Provenance-aware Secure Networks

Wenchao Zhou    Eric Cronin    Boon Thau Loo

University of Pennsylvania
Motivation

- **Network accountability**
  - Real-time monitoring and anomaly detection
  - Identifying and tracing malicious attackers
  - Enforcing trust management policies
  - Problem: Narrowly target specific security challenge/application

- **Provenance (or lineage)**
  - *Data provenance*: explain why a tuple is in a database.
  - *Well-studied query languages and systems in database community.*
  - *Network provenance*: explain why network state/event exists.

**Insight:** network accountability = distributed network provenance
Key Contributions

- Connecting provenance computation to network accountability
  - Usage scenarios for network accountability
  - Taxonomy of data provenance, relation to use scenarios
- Unified platform for provenance-aware secure networks
  - *Declarative networks* [SIGMOD ’06] for protocol specification and implementation
  - Extensions for *security policies* [NetDB ’07]
  - Distributed query-processing techniques for run-time provenance computation
- Techniques to optimize network provenance computation
  - Proactive vs. reactive fashion
  - Sampling, provenance granularity
Outline of Talk

- Motivation
- **Network Accountability in Practice**
  - Real-time Diagnostics
  - Forensics
  - Trust Management
- Background: Declarative Networks & Provenance
- Taxonomy of Network Provenance
- Optimizations
- Preliminary Evaluation
- Conclusion & Future Work
Network Accountability in Practice

- Real-time Diagnostics
  - Monitor networks and detect anomaly in network states
    - Distributed DoS, loss of convergence
    - Implementation bugs, malicious routers, router misconfigurations
  - Language/system support for debugging in distributed systems:
    - PIP [NSDI '06], FRIDAY [NSDI '07]

- Forensics
  - Historical data is required to correlate traffic patterns and prevent attacks
  - IP Traceback [SIGCOMM '00], TimeMachine [IMC '05], IP Forensics [ICNP '06]
    - Store the complete path in the packet
    - Maintain state at each router, perform subsequent traceback by a distributed query
Network Accountability in Practice

- Trust Management
  - Enforce trust policies based on *origins* and *intermediaries* ("chain of custody")
  - Real-world examples:
    - Path-vector protocols used in BGP carry the entire path during route advertisement
    - P2P data-sharing networks
  - Further explore *quantifiable* notion of trust:
    - Vote-based protocols (e.g. SPKI/SDSI, logic-based D1LP)
    - Granting an update only if over K principals assert it
Outline of Talk

- Motivation
- Network Accountability in Practice
- **Declarative Networks & Provenance**
- Taxonomy of Network Provenance
- Optimizations
- Preliminary Evaluation
- Conclusion & Future Work
Declarative Networking

- **Declarative framework for networks:**
  - Network Datalog (NDLog) language as the specification
  - Declarative specifications of networks, compiled to distributed dataflows
  - Distributed query engine to execute dataflows to implement protocols

- **Datalog syntax**
  - `<result> :- <condition1>, <condition2>, … , <conditionN>.
  - Types of conditions in body
    - Input tables: link(src,dst) predicate
    - Arithmetic and list operations
  - Head is an output table stored locally.
NDLog Example: Reachability

\[ r1: \text{reachable}(\text{@S,D}) \leftarrow \text{link}(\text{@S,D}) \]
\[ r2: \text{reachable}(\text{@S,D}) \leftarrow \text{link}(\text{@S,Z}), \text{reachable}(\text{@Z,D}) \]

\text{link(@a,b)} – “there is a link from node \text{a} to node \text{b}”

\text{reachable(@a,b)} – “node \text{a} can reach node \text{b}”

If there is a link from \text{S} to \text{D}, then \text{S} can reach \text{D}.

If there is a link from \text{S} to \text{Z}, AND \text{Z} can reach \text{D}, then \text{S} can reach \text{D}.

