Addressing Security and Privacy Issues in Named Data Networking
Communication

• For almost 150 years, communication meant:
  A wire connecting two devices

• The Web forever changed that:
  What matters is content, not the host it came from
Today’s Internet

- Tremendous, unexpected and surprisingly lasting global success story
- Architecture defined in RFC 791/793 (1981)
- Enables any host to talk to any other host
  - Names boxes and interfaces
  - Supports end-to-end conversations
  - Provides unreliable packet delivery via IP datagrams
  - Compensates for simplicity of IP via complexity of TCP
• Helped facilitate today’s content-centric world but was never designed for it

• Fundamental communication model: point-to-point conversation between two hosts (IP interfaces)

• The central abstraction is a host identifier corresponding to an IP address
Today’s Internet

• Last 20 years – profound change in nature of Internet communication
  o From email/ftp/telnet content, content and more content

• Massive amounts of data constantly produced and consumed
  • Web (esp. media sharing and social networking), audio-/video-conferencing, email, etc.
## DN vs. CN

<table>
<thead>
<tr>
<th></th>
<th>Communication</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Naming</strong></td>
<td>Endpoints</td>
<td>Content</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>Invisible, Limited</td>
<td>Explicit; Storage = Wires</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Communication process</td>
<td>Content</td>
</tr>
</tbody>
</table>

Today’s Internet: a communication network, used as a distribution network
Future Internet Architecture (FIA)

- NSF program: on-going, started Fall 2010

- Five projects + architectures:
  - MobilityFirst
  - XIA
  - NDN
  - ChoiceNet (2012)
  - Nebula
Named-Data Networking (NDN)

- An instance of ICN and CCN

NDN targets Content Distribution which is poorly served by today's Internet
Who is NDN?
Future Internet Architectures (FIA)

• Security and privacy in current Internet are NOT a success story

• Retrofitted, incremental, band-aid-style solutions
  ○ E.g., SSH, SSL/TLS, IPSec, IKE, AAA, etc.

• NSF FIA places emphasis on S&P from the outset

• S&P features prominent in all five FIA architectures
Content Distribution over IP

ISP

ISP

YouTube
NDN Basic Concepts

• Name
  ➢ Human-readable, path/url-like

• Roles:
  ➢ Consumer
  ➢ Producer
  ➢ Router

• Objects:
  ➢ Content
  ➢ Interest
As opposed to IP

• Host
• Interface address (IP address)
• Datagram/Packet
• Router
What’s in a name?

Human Readable:

User/App supplied name

/G/parc.com/videos/WidgetA.mpg/\v <timestamp>/s3

Globally-routable name
Organizational name
Conventional/automatic

Binary Encoding:

| 6 | 8 | parc.com | 6 | videos | 11 | WidgetA.mpg | 7 | FD04A... | 2 | 0003 |

Implicit Hash
How NDN works
(reader’s digest version)

- Carries content name
- No source/destination address

• Named data (content)
• Routed using state
### Addressing Security and Privacy Issues in Named-Data Networking

**Zooming In:**

<table>
<thead>
<tr>
<th>Interest</th>
<th>Incoming face</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ndn/uci/content</td>
<td>face0, face3</td>
</tr>
</tbody>
</table>

Every router has a:
- PIT: Pending Interest Table
- CS: Content Store (Cache)
- FIB: Forwarding Information Base
Forwarding

- Key operation is prefix-based longest match lookup, like IP
- Interests forwarded according to routing table, but multipoint forwarding, broadcast, local flooding are all okay
- Data follows interest path in reverse
Routing

- Routing based on name prefixes + reachability, as in IP
- Can reuse IP routing protocols, e.g., IS-IS, BGP
Live demo: www.arl.wustl.edu/~pcrowley/NDN_GEC13_demo.mp4
• **Now**: secure the pipe
  • Data is authentic because it emanates from the right box (which is an end-point of the right secure pipe)

• **NDN**: Integrity and trust as properties of content
  • Should be inferred from content itself
Securing Content: how?

Current SSL/TLS model not a good fit for NDN:

- Secures channel, not data
- Authentic content can come from anywhere
- But, access control (and accounting) is difficult
- After content retrieved from origin, it’s served by the network (from caches)
Authenticity of Content

Content can be retrieved from anywhere by any consumer

• How can it be trusted?
• How do we know who produced it?
• How do we know it is the right content?
Securing Content

NDN Content object:

- **Integrity**: is data intact and complete?
- **Origin**: who asserts this data is an answer?
- **Correctness**: is this an answer to my question?
- **Bonus feature**: routers can choose to verify content (with caveats)
Securing Content - Performance

• Signing/verifying every content is expensive

• Can reduce costs (a little) via techniques like:
  – MHTs, hash chains, etc.
  – Online/offline signatures
  – Probabilistic verification (spot-checking)
Private Content

Access to content can be restricted, e.g.:

- Encrypt once with a symmetric key
- Symmetric key distributed using “standard” techniques
- Access control on key rather than content
  - This can make long-term secrecy problematic
Trust Model?

