Unified Platform for Secure Networked Information System

Wenchao Zhou[1], Yun Mao[2], Boon Thau Loo[1], Martín Abadi[3]

[1]University of Pennsylvania, [2]AT&T Research, [3]Microsoft Research

This work was partially supported by NSF grants CNS-0721845 and IIS-0812270.

Motivation

- **Proliferation of new network architecture and protocols**
  - Overlay networks with new capabilities
    - Mobility, resiliency, anycast, multicast, anonymity, etc
  - Distributed data management applications
    - Network monitoring, publish-subscribe systems, content-distribution networks
- **Challenges - scalability and security threats**
- **Techniques proposed by security/networking community**
  - Distributed debugging: PIP [NSDI 06], FRIDAY [NSDI 07]
  - Forensics: IP traceback [SIGCOMM 00], IP forensics [ICNP 06]
  - Network accountability: PeerReview [SOSP 07], AIP [SIGCOMM 08]
  - Trust management: SD3 [Oakland 01], Delegation Logic [TISSEC 03]

Motivation

- **Problem: lacking generalized framework**
  - Designed for specific security threats
  - Implemented and enforced in different languages and environments
  - Lack of cross-layer integration with existing distributed query processors

A unified platform – network protocol specification, security policy, support for a variety of techniques for secure networks

Contributions

- **A unified declarative language:**
  - Declarative networking: network protocol specifications
  - Access control languages: logic for security policies
  - Securing network routing (S-BGP), DHTs, p2p query processing
- **Authenticated distributed query processing**
  - Extension of existing database techniques for *authenticated* communication
  - Implementation in a declarative networking engine
- ***Network provenance***
  - Data provenance: explain the existence of a tuple in database
  - Relate to real-world use cases in secure networked information systems
- **Experimental evaluation on a local cluster and Planetlab testbed**

Outline of Talk

- Introduction
- Unified Declarative Framework
  - Background: Declarative Networking and Access Control Languages
  - Secure Network Datalog (SeNDlog)
- Authenticated Distributed Query Processing
- Network Provenance
- Experimental Evaluation
- Conclusion & Future Work

Background: Declarative Networking

- **Declarative query language for network protocols**
  - Network Datalog (NDlog) – distributed Datalog [SIGCOMM 05, SIGMOD 06]
  - Compiled to distributed dataflows, executed by distributed query engine
  - *Location specifiers* (@ symbol) indicate the source/destination of messages

- **Example: Network Reachability**

r1: reachable(@S,D) :- link(@S,D)

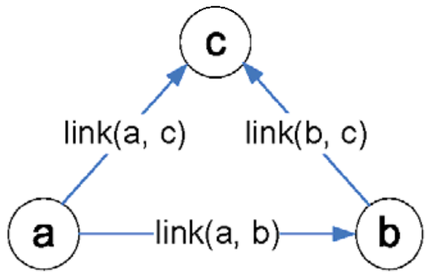
r2: reachable(@S,D) :- link(@S,Z), reachable(@Z,D)

link(@a,b) – "there is a link from node *a* to node *b*"

reachable(@a,b) – "node *a* can reach node *b*"

If there is a link from S to D, then S can reach D.

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



Node a	Node b
link(@a, b)	link(@b, c)
link(@a, c)	reachable(@b, c)
reachable(@a, c)	

Background: Access Control Languages

- **Access control is broadly defined as:**
  - Enforce security policies in a multi-user environment
  - Assigning credentials to principals to perform actions
- **Declarative interface:**
  - Analyzing and implementing security policies
  - Several runtime systems based on distributed Datalog/Prolog
- **Binder** [Oakland 02]**: a simple representative language**
  - **Context:** each principal has its own context where its rules and data reside
  - **Authentication:** “says” construct (credentials, signatures)

    At alice:

