HopAuth: Hop-by-Hop Authentication in Content-Centric Networking/Named Data Networking

Ruidong Li, Hitoshi Asaeda

National Institute of Information and Communications Technology (NICT)

Outline

• Content-Centric Network
• Adversary Model
• Traditional Approach
• Proposed Hop-by-Hop Authentication Mechanism (HopAuth)
• Conclusions
Content-Centric Network (CCN)

Packet Types: Interest/Data

Publisher: the entity that publishes data in network.
Consumer: the entity that retrieves data from network.
Copyholder: the entity that provides data to network. (Caching Router or Publisher)
Adversary Model

• **A1 (Content Poisoning Attack):** Impersonate a copy holder to provide fake data
  – Currently, the content is only signed with the key of the entity who publishes it.
  – Impossible to verify the publishers’ validity unless all routers use the authentication service of CA for all forwarded/cached data

• **A2 (Interest Flooding Attack):** Impersonate a Consumer to request data
  – Much existing work on restricting the Interest sending rate
  – Impossible to verify the consumer’s validity unless all the Copyholders (Router or Publisher) use the authentication service of CA
Content Poisoning Attack

Fake/corrupted data are cached along the path

Consumers who use the path always retrieve the wrong data, because the router does not detect the cached data validity (as it’s signed by attacker correctly)

Fake data are further cached, which pollute the routers as virus spreads.

Consumer2 and other potential consumers will also retrieve the fake data from routers.

Routers need to verify the data before caching. Consumers verify copyholder and path to identify the polluted entities besides data verifications.
Consumer1 floods the Interests to the network to malfunction routers. Even malicious Interests can be reduced by restricting rate, some malicious Interests still can reach the copyholder. It will enhance the effect to inhibit the Interest flooding attack, if routers verify the Interests before replying the data.
Traditional Authentication Approach

- **CA-based PKI (Certificate Authority-based Public Key Infrastructure)**
  - Routers rely on centralized server to discover the required certificates.
  - Routers need interact with CA to discover and acquire the corresponding certificates using additional messaging for authentication, which leads it to be infeasible during data retrieval.

- **Scalability Problem**
  For router, it is highly cost that each router needs to verify the signature of every data and every Interest.
  For consumer, the public key of all potential copy holders should be provided to users beforehand.
Hop-by-Hop Authentication Mechanism (HopAuth)

Key idea

- HopAuth constructs a certificate chain between copyholder and consumer along the path.
- HopAuth can be CA-independent, which also can partially use CA-based trust to shorten the length of certificate chain and form a suspension-chain based on web-of-trust concept.
- Packet forwarding with collection of certificate chain without extra messages.
- Extend the entity trust graph with the application trust graph.

Effects

Routers cache data only after authentication. Authentication can be performed offline. (Currently caching data without authentication) Publisher authentication, copyholder authentication, consumer authentication, and path authentication can be achieved as required.
Certificate Management

- **Physical entity:** the entity that communicates using a physical device. **Logical entity:** the entity that is involved in an application. (authorizer, sub-authorizer, publisher)

- In HopAuth, certificates are issued based on trust relationships, namely as neighbor-based trust, CA-based trust for physical entities, and authorization relationships in applications for logical entities.

- **Physical Entity Trust**
  - Neighbor-based trust: Routers issue certificates to each other in the neighborhood.
  - CA-based trusts: CA issues certificates to highly trustable physical entities, which form highly trustable router group (HTRG).

- **Logical Entity Trust**
  - Authorization relation-based trust: If logical entity A authorizes the right for entity B to manage a sub-category or publish data, A should provide a certificate for the true public key of B.
Forwarding-Integrated Authenticable Data Retrieval

Part of CEChain can be cached.
Suspension Chain Model (SCM)

**CA:** Certificate Authority  
**PN:** Publisher Node  
**HTRG:** Highly Trustable Router Group  
**pecChain:** Physical Entity Certificate Chain  
**lecChain:** Logical Entity Certificate Chain

Cx: Consumer x  
Ri: Router i  
PN: Publisher Node  
CA: Certificate Authority

→ : Neighbor-based trust relation  
→ - → : CA-based trust relation in HTRG  
→ → : Suspension CA trust relation
Security Levels

