Security & Privacy in Content-Centric Networking (CCN)

Gene Tsudik
Chancellor’s Professor
CS Department
University of California, Irvine (UCI)

My Current Research Topics

- Security of Embedded Devices (ES/CPS/IoT)
- Privacy-Agile Cryptographic Techniques
  - Cloud/DB apps
  - Genomic S&P
  - Input size-hiding
- Privacy in Social Networks
- Usable Security
- Biometrics + De-authentication + Attacks
- S&P in ICN/CCN/NDN

For more info see: sprout.ics.uci.edu
OUTLINE

• Internet
• CCN Overview
• CCN Security & Privacy
• Anonymous Content Retrieval
• Cache Privacy
• Denial of Service
• Network-Layer Trust
• Other Topics?
  – Access Control, Accounting, Fragmentation, NACKs

NEED TO KNOW (for this talk)

• Basic networking & Internet concepts

• Network security principles
  – Protocols

• Basic knowledge of applied cryptography
  – Basic cryptographic primitives
Today’s Internet

• Tremendous, unexpected, unprecedented and long-lasting global success story
• 35-year-old design: architecture defined in RFC 791/793 (1981 and earlier)
• Enables any host to talk to any other host
  o Names boxes and interfaces
  o Supports end-to-end conversations
  o Provides unreliable packet delivery via IP datagrams
  o Compensates for simplicity of IP via complexity of TCP

IP-Based Internet

• Helped facilitate today’s rich global-scale communication
• But, was not 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 to …
  o From a few thousands of users to …
  o From static wired nodes (computers, terminals) to …
  o From friendly, clubby, trusting ambience, to …
• Massive amounts of data constantly produced and consumed
  • Web (esp. media sharing and social networking),
  • Audio-/video-conferencing

• Note that:
  • Email and remote login are still around
  • Messaging too
  • Plus, there’s IoT…

Key Aspects of Internet Change

• Multimedia content
• Mobility / Wireless-ness
  → Delays and Disruptions
• Distribution Scale
• Cloud
• IoT?
Internet Security & Privacy

• S&P in the current Internet are certainly **NOT** a success story

• Retrofitted, incremental, bandaid-style solutions, e.g.:
  • SSH,
  • SSL/TLS (HTTPS),
  • IPSec + IKE + ISAKMP,
  • DNSSec,
  • sBGP,
  • AAA, etc.

NSF Future Internet Architectures (FIA) Program

• Targeted NSF-funded program, 2-tiered competition
• Major goals:
  • Design comprehensive next-generation Internet architectures
  • Accommodate current and emerging comm. paradigms
  • Security and privacy from the outset (by design)
• Started in 2010
  • Phase I: 2010-2014
  • Phase II: 2014-2018
• Projects:
  • Nebula (Phase I)
  • MobilityFirst (Phases I an II)
  • XIA: eXpressive Internet Architecture (Phases I and II)
  • NDN: Named-Data Networking (Phases I and II)
  • ChoiceNet (started in 2012, not strictly speaking FIA)
Caveat Auditor!

- I was part of the NDN FIA project 2010-2014
- Work(ed) on S&P in NDN (and CCN)
- Was funded by the NSF (‘till 09/15)
- Thus... take everything with a grain of salt, draw your own conclusions, and explore further

Also:
- I focus on CCN = NDN and CCNx
- There are other ICN efforts, e.g., for mobile nets

CCN = NDN + CCNx
Pointers

• Named data networking project (NDN), http://named-data.org
• Content-centric networking (CCNx) project, http://www.ccnx.org
• Intro: “Networking named content”, ACM CoNEXT, 2009

• IEEE Infocom NOMEN Workshop 2012, 2013
• Very active IRTF ICN Research Group (ICNRG)
  https://trac.ietf.org/trac/irtf/wiki/icnrg
  https://irtf.org/icnrg
• Dagstuhl Seminars on:
  – General ICN (3 total)
  – ICN Security & Privacy (2 total), latest:
    http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=16251

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
### DN vs. CN

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

Today’s Internet: a communication network, used as a distribution network.