    ```
    b1: access(P,O,read) :- good(P).
    b2: access(P,O,read) :- bob says access(P,O,read).
    ```
  - “In alice's context, any principal P may access object O in read mode if P is good (b1) or, bob says P may do so (b2 - delegation)”

Secure Network Datalog (SeNDlog)

- Rules within a context
  - Untrusted network
  - Predicates in rule body in local context
- Authenticated communication
  - "says" construct
  - *Import predicate:* "X says p"
    - X asserts the predicate p.
  - *Export predicate:* "X says p@Y"
    - X exports the predicate p to Y.

```
r1: reachable(@S,D) :- link(@S,D).
r2: reachable(@Z,D) :- link(@S,Z),
                       reachable(@Z,D).
```

⇩ *localization rewrite*

At S:

```
s1: reachable(@S,D) :- link(@S,D).
s2: linkD(D,S)@D :- link(S,D).
s3: reachable(Z,D)@Z :- linkD(@S,Z),
                        reachable(@S,D).
```

⇩ *authenticated communication*

At S:

```
s1: reachable(@S,D) :- link(@S,D).
s2: S says linkD(D,S)@D :- link(S,D).
s3: S says reachable(Z,D)@Z :-
            Z says linkD(@S,Z),
            W says reachable(@S,D).
```

Example Protocols in SeNDlog

- **Secure network routing**
  - Nodes import/export signed route advertisements from neighbors
  - Advertisements include signed sub-paths (*authenticated provenance*)
  - Building blocks for secure BGP
- **Distributed hash table overlay**
  - Chord DHT – authenticate the node-join process
  - Signed node identifiers to prevent malicious nodes from joining the DHT
- **P2P query processing – application layer**
  - PIER - built upon Chord DHT
  - Capability of *layered authentication*

Outline of Talk

- Introduction
- Unified Declarative Framework
- Authenticated Distributed Query Processing
  - Authenticated Pipeline Semi-Naïve
  - Dataflow Architecture
- Network Provenance
- Experimental Evaluation
- Conclusion & Future Work

Authenticated Query Processing

- **Semi-naïve Evaluation**
  - Standard technique for processing recursive queries
  - Synchronous rounds of computation
- **Pipelined Semi-naïve Evaluation** [SIGMOD 06]
  - Asynchronous communication in distributed setting
  - No requirement on expensive synchronous computation
- **Authenticated Semi-naïve Evaluation**
  - Modification for “says” construct, in p’s context:

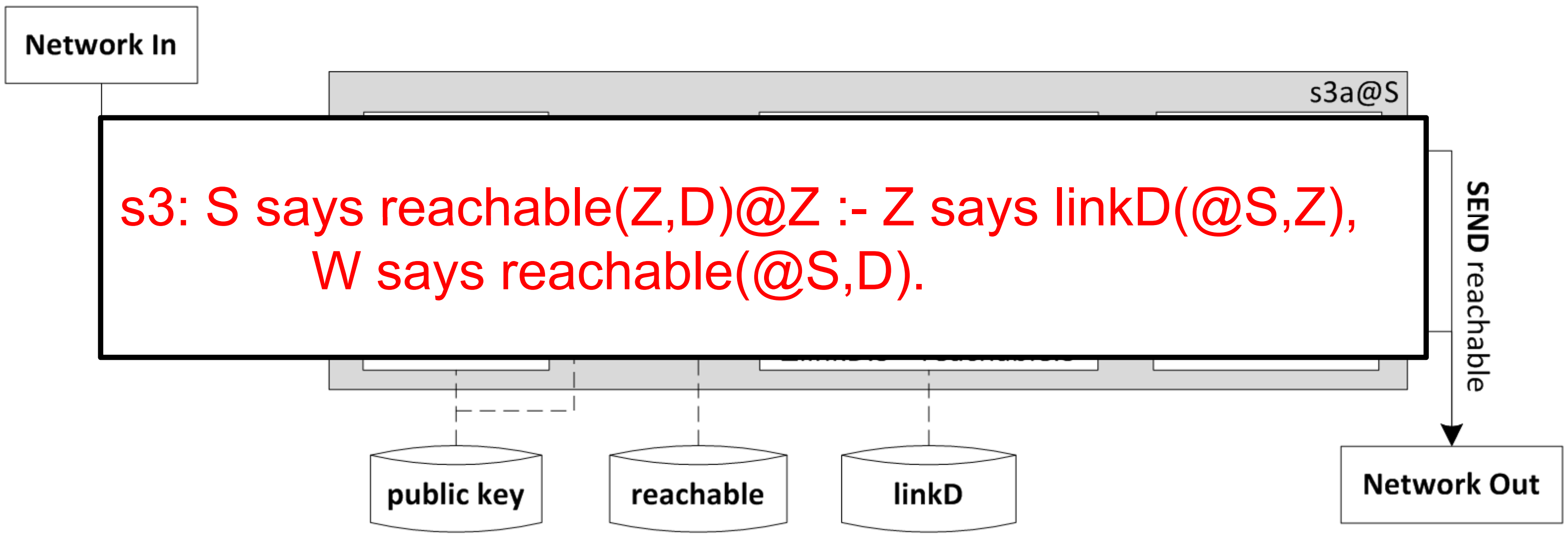
    a :- d_1, ..., d_n, b_1, ..., b_m, p_1 says a_1, p_2 says a_2, ..., p_o says a_o.