• All content is signed
• **Interests are not**…
• NDN is PKI-agnostic
• Application-specific vs network-layer trust
NDN: Privacy Benefits

- Interest has no source address/identifier
- Content can be routed without knowing consumer identity and/or location
- One observed interest may correspond to multiple consumers at various locations
- Router caches reduce effectiveness of observers close to producers
NDN: Privacy Challenges

- Name privacy in interests
  `/ndn/us/wikipedia/STDs/herpes`
- Name privacy in content
  `/ndn/zimbabwe/piratebay/XSOQW(#E@UED$%/.mp3`
- Signature privacy
  -Leaks content publisher identity
  -Classical privacy vs. security conflict
- Cache privacy
  -Detectable hits/misses
NDN: Security Benefits

- Simplicity
- All content is signed
- No need for security handshakes in real time
- A producer’s public key is a type of content
  - Pull it first, then request content
NDN: Security Challenges

• Router state is both a blessing and a curse
• Any such state can be abused
• DoS attacks:
  – Interest Flooding
  – Content Poisoning: proactive & reactive
• Trust management at the network layer
NDN: quick recap

PRODUCER
• Announces name prefixes
• Names and signs content packets
• Injects content by answering interests

CONSUMER
• Generates interest packets referring to content by name
• Receives content, verifies signature, decrypts if necessary

ROUTER
• Routes interests based on (hierarchical) name prefixes – inherently multicast
• Remembers where Interests came from (PIT), returns content along same path
• Optionally caches content (in CS)
• May verify content signatures
Why Name Privacy?

NDN names are expressive and meaningful, but...

- Leak information about requested content
- Easy to filter/censor content, e.g., block everything like:
  
  /ndn/cnn/world-news/china/

However:

- NDN names are opaque to the network
- Routers only need to know name component boundaries
- Names can carry binary data
ANDaNA: Anonymous Named Data Networking Application

- Observers close to consumer should not learn what content is being requested
- Low-to-medium-volume interactive communication
- Producers may not be aware of ANDaNA
ANDaNA
ANDaNA
ANDaNA

The New York Times

/OR1

/OR2

The New York Times
ANDaNA

- Privacy with 2 hops comparable to Tor with 3
  - Why? Lack of source address in interests
  - Anonymizing routers do not learn origin of traffic (only the previous hop)
  - Lower overhead
Router Caching is good for performance
  • Better bandwidth utilization
  • Lower latency
But… bad for privacy
  – Timing attacks
  – Cache harvesting attacks
Cache Privacy

- Who could the adversary be?
  - Another host or router
  - A malicious application on victim’s device

- Where could the adversary be?
  - Near consumer, e.g., same LAN/WLAN
  - Near producer (opposite sides of first hop router)
  - In both places at once
Scenario 1: Victim=Consumer

/ndn/org/wikileaks/2012/july/31
Scenario 2: Victim=Producer

Consumer → Interest → Interest → Interest → Interest → Interest → Producer

Adversary
Scenario 3: Victims=Both

Are Alice and Bob talking?
Countermeasures

• Do not cache content at all
  • Bad idea…

• Cache and delay
  • Which content? Who decides?
  • How long to delay?
Countermeasures

• Two types of traffic:
  • Private
  • Non-private

• Two communication types:
  • Low-latency (interactive) traffic
    • Use unpredictable content names
  • Multicast (distribution) traffic – details
    • Random delay
    • Content-specific delay

• Introduce a privacy bit in interests and/or content?
DoD/DDoS Resistance?

Some current DoS + DDoS attacks become irrelevant because of NDN architecture

- Content caching mitigates targeted DoS
- Content is not forwarded without prior state set up by interests
- Multiple interests for same content are collapsed
- Only one copy of content per “interested” interface is returned
Interest Flooding

Adversary generates numerous non-sensical interests, e.g.:

```
/ndn/legitimate-producer/random-string
```

- Consumes precious router resources (PIT entries)
- Affects both routers and producers
Interest Flooding

Potential countermeasures:

1. Unilateral rate limiting/throttling
   - Resource allocation determined by router state

2. Collaborative rate limiting/throttling
   - Routers push back attacks by interacting with neighbors
Content Poisoning...

