Schematized Trust

Design and Application

NDNcomm 2015
September 28, 2015

Alex Afanasyev
University of California, Los Angeles

Overview

NDN architecture mandates signature

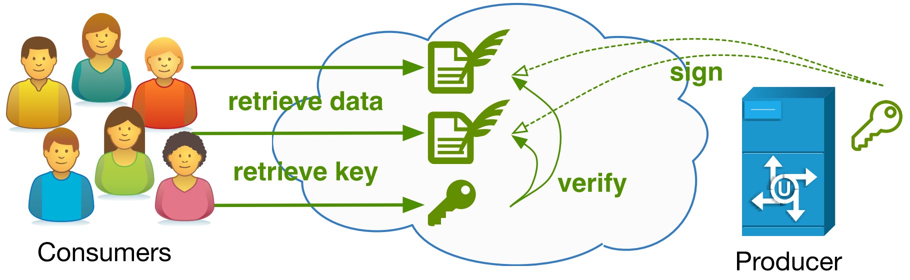
- Effectiveness of the mandate depends on the implementation
- If too complex, developers will shortcut
  - “temporarily” disable
  - use non-secure/fake signatures

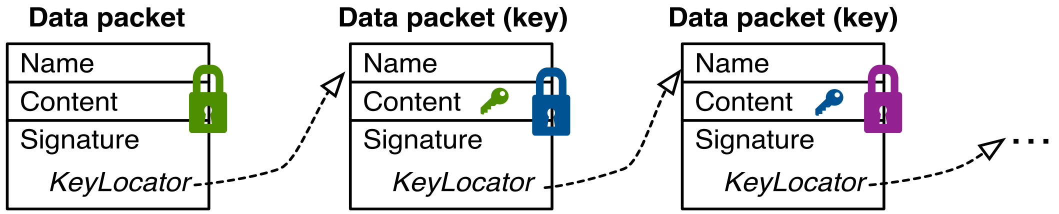
Need a tool to make security usable

need automation

Data-Centric Security in NDN

- Data is named and is retrieved using name
- Name and content are bound together with a crypto signature
- Data packet includes a name of the public key to verify the signature
    - Key is also a data packet and retrievable by name





Data Authentication

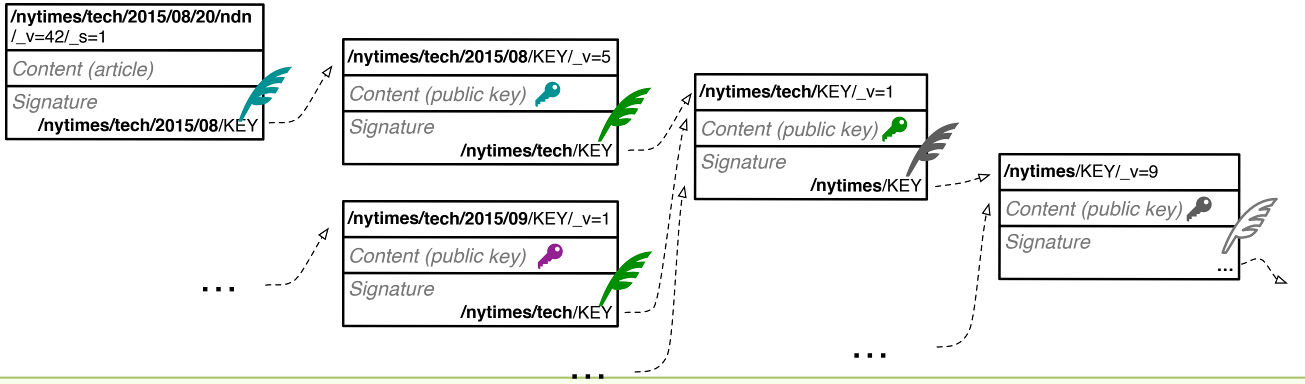
- To authenticate data, one needs a trust model
  - which keys are authorized to sign which data (trust rules)
  - one ore more trusted keys
  - requires crypto properties

- Given trust model, anybody can verify data
  - applications
  - dedicated storage
  - routers