    for kth *import predicate*, an authenticated delta rules is generated:

    p says Δa :- d_1, ..., d_n, b_1, ..., b_m, p_1 says a_1, ..., p_k says Δa_k, ..., p_o says a_o.

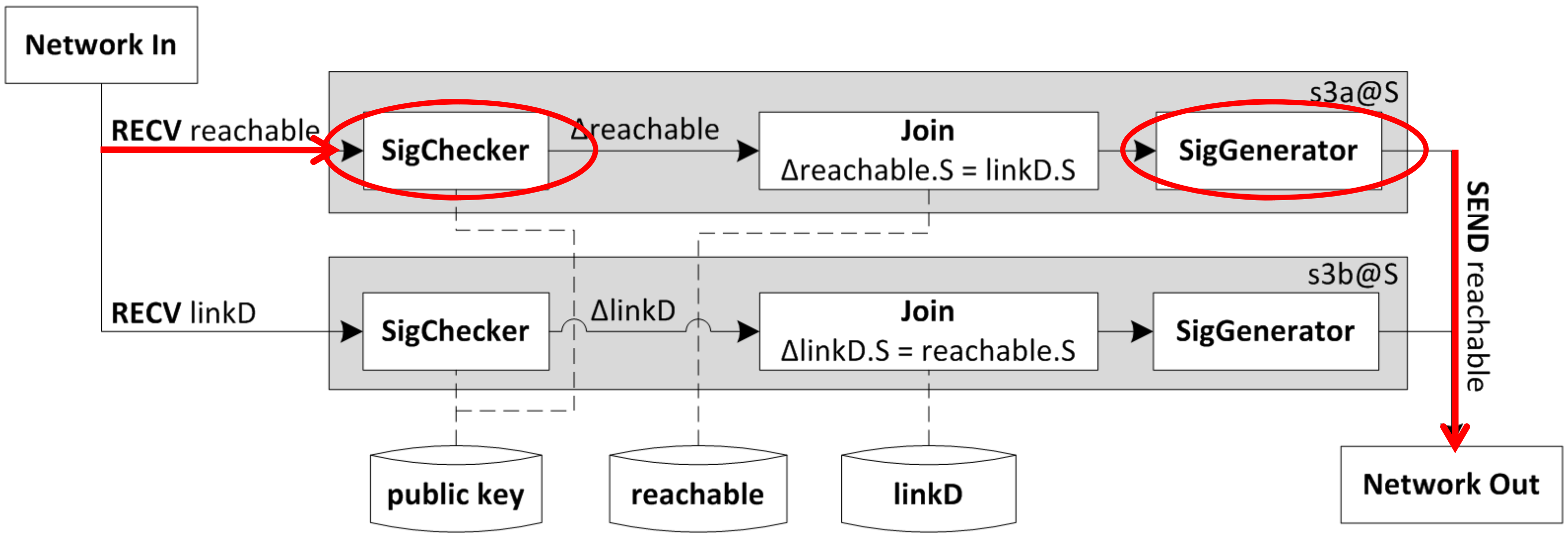
Architectural Overview of Dataflow

- **Dataflow Architecture**
  - Based on the P2 declarative networking system *[http://p2.cs.berkeley.edu/]*
  - Additional modules to support authenticated communication



Architectural Overview of Dataflow

- **Dataflow Architecture**