- NDN objective is content distribution
- Facilitated by caches + PITs in routers
- Consumer must verify content signatures
- Routers are not obliged to verify signatures
- But ... how to flush fake content from router caches?
  - Immediate flush: **DoS**
  - Verifying signature: expensive + another DoS type
- Consumer authentication contradicts interest opacity
Keys in NDN

- A public key is a type of content
- Contains authorized name prefixes
- For example:
  /cnn/usa/web/key OR /verisign/europe/key
Content Poisoning

• Two reasons:
  • Ambiguous interests
  • No unified trust model: applications are a diverse and dynamic set

AXIOM: Network-layer trust and content poisoning are inseparable in NDN

• Routers should do minimal work:
  • not verify/validate public keys (except for routing)
  • verify at most one signature per content
Interest-Key Binding Rule (IKB)

**IKB:** An interest must reflect the trust context of the consumer, easily enforceable at the network layer.

**IKB (NDN):** An interest must indicate the public key of the content producer.
Interest-Key Binding Rule (contd.)

IKB (NDN): An interest must indicate the public key of the content producer

- PublisherPublicKeyDigest (PPKD) field mandatory in every interest

- Consumers obtain and validate keys, using
  - Pre-installed root keys
  - Key Name Service (KNS)
  - Global search-based service
Interest-Key Binding Rule (contd.)

- **Producer:**
  - Includes public key in each content’s `KeyLocator` field

- **Router:**
  - Matches `KeyLocator` digest to PPKD in PIT
  - Verifies signature using `KeyLocator`
  - No fetching, storing, parsing of public keys

→ Note: PIT entry collapsing takes PPKD into account
CLAIM:
Adherence to IKB $\Rightarrow$ security against content poisoning

- Assume:
  - All nodes abide by IKB
  - Consumer not malicious
  - Consumer-facing routers – not malicious
  - Consumer (logical) link to facing routers – not compromised
Is this Secure?

- Consumer sends interest containing PPKD
- Router ensures that:
  - Valid content signature using key in KeyLocator
  - Digest of KeyLocator matches PPKD in PIT
- Consumer-facing router not malicious ➞ only possibility is hash collision

- If upstream malicious routers send fake content:
  - Facing router detects it and drop it
Optimizations

• Include keys in interest:
  ✓ Save storage
  ✗ Requires changes to interest+content structure

• Only AS border routers implementing IKB
  ✓ Better performance
  ✗ Possible fake content in caches
    But … detectable by border routers

NOTE: each router must at least do a PPKD match
Optimizations (contd.)

- Self-certifying name (SCN)
  - Hash of content (including name) is the last component of name

- Benign consumers use (SCN) ➔ network delivers “valid” content

- **No** signature verification by routers:
  - Only a hash

- How to get content hash in the first place?
Catalogs and SCN-s

Catalog:
- Contains one or more SCN-s, nesting arbitrary
- Any authenticated data structure
  - Hash chains, MHTs, skip-lists, etc.
- Structure is application-specific
- Use IKB to bootstrap (fetch catalogs)

• SCN obtained by catalogs:
  - No signature verification by routers/consumers
  - No need to sign content by producers
Consider 2 types of traffic

1. Content Distribution, e.g.:
   - Video streaming:
     - One big catalog containing SCNs of all segments
     - Or, hash chains (with data), or MHT, etc.
   - Browsing:
     - HTML file as a catalog
     - Contains SCN of sub-pages/components
     - Only for static content
2. Interactive Traffic

- Content generated on demand (real-time)
- Catalogs not viable
- Content must be requested by setting PPKD in interest
References

“Named data networking project (NDN)”, http://named-data.org

“Content centric networking (CCNx) project”, http://www.ccnx.org

V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard,

S. DiBenedetto, P. Gasti, G. Tsudik and E. Uzun,

J. Burke, P. Gasti, N. Nathan and G. Tsudik,

G. Acs, M. Conti, C. Ghali, P. Gasti and G. Tsudik,
Cache Privacy in Name-Data Networking, IEEE ICDCS 2013.

P. Gasti, G. Tsudik, E. Uzun, and L. Zhang,

A. Afanasyev and P. Mahadevan and I. Moiseenko and E. Uzun and L. Zhang,
Interest Flooding Attack and Countermeasures in Named Data Networking, IFIP Networking 2013.

A. Compagno, M. Conti, P. Gasti and G. Tsudik

C. Ghali, G. Tsudik and E. Uzun,

M. Almishari, P. Gasti, N. Nathan and G. Tsudik,
Optimizing Bi-directional Low-Latency Communication in Named Data Networking, ACM SIGCOMM CCR 2014.

M. Conti, P. Gasti and G. Tsudik,
Exploring Covert Channels in Named Data Networking, AsiaCCS 2014.