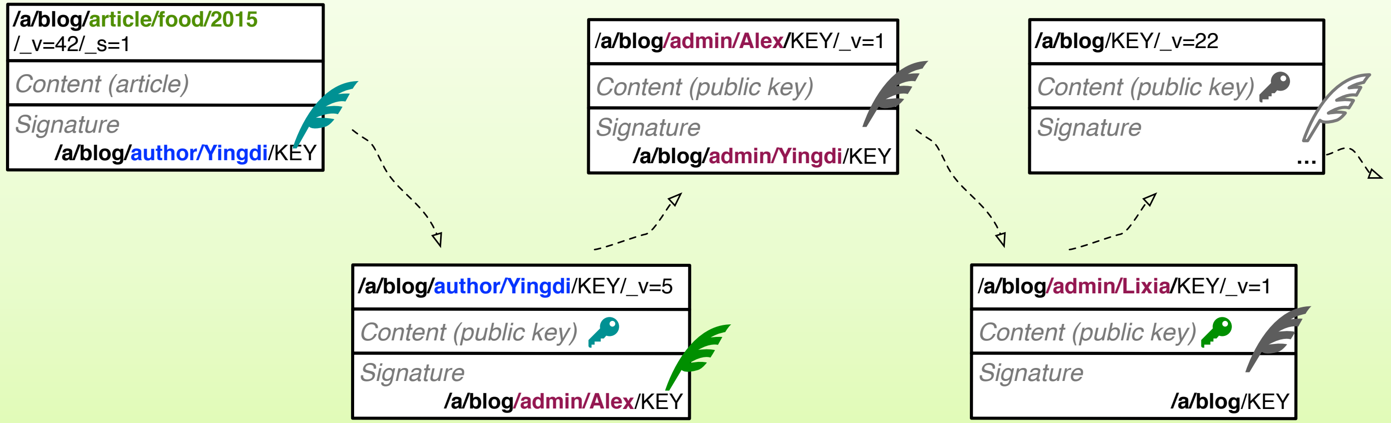
- **Trust model needs to be easily expressible**
  - help consumer to authenticate data
  - help producers to sign data

NDN Insight: Trust can be defined as a set of relationships between data and key names

Hierarchical trust relations



Cross-namespace trust relations

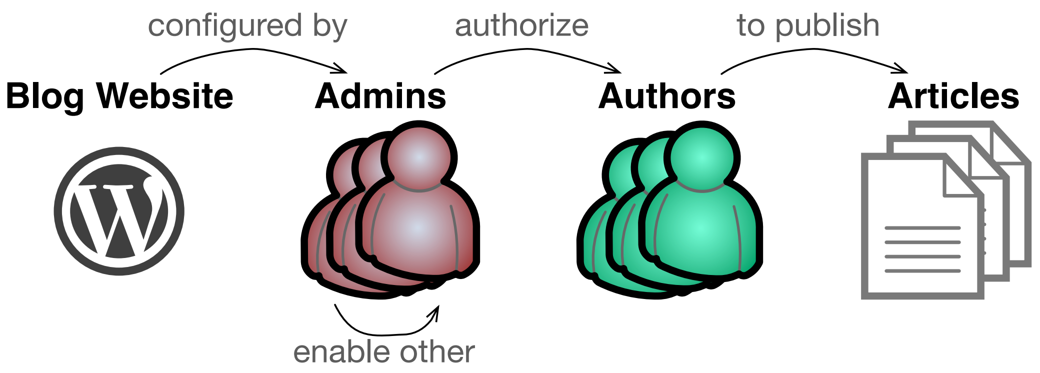


Desired Properties for Trust Policy Definition

- Clear definition of relationship rules
  - Use names and name patterns to define rules
    - data with `/some/site` prefix can be only signed with `/some/site/key/<any-id>`
    - keys `/some/site/key/<any-id>` can be only signed with `/another/key/id=5`
  - Pre-configured trust anchors to bootstrap trust
    - `/anoth`
- Least privileg
  - Limited usage scope
  - Limited time-span
- Re-use of trust models between applications
  - Define, debug, and refine common trust models
- Make security easy to use