  - Based on the P2 declarative networking system *[http://p2.cs.berkeley.edu/]*
  - Additional modules to support authenticated communication

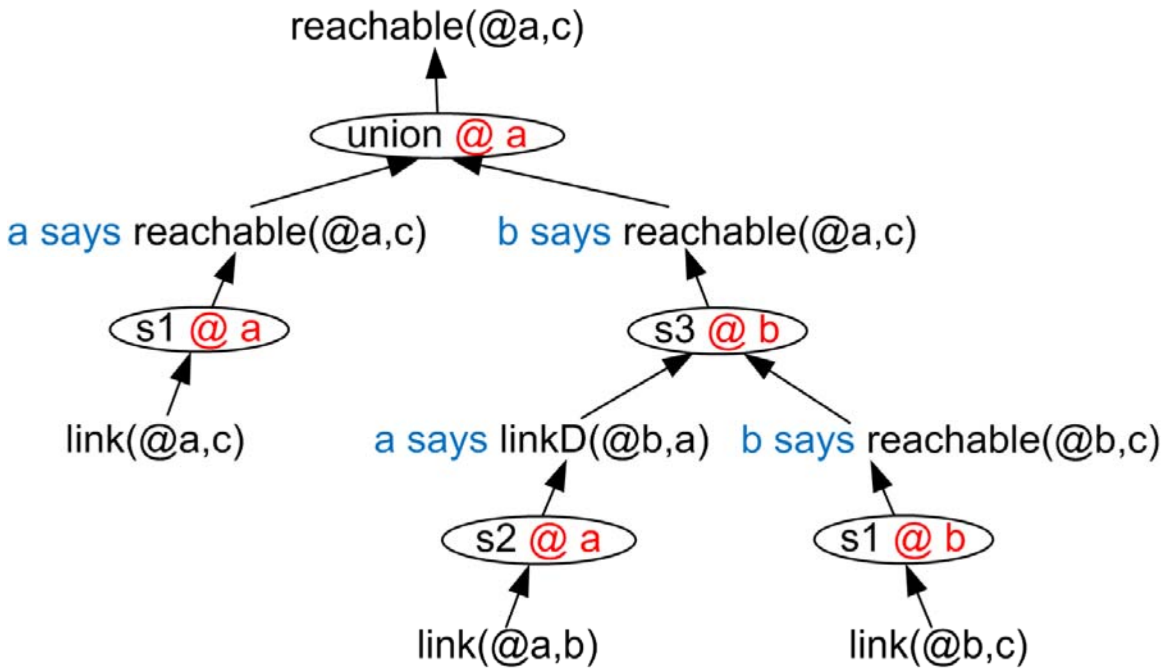


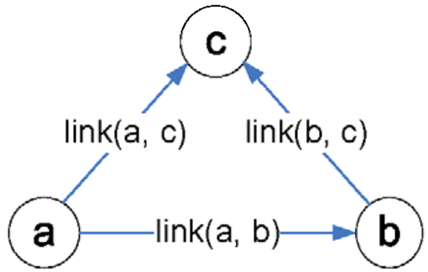
Outline of Talk

- Introduction
- Unified Declarative Framework
- Authenticated Distributed Query Processing
- Network Provenance
  - Network Provenance
  - Wide Application of Network Provenance
- Experimental Evaluation
- Conclusion & Future Work