### NDN & CCNx

✧ Both are instances of ICN
✧ Together referred to as “CCN”

NDN/CCNx focus on: **Scalable Content Distribution** which is poorly served by today’s Internet.
Who is/was NDN?

Rough History of CCN/NDN

- CCN birth at PARC
- NDN/CCNx split
- PARC sells CCNx to CISCO
- Convergence effort starts
- Ca. 2009
- Ca. 2013
- Ca. mid-2016
- NDN (NSF/UCLA)
- CCNx (PARC)
What is CCN good for? i.e., what is its “claim to fame”?

Content Distribution over IP
Content Distribution over CCN

CCN Basic Concepts

• Name
  - Human-readable, similar to URI
  - Can be considered as a network-layer URL

• 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:**

```
/parc.com/videos/WidgetA.mp4/v <timestamp>/s3
```

**Binary Encoding:**

```
6 8 parc.com 6 videos 11 WidgetA.mp4 7 FD04A... 2 0003
```
How NDN/CCN works
(abbreviated version)

- Carries content name
- No source/destination address

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

Inside a Router:

<table>
<thead>
<tr>
<th>Interest</th>
<th>Incoming IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ccn/uci/content</td>
<td>IF0, IF3</td>
</tr>
</tbody>
</table>

/ccn/uci/content

Interest: /ccn/uci/content

Interest: /ccn/uci/content

Interest: /ndn/uci/content
Forwarding

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

Routing

- Routing based on name prefixes + reachability, like IP
- Can reuse IP routing protocols, e.g., IS-IS, BGP
Security

- **Now:** secure the pipe
  - Data is authentic because it emanates from the right box (which is an end-point of the right secure pipe)

- **CCN:** Integrity and trust are properties of content
  - Should be inferred from content itself

Is it real? See testbed: http://ndnmap.arl.wustl.edu/
Securing Content: how?

Current SSL/TLS 3-way handshake model is not a good fit for CCN:

- 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)

IPSec is also not a good fit for CCN…

Authenticity of Content

Content requested by a consumer can be retrieved from anywhere

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

CCN Content object:

- **Integrity**: is data intact and complete?
- **Origin**: who produced it?
- **Correctness**: is this (content) what consumer wants (based on interest)?
- **Bonus feature**: routers can choose to verify content (with caveats)

Private Content
(aka Content Access Control)

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

- Encrypt once with a symmetric key
- Distribute this key to authorized consumers using “standard” techniques (pigeons?)
- Access control on key rather than content
  - This can make long-term secrecy problematic

Time permitting, we might come back to this topic…
Trust Model?

- All content is signed
- Interests are not...
- CCN is PKI-agnostic
- Application-specific vs. network-layer trust

CCN: 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
CCN: Privacy Challenges

- Name privacy in interests
  
  /CCN/us/wikipedia/STDs/herpes

- Name privacy in content
  
  /CCN/zimbabwe/piratebay/XSOQW(#E@UED$%.mp3

- Signature privacy
  - Leaks content publisher identity
  - Classical privacy vs. security conflict

- Cache privacy
  - Detectable hits/misses

CCN: 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
  - Consumer first fetches producer’s PKC, then requests content (signed by that producer)
BTW: Keys in CCN

- A producer’s public key is a type of content, i.e., a public key certificate (PKC)
  - Reminder: a consumer doesn’t need a key
- Contains authorized name prefixes under which content can be published
- Binds them to a public key
- For example:
  - `/ccn/cnn/usa/web/key`
  - `/ccn/verisign/europe/key`
  - `/ccn/us/ca/edu/uc/uci/cs/gene.tsudik/key`

CCN: Security Challenges

- State in routers is both a blessing and a curse
- Such state is a resource that can be abused
- DoS attacks:
  - Interest Flooding
  - Content Poisoning: proactive & reactive
- Covert Channels & Geo-location
- Content Access Control
- Trust management at the network layer
CCN: quick recap

**PRODUCER**
- Announces name prefixes
- Names and signs content packets
- Injects content into the network 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)
- Optionally verifies content signatures
  - (1) before forwarding, (2) before caching, or (3) whenever it has time