• We identify **three security levels** as follows.
  – **Security Level 1 (Low):** Publisher Authentication
    • Authenticate the entity who publishes data.
    • For one piece of data, the suspension chain will be constructed only once.
  – **Security Level 2 (Medium):** Copyholder Authentication
    • Publisher Authentication + Authenticate the router who provides data.
    • For each copyholder during data retrieval, the suspension chain will be constructed once.
  – **Security Level 3 (High):** Path Authentication
    • Copyholder Authentication + Authenticate the path from which data are retrieved.
    • For the first time data chunk retrieval from one path, the suspension chain will be constructed.
    • For the following data chunk retrieval from the same path, a chain of certificate name will be appended.
# HopAuth Packet Format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Version of the packet</td>
</tr>
<tr>
<td>PT_CONTENT</td>
<td>Packet Type</td>
</tr>
<tr>
<td>PacketLength</td>
<td>Length of the packet</td>
</tr>
<tr>
<td>Reserved</td>
<td>Reserved bits</td>
</tr>
<tr>
<td>Flags</td>
<td>Flags for the packet</td>
</tr>
<tr>
<td>HeaderLength</td>
<td>Length of the header</td>
</tr>
<tr>
<td>T_HOPAUTH</td>
<td>Length of T_HOPAUTH TLV</td>
</tr>
<tr>
<td>Total length of inserted T_HOPAUTH_CERT TLVs</td>
<td>Total length of inserted T_HOPAUTH_CERT TLVs</td>
</tr>
<tr>
<td>T_Object</td>
<td>Object Message Length</td>
</tr>
<tr>
<td>Object Message</td>
<td>Object Message contents</td>
</tr>
<tr>
<td>T_VALIDATION_ALG</td>
<td>Validation Algorithm Length</td>
</tr>
<tr>
<td>Validation Algorithm Data</td>
<td>Validation Algorithm Data</td>
</tr>
<tr>
<td>T_VALIDATION_PAYLOAD</td>
<td>Signature Length</td>
</tr>
<tr>
<td>Signature Data</td>
<td>Signature Data</td>
</tr>
<tr>
<td>T_HOPAUTH_CERT</td>
<td>CE1 Length</td>
</tr>
<tr>
<td>CE1(C0-&gt;R1)</td>
<td>CE1(C0-&gt;R1)</td>
</tr>
<tr>
<td>T_HOPAUTH_CERT</td>
<td>CE_n Length</td>
</tr>
<tr>
<td>CE_n(R_(n-1)-&gt;R_n)</td>
<td>CE_n(R_(n-1)-&gt;R_n)</td>
</tr>
</tbody>
</table>
Certificate Size

- DTLS (Datagram Transport Layer Security in Constrained Environments) [REF.1]
  - ECDSA P-256: **91 bytes**
  - ECDSA P-384: **120 bytes**
  - ECDSA P-521: **156 bytes**

For 2K bytes size data chunk, 546 bytes (26.7%) are used to accommodate 6 certificates, if ECDSA P-256 is employed.

<table>
<thead>
<tr>
<th>RSA Public Key Length (bit)</th>
<th>ECDSA Public Key Length (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>160</td>
</tr>
<tr>
<td>2048</td>
<td>224</td>
</tr>
<tr>
<td><strong>3072</strong></td>
<td><strong>256</strong></td>
</tr>
<tr>
<td>7680</td>
<td>384</td>
</tr>
<tr>
<td>15360</td>
<td>512</td>
</tr>
</tbody>
</table>

[REF.1] Datagram Transport Layer Security in Constrained Environments

ECDSA: Elliptic Curve Digital Signature Algorithm
Q1. Can it be used for partially deployed CCN?

• **Answer:** Yes.
Q2. How about using blockchain for authentication?

- Routers still need to interact with Blockchain to query and acquire the certificate chain.

- Blockchain is interesting. We use blockchain to realize data lifecycle management for CCN.

Conclusions

• As an authentication protocol, we propose HopAuth to enable hop-by-hop certificate collection and authentication.

• As a trust model, we propose SCM, which merges web-of-trust and partially CA-based Trust to construct a suspension chain to enable content-centric trust.

• The scientific version of this proposal is published as;

• HopAuth I-D:

• Any comment?
Thank you!