Schematized Trust in NDN

Van Jacobson

FIA-NP Meeting
Arlington, VA
May 19-20, 2014

- Today we (attempt to) secure the <u>process</u> of communication by adding cryptographic wrappers to the packet transport.

- This hasn't worked well. One serious failing is an intrinsic one-size-fits-all model of trust based on endpoint identity.

- Information-centric architectures secure content, not just the process of communicating it. They have the potential to support richer and more granular trust.

- To be successful, content-based trust machinery must be easy to understand, configure and use.

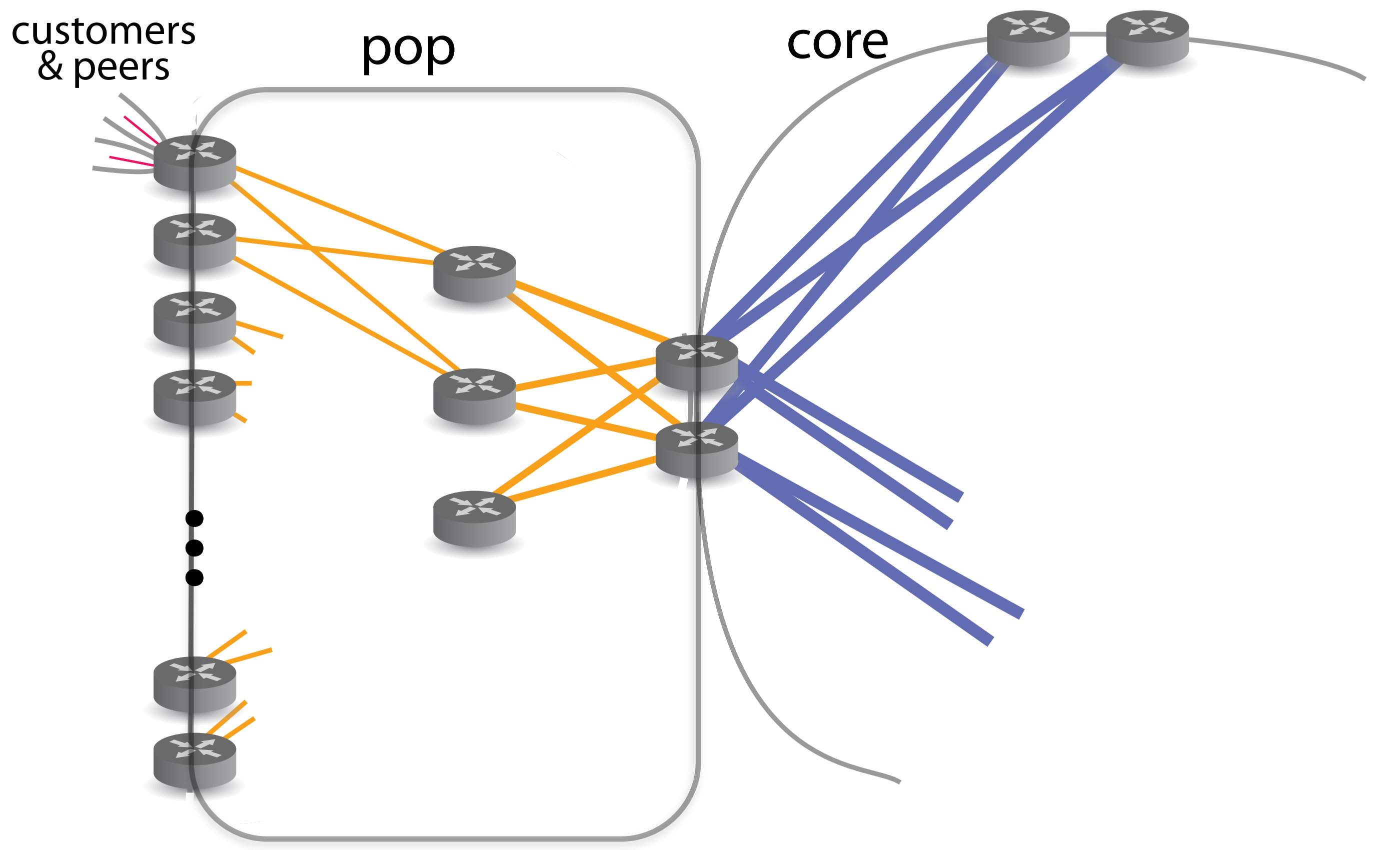
- Simple ‘trust schemas’ (patterns / templates / …) that can applied to whole classes of applications would help achieve this.

- As a proof-of-principle, NDN has developed a schematized trust framework and successfully applied it IGP routing and IoT (building control and instrumented environments).

A router in an ISP PoP typically participates in multiple routing instances, often administered by different groups.

Since routing protocols broadcast, software or hardware misconfiguration can cross-connect instances.