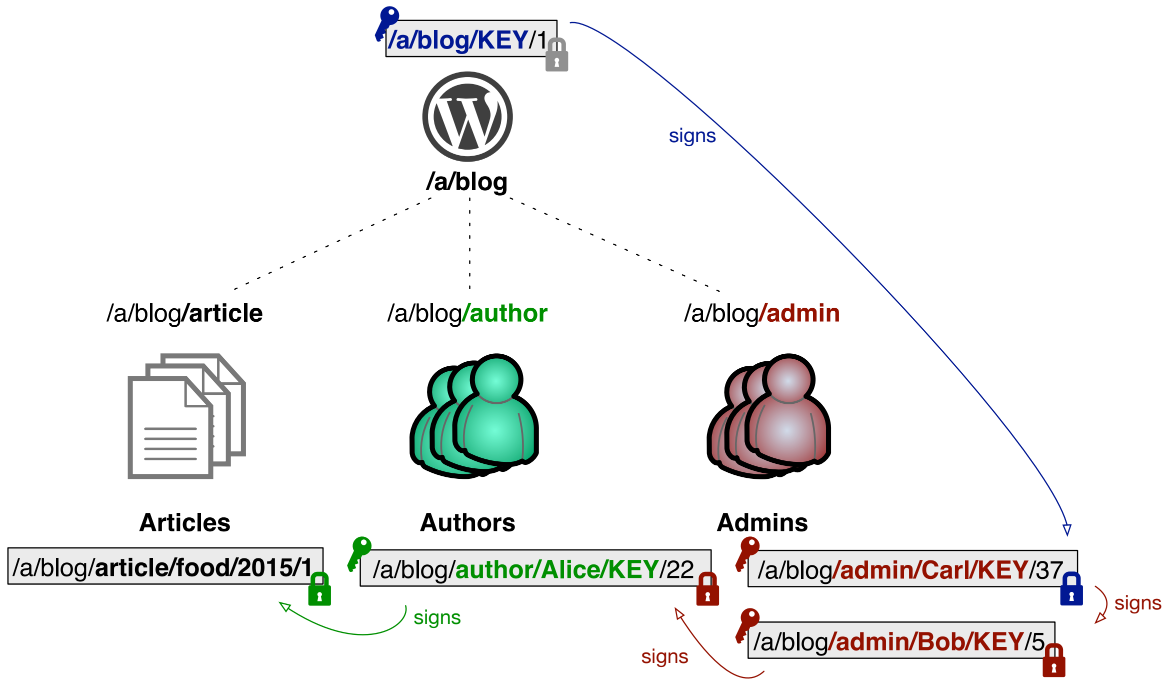
Trust Schema to Schematize and Generalizing Trust

Example: Web Blog



- Articles authored and signed by authors
- Authors are given permissions to publish on the blog by administrators
- Administrators are configured by blog configuration or other administrators

Web Blog: Name-Based Trust Relationships



- Articles authored and signed by authors
- Authors are given permissions to publish on the blog by administrators
- Administrators are configured by blog configuration or other administrators

Generalized Rules for Name-Based Trust

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

Generalizing relationship

- **blogPrefix** + “blog” + “**article**” + **category** + miscInfo <->
  - **blogPrefix** + “blog” + “**author**” + **name** + “KEY” + keyid