Network Provenance

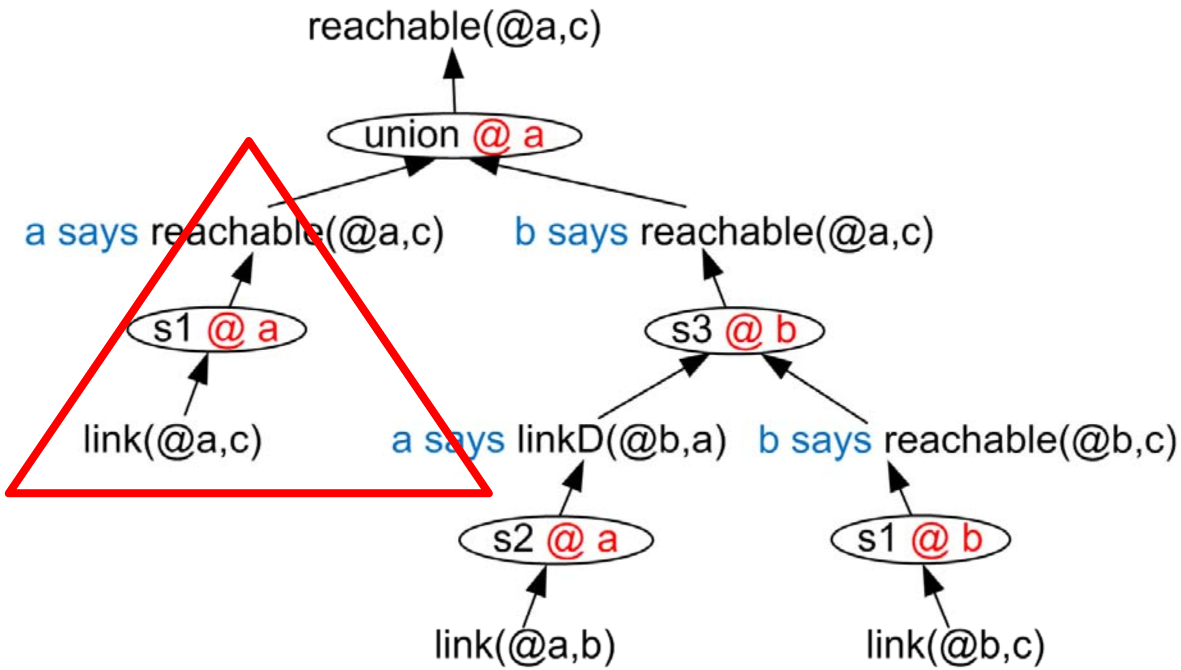
- Naturally captured within declarative framework
- Explain the existence of any network state
- Similar notion in security community: *proof-trees*

