A Framework for Secure End-to-End Delivery of Messages in Publish/Subscribe Systems

Shrideep Pallickara, Marlon Pierce, Harshawardhan Gadgil, Geoffrey Fox, Yan Yan, Yi Huang

spallick@indiana.edu
Community Grids Lab, Indiana University
Motivation

As applications have gotten increasingly distributed there is a need for ensuring the secure and authorized distribution of data.
Messaging Systems

- Messaging is the routing of content from the producer to the consumer.
  - This can be point-to-point or many-to-many.

- Messaging Infrastructures underlie most complex systems

- Approaches to messaging include systems such as queuing, P2P systems and publish/subscribe.
Publish/Subscribe Systems

Middleware

CONSUMER

PRODUCER
# Messages & Selectivity

<table>
<thead>
<tr>
<th>Headers</th>
<th>Properties {User/System}</th>
<th>Content Descriptors</th>
<th>Payload</th>
</tr>
</thead>
</table>

## Content Descriptors

#### Topics
- STRING
- `<tag=value>`
- Descriptive Text
- Properties
- XML Document

#### Constraints specified on the Content Descriptors
- STRING
- Tag=* wildcards
- Regular Expressions
- SQL Queries
- XPath or XQuery

### Subscriptions
- Descriptive Text
- Regular Expressions
- SQL Queries
- XPath or XQuery
Topic Discovery Scheme

- Create topics that are unique in space and time in a decentralized fashion
- Establish topic provenance
  - Deterministic cryptographic verification of ownership
- Restrict discovery of topics to only those that possess valid credentials or within a defined set
- Establish topic life-cycle
- Manage topic collections & organization
Security Scheme: Desiderata

- Thwart eavesdropping
- Tamper-evidence
- Authorized data generation/consumption
  - Specify allowed actions
  - Duration of rights
- Identity assertion & Non-repudiation
- Transport-independent
- Cope with attack scenarios
Leveraged cryptographic tools

- Symmetric keys for payload encryptions and decryptions
- Message digests for tamper evidence
- PKI for signing and verifications
- For secure “dialogue” between two entities use combination of symmetric and asymmetric keys
Security Scheme: Components

NaradaBrokering Broker Cloud

Certificate Authority

Key Management Center. (One or more)

Topic Discovery Node. (One or more)

Clients/Entities – Publishers or subscribers.
Communications/Interactions

- All interactions are through the exchange of discrete messages
  - Need to know the communication topic

- Entities are selective about who can discover its communication topic(s)

- Topic owner needs to first discover the KMC willing to host the secure topic
  - Based on credentials supplied during discovery
  - Willing KMCs will respond with their communication topics (secured) in the responses
Entity-KMC Interactions

<table>
<thead>
<tr>
<th>Topic Advertisement</th>
<th>Topic Owner Credentials</th>
<th>Symmetric Algorithm</th>
<th>Key Size</th>
<th>Padding Scheme</th>
<th>ACL, Rights and Duration</th>
</tr>
</thead>
</table>

- **Register** Secure-Topic
- **Verify Ownership**
- **Process Request**
- **Generate Secret Key**
- **Verify Permissions**
- **Generate Security Token**
- **Retrieve Secret Key & Crypto Params**
- **Retrieve Key & Token**

**Key Management Center**

**Publisher/Subscriber**

**SECURITY TOKEN**

<table>
<thead>
<tr>
<th>Entity Credentials</th>
<th>Entity Rights</th>
<th>Topic Info</th>
<th>Duration</th>
<th>KMC Signature</th>
</tr>
</thead>
</table>

IEEE/ACM GRID 2006  
http://www.naradabrokering.org
Security Scheme: Broker Processing

1. Publisher
   - Secure Message
   - Publisher Signature
   - Security Token
   - Topic Info
   - Encrypted Payload

2. Subscriber
   - Subscription

3. Broker
   - VerifyMessage
   - Verify Security Token
     - KMC Certificate
     - Token Expiration
   - Publish Rights
   - Subscribe Rights
   - Compute Destinations
   - Propagate Subscription
   - Verify Subscriber Authorizations
   - Route to Valid Subscribers

Security Scheme: Broker Processing
Coping with a couple of attacks

- Denial of Service attacks
  - Unauthorized generation of data is not allowed
  - Deluging KMC is difficult
    - Quite hard to “guess” the 128-bit UUID
    - Network location know only to hosting broker

- Replay attacks
  - For every entity maintain information about last timestamp
    - Discard messages published in the past
    - For higher publish rates, maintain combination of NTP timestamps and message numbers
    - No need to keep track of message identifiers
End-to-End Delivery of Messages for different topologies
Cryptographic Profile: 256-bit AES, 7PKCS padding
1024-bit RSA keys and 160-bit SHA-1

1 Broker, 1 Subscriber
1 Broker, 101 Subscribers
3 Brokers, 301 Subscribers
Benchmark Topologies

Topology I: 1 Broker, 1 Subscriber

Topology II: 1 Broker, 101 Subscribers

Topology III: 3 Brokers, 301 Subscribers
<table>
<thead>
<tr>
<th>Operation</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Publisher Costs</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initialization Vector</td>
<td>1.108</td>
<td>0.025</td>
<td>0.003</td>
</tr>
<tr>
<td>Encryption</td>
<td>1.421</td>
<td>0.055</td>
<td>0.005</td>
</tr>
<tr>
<td>Signing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td>15.518</td>
<td>0.126</td>
<td>0.013</td>
</tr>
<tr>
<td>Header</td>
<td>15.238</td>
<td>0.112</td>
<td>0.011</td>
</tr>
<tr>
<td><strong>Broker Costs</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Token and Message Validation</td>
<td>6.989</td>
<td>0.199</td>
<td>0.020</td>
</tr>
<tr>
<td>Replay-attack check</td>
<td>0.031</td>
<td>0.005</td>
<td>0.0</td>
</tr>
<tr>
<td>Subscription validity</td>
<td>0.027</td>
<td>0.004</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>Subscriber Costs</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verify Token + Header</td>
<td>3.74</td>
<td>0.13</td>
<td>0.013</td>
</tr>
<tr>
<td>Verify Payload</td>
<td>1.64</td>
<td>0.032</td>
<td>0.003</td>
</tr>
<tr>
<td>Decryption</td>
<td>1.41</td>
<td>0.021</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Encryption \{AES 256, PKCS7 padding CBC mode\}
Signing \{1024-bit RSA, 160-bit SHA-1\}
Conclusions

- Topic provenance lays the groundwork for the security framework.
- Since the scheme is transport independent, it is applicable for systems that can’t use SSL.
  - E.g. Audio/Video conferencing systems.
- Overheads introduced by the security scheme relate to cryptographic operations.
  - No significant increase in message size.
  - Jitter introduced by scheme is quite low.
- Since the nature of processing is determined by the contents in autonomous messages the system can enforce secure and best-effort schemes equally well.
Future Work

- Detecting security compromises
  - Issue authentication challenges at regular intervals
  - Issue queries from a previously negotiated set of queries/responses during initialization
  - Shorter key lifetimes

- In case of a compromise
  - Compute new keys
  - Propagate compromise info to relevant nodes within the system
Related Work

- GKMP – For Multicast
- Groove – Secure shared spaces
- GSI
- WS-Security