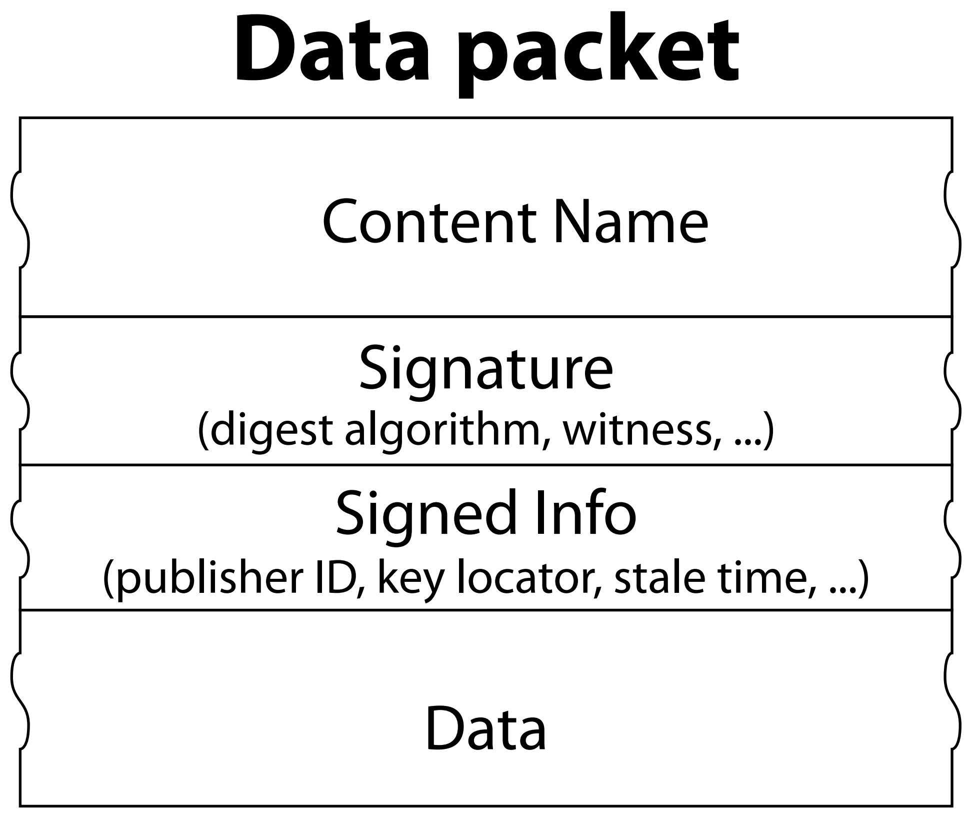
This can create a configuration and maintenance nightmare.



NDN packets

NDN Data packets are structured objects with three parts:

- (opaque) data bytes
- A name for the data
- A signature over the name and data together with the name of the signing key (another NDN packet).


Data packet
Content Name
Signature
(digest algorithm, witness, ...)
Signed Info
(publisher ID, key locator, stale time, ...)
Data

BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/LSP/678

Name of a Link State Packet generated by the OSPF routing process running on SFpop router 72 signed by

BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/LSP/678

BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/key

Public key given to the OSPF routing process when it was started by the router. Every packet sent by the process is signed with this key.

signed by

BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/LSP/678

BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/key

Public key given to the OSPF routing process when it was started by the router. Every packet sent by the process is signed with this key.

BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/LSP/678

BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/key

BigCo/NetOps/SFpop/rtr/72/key

Public key given to the router when it was configured.

BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/LSP/678

BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/key

BigCo/NetOps/SFpop/**rtr/72**/key

BigCo/NetOps/SFpop/**config/empl/975**/key

Public key given to the employee who configured the router.

BigCo/NetOps/SFpop/**config**/key

Public key authorizing SFpop router configuration. (SFpop trust root)

Trust Schema

k4 = my.config.root	BigCo/NetOps/SFpop/config/key
k3 = k4 +“empl”+ n	BigCo/NetOps/SFpop/config/empl/975/key
k2 = k3[-4] +“rtr”+ n	BigCo/NetOps/SFpop/rtr/72/key
k1 = k2[-3] +“OSPF”+ k2[2-1] +“pid”+ n	BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/key
pkt = k1 +“LSP”+ n	BigCo/NetOps/SFpop/OSPF/rtr/72/pid/345/LSP/678

Usage

```
if (validTrustChain(pkt, schema) && signatureValid(pkt))
    process the packet
```

Since schema is just lexical constraints on key names, validation normally only has to check that key name is appropriate for data name.

Only have to validate chain & signature for a key once.

Why so many names?

- Context provided by naming detects and prevents misconfiguration and misbehavior.
- Names provide fine-grain trust that minimizes damage from key exposure.
- Naming strictly limits scope of keys and prevents repurposing.

Model Properties

- Complete local autonomy - all keys are locally generated and signed.
- No key distribution problem. Apps get their entire trust chain from router’s config then announce any new keys to their peers.
- Once a trust schema has been picked, everything else is simple and automatic.