Schematizing Trust in Named Data Networking

Yingdi Yu¹, Alex Afanasyev¹, David Clark², kc claffy³, Van Jacobson¹, Lixia Zhang¹

1. UCLA  2. MIT  3. CAIDA
Motivation

- Usability is critical to security solutions
- Tool to explicitly express trust model
- Mechanism to automate trust management
Data Authentication in NDN

• Data-centric authenticity
  
  • mandate signature on every data packet

• Data authentication needs public key only
  
  • independent from where/how data packet is retrieved
  • privilege of online signing key can be restricted
Trust Model

• Data signing and verification require a trust model
  • one or more pre-trusted keys
  • which key is authorized to sign/verify which data
    • key is just another type of data
    • defines strict authentication path for each data

• Trust model is application specific
  • keys may have different privileges

• Trust may go across different namespaces
NDN Insight

- Trust model can be defined in a set of relationships between data names and key names.

**Trust Schema to Schematize and Generalize Trust Model**
Usable Security

• Need to be easily expressible
  • trust model is application specific
  • given a trust schema, anyone can authenticate data
    • consumers, dedicated storages, routers, …
  • help producers to sign data

• Need to be automated
  • otherwise developers will “temporarily” disable security
    • fake signature, no authentication

• Better to be re-usable
  • applications may share the same trust model
Trust Between Entities

- Blog website framework
  - used by many people to set their own website

- authors can publish articles
- admins can create author account
- blog configuration and admins can designate other admins
Name-based Trust

- Blog framework namespaces

![Diagram showing blog framework namespaces with namespaces for Articles, Authors, and Admins, each containing specific paths for signing.]
Generalize Trust Relationship

• 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
Generalize Trust Relationship

• Relationship between data and key names

\[ /a/blog/article/food/2015/3 \leftrightarrow /a/blog/author/Alice/KEY/22 \]
\[ /a/blog/article/drink/2014/9 \leftrightarrow /a/blog/author/Zach/KEY/5 \]

• Generalize relationship

\[ \text{blog_prefix} + "blog" + "article" + \text{category} + \text{misc_info} \]
\[ \text{blog_prefix} + "blog" + "author" + \text{name} + "KEY" + \text{key_id} \]

• Regex-based syntax

\[ (<>*)<\text{blog}><\text{article}>[\text{category}]<> \]
\[ \backslash 1<\text{blog}><\text{author}>[\text{user}]<\text{KEY}>[\text{Id}] \]
Key Name Pattern Derivation

Data Name

(article)<(>*)(<blog><article>[category]<><><>

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

<Key Name Pattern Derivation>

(article)(<>*)(<blog><article>[category]<><><>

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

Content (article)

Signature

/signatures

/signatures

/signatures

/signatures

/signatures
Enforce Least Privilege

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Key Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>article</td>
<td>author</td>
</tr>
<tr>
<td>author</td>
<td></td>
</tr>
</tbody>
</table>

(article)<(>*)<blog><article>[category]<>><

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

/aws/blog/admin/Carl/KEY/37
/aws/blog/admin/Bob/KEY/5
/aws/blog/author/Eve/KEY/11
/aws/blog/author/Alice/KEY/22
/aws/blog/article/food/2015/1
/aws/blog/author/Alice/KEY/22
/aws/blog/author/Eve/KEY/11
/aws/blog/author/Alice/KEY/22

Link Trust Relationship

Data Name

<table>
<thead>
<tr>
<th>article</th>
<th>(&lt;&gt;*)&lt;blog&gt;&lt;article&gt;[category]&lt;&gt;&lt;&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td>(&lt;&gt;*)&lt;blog&gt;&lt;author&gt;[user]&lt;KEY&gt;[id]</td>
</tr>
<tr>
<td>admin</td>
<td>(&lt;&gt;*)&lt;blog&gt;&lt;admin&gt;[user]&lt;KEY&gt;[id]</td>
</tr>
</tbody>
</table>

Key Name

<table>
<thead>
<tr>
<th>author</th>
<th>author(\1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>admin(\1)</td>
</tr>
</tbody>
</table>

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

Content (public key) 