<table>
<thead>
<tr>
<th>Node a</th>
<th>Node b</th>
</tr>
</thead>
<tbody>
<tr>
<td>link(@a, b)</td>
<td>link(@b, c)</td>
</tr>
<tr>
<td>link(@a, c)</td>
<td>reachable(@b, c)</td>
</tr>
<tr>
<td>reachable(@a, c)</td>
<td></td>
</tr>
</tbody>
</table>
Secure Network Datalog (SeNDLog)

- Secure Network Datalog
  - Combine NDLog and logic-based access control languages.
  - Unified declarative language for specifying networks and security policies.
  - “Says”: abstraction of detailed authentication (e.g. certificate)

- Reachability Example
  - “Says”: abstraction of authentication
  - Context: where operators take place
  - Network provenance: store traversed principals in algebra expressions
  - + means union, * means join

04/07/2008

Wenchao Zhou et.al – NetDB ’08
Outline of Talk

- Motivation
- Network Accountability in Practice
- Declarative Networks & Provenance
- Taxonomy of Data Provenance & Usage Scenarios
  - Local vs. Distributed Provenance
  - Condensed Provenance
- Optimizations
- Preliminary Evaluation
- Conclusion & Future Work
# Taxonomy of Provenance

<table>
<thead>
<tr>
<th>Provenance Taxonomy</th>
<th>Real-time Diagnostics</th>
<th>Forensics</th>
<th>Trust Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local / Distributed</td>
<td>✓</td>
<td>✓</td>
<td>✓ (Local)</td>
</tr>
<tr>
<td>Online / Offline</td>
<td>✓ (Online)</td>
<td>✓ (Offline)</td>
<td>✓ (Online)</td>
</tr>
<tr>
<td>Authenticated</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Condensed</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Quantifiable</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Local vs. Distributed Provenance

- Local provenance
  - The entire provenance is stored with each tuple.
  - E.g. node a scores the entire derivation tree for reachable(@a, c)

- Distributed provenance
  - Pointers to the direct derivations are stored.
  - E.g. maintain pointers to link(@a, b) and reachable(@b, c) for reachable(@a, c).

- Tradeoffs between local and distributed provenance
  - Local provenance: provenance querying is cheap as they are available locally
  - Distributed provenance: no extra communication overhead
Condensed Provenance

- Condense the size of local provenance
  - *Provenance semirings* annotates provenance in Boolean expressions
  - Encode in *Binary Decision Diagrams*

- Condensed + Authenticated:
  - Retain sufficient information for trust management.
  - If a is trusted: derivable from a single principal a; accept.
  - If a is untrusted: derivations all depend on principal a; reject.
  - Principal b is inconsequential
Other Optimizations

- Proactive vs. Reactive Provenance
  - Proactive mode: all provenance are eagerly propagated throughout the network
  - Reactive mode: provenance are triggered only by specified network events

- Sampling
  - Record only a portion of the provenance
  - E.g. IP Traceback records messages 1/20,000th of the time

- Provenance Granularity
  - Aggregate and maintain provenance at different granularities
  - A balancing choice between accuracy and performance
Outline of Talk

- Motivation
- Declarative Networks & Provenance
- Network Provenance in Practice
- Taxonomy of Network Provenance
- Optimizations
- Preliminary Evaluation
- Conclusion & Future Work
Preliminary Evaluation

- **P2** declarative networking system with security extensions and provenance support
- On a quad-core machine, running multiple **P2** nodes on different ports
- **Path-vector** query as the workload to compute shortest paths between all pairs of nodes.
- Measure CPU and bandwidth overhead, affordable for provenance and authentication computations.
Conclusion & Future Work

- **Conclusion**
  - Connection between provenance computation to network accountability
  - Unified declarative networks with security and provenance extensions
  - Optimizations and preliminary feasibility evaluation.

- **Future work**
  - Validate our system with a variety of secure networks (e.g. secure Chord)
  - Explore other practical aspects of our system
    - Query optimizations
    - Security vs. performance balancing
  - Extensible security typing systems, evidence-based auditing
  - Incorporate probabilistic database to the quantifiable notion of trust
Thank You ...