Some Recent & Ongoing Work on CCN Security/Privacy

- Anonymous content retrieval: ANDaNA/AC3N
- DoS/DDoS:
  - Content poisoning countermeasures
  - Interest flooding mitigation
- Privacy of Router-Side Caching
- Covert channels & Geo-location
- Secure content fragmentation
- NDN security in non-distributive settings (e.g., sensing, actuation)
- Network-Layer Trust Management
- Secure Content Deletion
- Secure Accounting
- Data Privacy
- Network Names
- PIT-less CCN Design
- Secure Content Deletion
- Content Access Control
- NACKs and their Security Implications
Name Privacy and Anonymous Content Retrieval in CCN

Why Name Privacy?

CCN names are expressive and meaningful, but…
- Leak information about requested content
- Easy to filter/censor content, e.g., block everything like:
  
  /CCN/cnn/world-news/russia

However:
- CCN names are opaque to the network
- Routers only need to know name component boundaries – “/”
- Names can carry binary data
ANDaNÁ: Anonymous Named Data Networking Application

- Observers close to consumer should not learn what content is being requested
- Target: low-to-medium-volume interactive communication
- Producers might not be aware of ANDaNÁ
ANDaNA

The New York Times

/OR1

/OR2

nytimes.com/today

?/nytimes.com/today

ANDaNA

The New York Times

/OR1

/OR2
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
Cache Privacy in CCN

CCN Cache Privacy

- Router content 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., on the same LAN/WLAN segment
  - Near producer (opposite sides of first hop router)
  - In both places at once

Scenario 1: Victim=Consumer

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

Recall: consumers must verify content signatures
Therefore, Alice & Bob must first fetch each other’s PK
## 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
- Who should dictate privacy?
  - consumer, producer, router?
- Two communication types:
  - Low-latency (interactive) traffic
    - Use unpredictable content names
  - Content distribution traffic; see paper for details (IEEE ICDCS’13)
    - Random delay
    - Content-specific delay
- Privacy bit in header of interests and/or content?
DoS/DDoS in CCN

DoD/DDoS Resistance?

Some current DoS+DDoS attacks become irrelevant:

• Content caching mitigates targeted DoS

• Content is not forwarded without prior PIT state set up by interest(s)

• Multiple interests for the same content are collapsed

• Only one copy of content per “interested” interface is returned

• Consumer can’t be “hosed” with unsolicited content

>>> THIS IS AN IMPORTANT ADVANTAGE OF CCN!!!
DoS/DDoS

- Attacks on infrastructure
  - Loop-holing/black-holing
  - Interest flooding
  - Router resource exhaustion

- Attacks on consumers & router caches
  - Content flooding
  - Cache pollution
  - Content/cache poisoning

Interest Flooding

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

/CCN/us/ca/uc/uci/cs/gene.tsudik/random-string

Any legitimate producer prefix

- Guaranteed to reach the producer
- Consumes precious router resources (PIT entries)
- IF attack 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

*Open problem:* so far, no deterministic countermeasure!

Content Poisoning

1. Adversary on the path to producer (e.g., a router)
   - Intercepts genuine interest, replies with fake content
   - Content settles in routers

2. Adversary NOT on the path to producer
   - Anticipates demand for content
   - Issues own interest(s), replies with fake content
   - Content settles in routers
Content Poisoning

Potential countermeasures:

- Signature verification in routers?
- Consumer feedback?
- AS egress router verification only?

BTW: what is “fake” content?

- Bad signature (fails verification)
- Bad signing key

Content Poisoning Mitigation

- CCN main objective is content distribution
- Facilitated by caches + PITs in routers
- Consumer must verify content signatures
- But … how to flush fake content from router caches?
- CCN allows exclusion filters in interests (by hash)
  - Can be used, with very limited efficacy
  - Immediate flush \( \Rightarrow \) DoS
  - Verifying signatures \( \Rightarrow \) expensive + another DoS type
- Consumer authentication contradicts interest opacity
Reminder: Public Keys in CCN

