDN Forwarding
  - NLSR: NDN Link State Routing Protocol
  - NDNS: NDN Domain Name System
  - Repo-ng: NDN Data Repository
  - ChronoChat: server-less multi-party chat application over NDN

Conclusion

- Usability is critical to all security solutions
- A useful step forward in automating NDN data signing/authentication
  - explicitly defines trust relations between namespaces
  - identify common security patterns to generalize solutions
- Trust schema can be authenticated and fetched as any other NDN data packets
- Potentially applicable to other configuration/ automation challenges