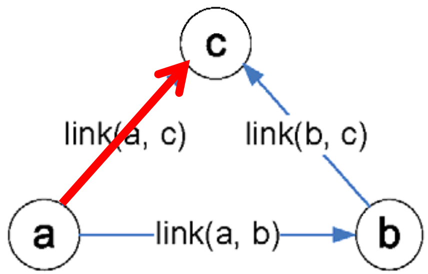




Network Provenance

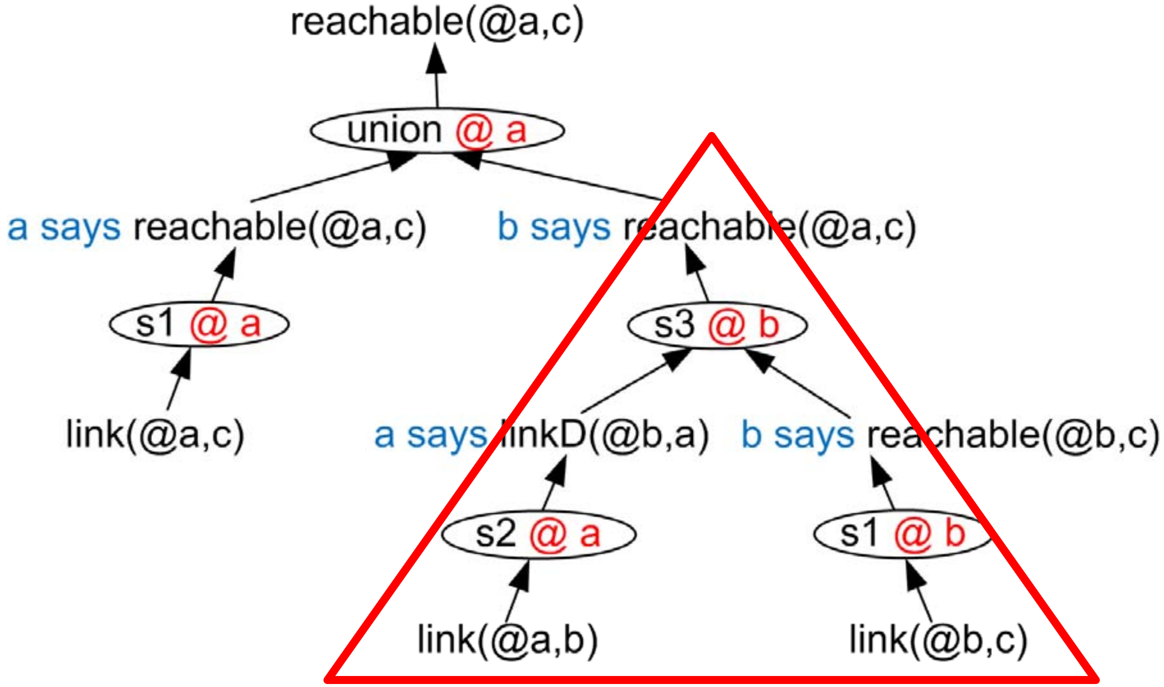
- Naturally captured within declarative framework
- Explain the existence of any network state
- Similar notion in security community: *proof-trees*

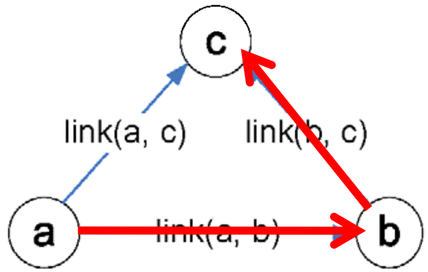




Network Provenance

- Naturally captured within declarative framework
- Explain the existence of any network state
- Similar notion in security community: *proof-trees*





Wide Application of Network Provenance

Provenance Taxonomy	Distributed Debugging	Forensics	Network Accountability	Trust Management
Derivation Tree / Algebra Expr.	Both	Derivation Tree	Both	Algebra Expr.
Local / Distributed	Both	Both	Both	Local
Online / Offline	Online	Offline	Offline	Online
Boolean/ Quantifiable	Both	Boolean	Boolean	Both

- Distributed debugging: PIP [NSDI 06], FRIDAY [NSDI 07]
- Forensics: IP traceback [SIGCOMM 00], IP forensics [ICNP 06]
- Network accountability: PeerReview [SOSP 07], AIP [SIGCOMM 08]
- Trust management: SD3 [Oakland 01], Delegation Logic [TISSEC 03]