Signature

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

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

Articles

| /\a/blog/article/food/2015/1 |

Authors

| /\a/blog/author/Alice/KEY/22 |
| /\a/blog/admin/Carl/KEY/37 |

Admins

| /\a/blog/admin/Bob/KEY/5 |
## Multiple Trusted Signers

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Key Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>article</td>
<td>author(1)</td>
</tr>
<tr>
<td>author</td>
<td>admin(1)</td>
</tr>
<tr>
<td>admin</td>
<td>admin(1)</td>
</tr>
</tbody>
</table>

```
<a>blog</a><article>
[category]<<<
```

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

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

---

### Content (public key)

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

### Signature

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

---

```
Content (public key)
```

```
Signature
```

---

```
/a/blog/article/food/2015/1
```

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

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

---

```
Articles
```

```
Authors
```

```
Admins
```

---

```
signs
```

---

```
signs
```

---

```
signs
```

---

```
signs
```

---

```
signs
```

---

```
signs
```

---

```
signs
```

---

```
signs
```

---

```
signs
```

---

```
signs
```
Link Trust Anchor

Data Name

article
(<>*)<blog><article>[category]<>>

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

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

Key Name

author(\1)

admin(\1)
## Trust Schema

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Key Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>article</td>
<td>author</td>
</tr>
<tr>
<td></td>
<td>admin</td>
</tr>
<tr>
<td>author</td>
<td>admin</td>
</tr>
<tr>
<td>admin</td>
<td>root</td>
</tr>
</tbody>
</table>

### Key Name

- **article**: author\(\{1\}\)
- **author**: admin\(\{1\}\)
- **admin**: admin\(\{1\}\)
- **root**: root\(\{1\}\)

### Key

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

Different trust anchor for different blog website.
## Re-usability

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Key Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>article</td>
<td>(&lt;&gt;*)&lt;blog&gt;&lt;article&gt;[category]&lt;&gt;&gt;&lt;&gt; author(\1)</td>
</tr>
<tr>
<td>author</td>
<td>(&lt;&gt;*)&lt;blog&gt;&lt;author&gt;[user]&lt;KEY&gt;[id] admin(\1)</td>
</tr>
<tr>
<td>admin</td>
<td>(&lt;&gt;*)&lt;blog&gt;&lt;admin&gt;[user]&lt;KEY&gt;[id] admin(\1) root(\1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>(&lt;&gt;*)&lt;blog&gt;&lt;KEY&gt;[id] /another/blog/KEY/1 (0x43 0x5a ...)</td>
</tr>
</tbody>
</table>

> /another/blog/article/drink/2014/3
> /another/blog/author/Jason/KEY/25
> /another/blog/admin/Mark/KEY/2
> /another/blog/admin/Karl/KEY/73
Automation

- Trust schema $\rightarrow$ FSM
Automated Signing

1. Derive key name for the article
2. Lookup key in TPM
3. Derive key name for author’s key
4. Lookup key in TPM
5. Expand author’s key name and generate key
6. Sign data

/a/blog/article/snacks/2015/3

/a/blog/article/snacks/2015/3

/a/blog/author/Alex/KEY/40

/a/blog/admin/Alex/KEY/5
Trust schema is more than that …

• Universal tool for trust management

• Representable in a data packet
  • can be retrieved and executed by any NDN entity
    • end application, dedicated storage, routers, …
  • can be (recursively) authenticated using higher-level schemas

• Security design pattern
  • regulate the behavior of applications
  • a set of common trust models
  • application developer can simply select a pre-defined trust model
Implementation

• Available in all the NDN platform libraries
  • 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

• Powers data and interest authentication in:
  • NFD: NDN Forwarding
  • NLSR: NDN Link State Routing Protocol
  • NDNS: NDN Domain Name System
  • Repo-ng: NDN Data Repository
  • ChronoChat: server-less multi-party chat application over NDN
Conclusion

• Usability is critical to all security solutions

• A useful step forward in automating NDN data signing/authentication
  • explicitly defines trust relations between namespaces
  • identify common security patterns to generalize solutions

• Trust schema can be authenticated and fetched as any other NDN data packets

• Potentially applicable to other configuration/automation challenges