- A producer’s public key is a type of content, i.e., a public key certificate (PKC)
  - Reminder: a consumer doesn’t need a key
- Contains authorized name prefixes under which content can be published
- Binds them to a public key
- For example:
  - /ccn/cnn/usa/web/key
  - /ccn/verisign/europe/key
  - /ccn/us/ca/edu/uc/uci/cs/gene.tsudik/key

Content Poisoning

Two reasons:
- Ambiguous interests
- No unified trust model: applications are diverse & dynamic

AXIOM: Network-layer trust and content poisoning are inseparable

Routers should do minimal work:
- Not verify/fetch public keys (except for routing)
- Do bounded, fixed amount of work per content
  - e.g., verify at most one signature
Interest-Key Binding Rule (IKB)

IKB: An interest must reflect the trust context of the consumer’s application, thus making it (easily) enforceable at the network layer

IKB (CCN): An interest must reflect the public key of the content producer

Interest-Key Binding Rule (contd.)

IKB (CCN): An interest must reflect the public key of the content producer

- Make 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:
  o Includes public key in each content’s KeyLocator field

• Router:
  o Matches KeyLocator digest to PPKD in PIT
  o Verifies signature using KeyLocator
  o No fetching, storing, parsing of public keys
    ➔ Note: PIT entry collapsing takes PPKD into account

Is this Secure?

CLAIM:
Adherence to IKB ➔ security against content poisoning

• Assume:
  o All nodes abide by IKB
  o Consumer not malicious
  o Consumer-facing routers – not malicious
  o Consumer ↔ first-hop router link 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 of poisoned content is if a hash collision occurs

What if upstream malicious routers send fake content:
- Consumer-facing router detects and drops it

Optimizations

- Include keys in interest:
  - ✓ Save storage
  - ✗ Requires changes to interest & content structure

- Only AS border routers implement IKB
  - ✓ Better performance
  - ✗ Possible attacks within AS
    - 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) as last component of name

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

- **No** signature verification by routers:
  - Only one hash re-computation

- How to get content hash in the first place?

Catalogs/Manifests and SCN-s

A catalog or manifest:
- An authenticated (signed) data structure
- Contains one or more SCN-s, nesting is arbitrary
- Any authenticated data structure
  - Hash chains, MHTs, skip-lists, etc.
- Structure is application-specific
  - Use IKB to bootstrap (i.e., fetch a catalog)

- SCN obtained from a catalog:
  - **No** addl. signature verification by routers/consumers
  - **No** need for producers to sign individual content
Example: Authenticated Data Structure

CCN Manifest Specification (Internet Draft)
**Two 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.
   - Fore example, Web browsing:
     - HTML file as a catalog
     - Contains SCN of sub-pages/components
     - Works only for static content

**Two types of Traffic (contd.)**

2. **Interactive Traffic**
   - Content generated on demand (real-time), e.g., audio/video conferencing,
   - Catalogs are not viable
   - Content must be requested by setting PPKD in interest
Content NACKs: what if?

- Consumer obtains hash $H$ of content $C$ from $P$’s catalog
- Consumer generates interest for $C$, referring to $H$
- But, $C$ is no longer available at $P$
- $P$ receives interest and ???
  - Just drops it: bad for Consumer
  or:
  - Generates a NACK: routers will drop it since a NACK’s hash doesn’t match $H$

**Bottom-line**: need to augment iKB and interest format to allow for SCN-carrying interests to still refer to $P$’s public key. This can be used as a fallback if SCN enforcement fails.

---

Some Recent & Ongoing Work in CCN Security/Privacy

- Anonymous content retrieval: ANDaNA/AC3N
- DoS/DDoS:
  - Content poisoning countermeasures
  - Interest flooding mitigation
- Privacy of Router-Side Caching
- Covert channels & Geo-location
- Secure content fragmentation
- NDN security in non-distributive settings (e.g., sensing, actuation)
- Network-Layer Trust Management
- Secure Content Deletion
- Secure Accounting
- Data Privacy
- Network Names
- PIT-less CCN Design
- Secure Content Deletion
- Content Access Control
- NACKs and their Security Implications
EoP

http://sprout.ics.uci.edu/projects/ndn/