Outline of Talk

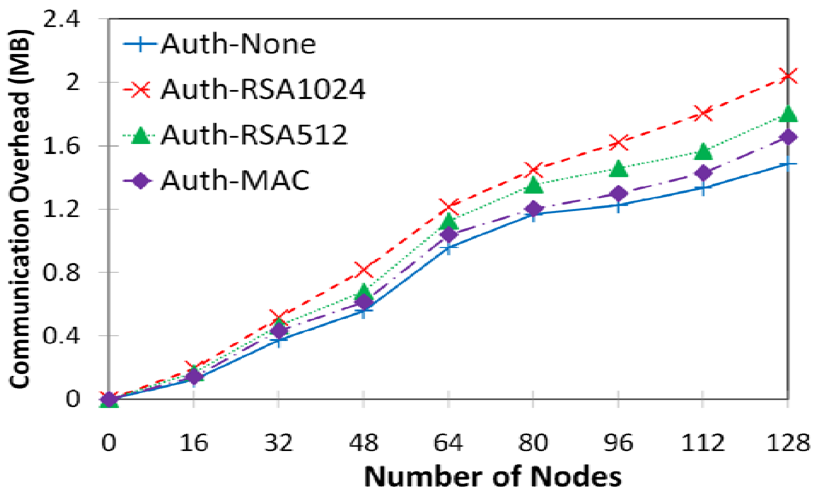
- Introduction
- Unified Declarative Framework
- Authenticated Distributed Query Processing
- Network Provenance
- Experimental Evaluation
- Conclusion & Future Work

Experimental Setup

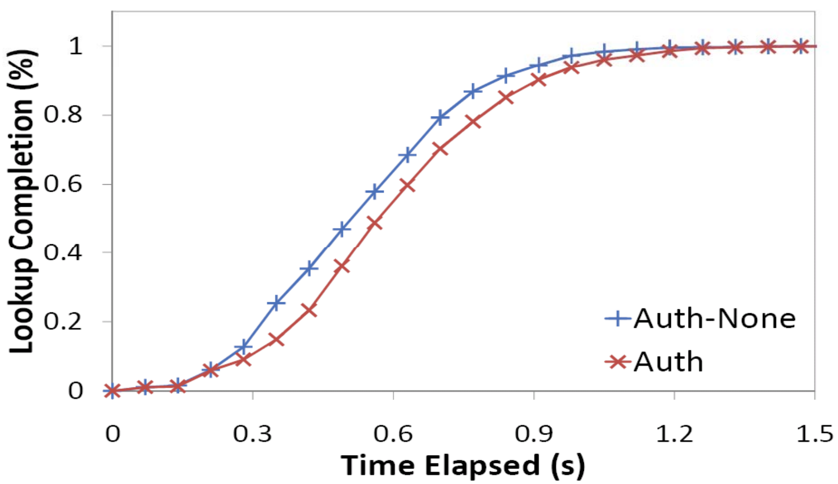
- **P2 declarative networking system**
  - Extensions for security and provenance support
- **Workload**
  - Path-vector – network routing
  - Chord – distributed hash table
  - PIER – p2p query processing
- **Test-bed**
  - A local cluster with 16 quad-core machines
  - Planetlab testbed with 80 nodes
- **Metrics**
  - Communication overhead
  - Query completion time / lookup latency