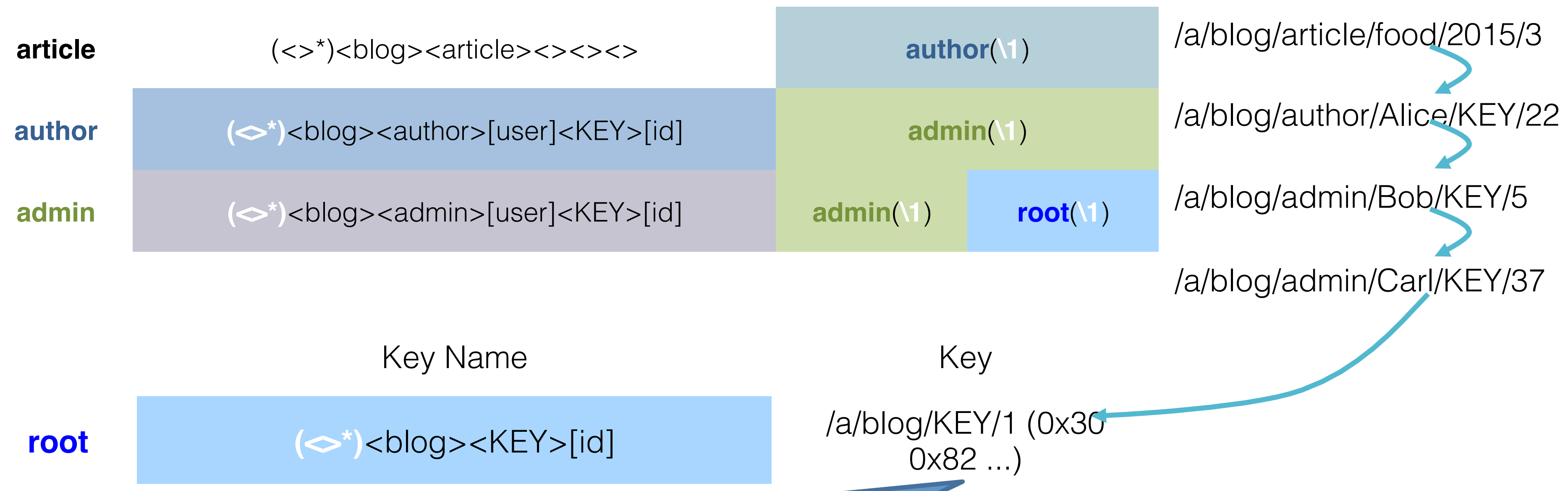
Use regular-based syntax to capture the relationship

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

Web Blog: Trust Schema

Regex-like pattern with grouping
(group values accessible as \1, \2, \3 ...)

Name or other rule specializations



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

/a/blog/article/food/2015/3

/a/blog/author/Alice/KEY/22

/a/blog/admin/Bob/KEY/5

/a/blog/admin/Carl/KEY/37

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

Different trust anchor for different blog website

Trust Rule Processing

/a/blog/article/food/2015
/_v=42/_s=1

Content (article)

Signature
/a/blog/author/Yingdi/KEY

author (<>*)<blog><author>[user]<KEY>[id] **admin**(\1)

/a/blog/article/food/2015/3 =>> \1 = /a

article must be signed with the key with name expanded from **author("/a")**

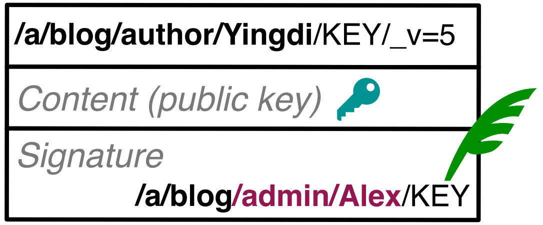
[user] -> accepts any user name (auth)
-> generates use name (keygen)

[id] -> accepts any key id (auth)
-> generates unique key id (keygen0

author (<>*)<blog><author>[user]<KEY>[id]

<a><blog><author>[user]<KEY>[id]

Trust Rule Processing


/a/blog/author/Yingdi/KEY/_v=5

Content (public key)

Signature
/a/blog/admin/Alex/KEY

author	(<>*)<blog><author>[user]<KEY>[id]	**admin**(\1)

/a/blog/author/Yingdi/KEY/_v=5 =>> \1 = /a

author key must be signed with the key with name expanded from **admin("/a")**

admin	(<>*)<blog><admin>[user]<KEY>[id]

<a><blog><admin>[user]<KEY>[id]

Trust Schema Implementation Status

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

Trust schema powers data and interest authentication in

- NFD: NDN Forwarding
- NLSR: NDN Link State Routing Protocol
- Repo-ng: NDN Data Repository
- ChronoChat: a chat application over NDN
- NDNS: NDN Domain Name System

Works! Even better implementations coming really soon

Making Trust Schema Universal Tool for Trust

Captures data/key name relationships using generalizations and patterns

- formally describes and defines trust model
- enforces trust model in automatic way
  - both authentication and signing paths

Representable in a data packet

- can be retrieved and executed by **any** NDN entity
- can be (recursively) authenticated using higher-level schemas

Trust schema also defines security design pattern

- regulate the behavior of applications
  - an operating system can define a trust schema to authenticate the trust schema of applications
  - only install and execute apps with authenticated trust schema