Feasibility Study of SeNDlog

Path-vector on cluster



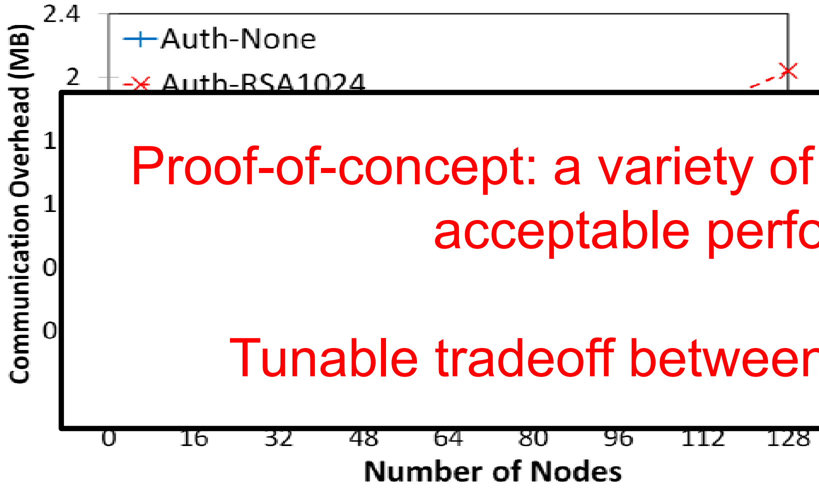
Lookup latency for Chord on cluster



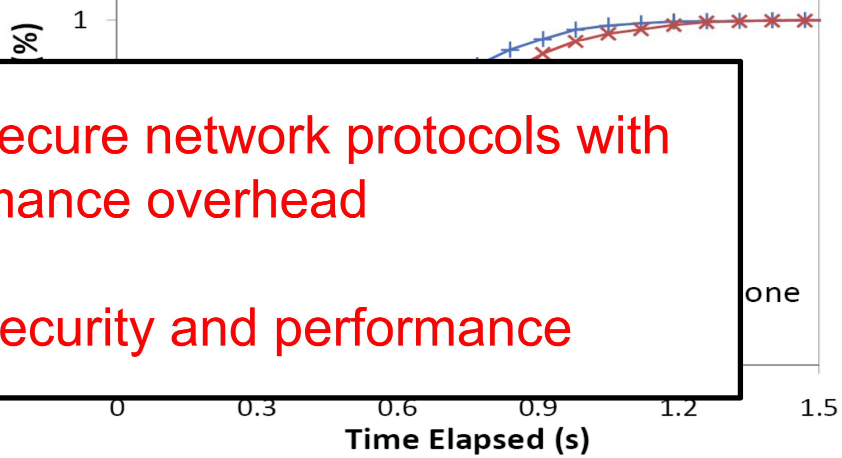
- Path-vector protocol
  - 128 nodes, 6 neighbors per node
  - Auth-HMAC – 10% increase
  - Auth-RSA512 – 20% increase
  - Auth-RSA1024 – 40% increase
- Chord DHT protocol
  - 128 Chord nodes, random lookups
  - Auth (with RSA1024) – less than 10% increase to finish 50% lookups

Feasibility Study of SeNDlog

Path-vector on cluster



Lookup latency for Chord on cluster



Proof-of-concept: a variety of secure network protocols with acceptable performance overhead

Tunable tradeoff between security and performance

- Path-vector protocol
  - 128 nodes, 6 neighbors per node
  - Auth-HMAC – 10% increase
  - Auth-RSA512 – 20% increase
  - Auth-RSA1024 – 40% increase
- Chord DHT protocol
  - 128 Chord nodes, random lookups
  - Auth (with RSA1024) – less than 10% increase to finish 50% lookups

Other Evaluation Results

- **Planetlab Experiments**
  - WAN effect: high inter-node latency, low availability of computation resources
  - Relative overhead increase is amortized

- **Validate the feasibility of network provenance**
  - Packet delivery: routing tables pre-computed using SeNDLog programs
  - Use local network provenance to trace the route taken by a packet
  - Acceptable performance: 0.05s increase in packet delivery latency in LAN

Conclusion & Future Work

- **Conclusion**
  - SeNDlog: Unified language for declarative networking and access control
  - Authenticated query processing techniques for distributed settings
  - Support for network provenance

- **Future Work**
  - Possible language extensions
    - Secrecy / encrypted facts
    - Restricted delegation / “speaks-for” primitive
  - Optimizations opportunities
    - Performance / security tradeoff
    - Bandwidth optimization for network provenance

Thank You …