Towards a Data-centric View of Cloud Security

Wenchao Zhou[*], Micah Sherr[#], William R. Marczak[+], Zhuoyao Zhang[*], Tao Tao[*], Boon Thau Loo[*], and Insup Lee[*]

[*]*Univ. of Pennsylvania* [#]*Georgetown Univ.* [+]*Univ. of California, Berkeley*

Introduction

- **Success of cloud**
  - Economics of outsourcing data and computation
  - Continued migration of applications to the cloud
  - Amazon EC2, salesforce, Microsoft Office…
- **Security: one barrier that prevents further success**
  - Enforce the privacy and integrity of user data
  - Current solutions mostly focus on OS and virtual machines
- **Cloud applications are increasingly interdependent**

Motivating Examples

- **Interconnecting enables more adaptable systems**
- **Online market-places**
  - Retail portals such as Yahoo!, Amazon serve as storefronts
  - Collect product and inventory information from sellers
  - Should prevent from …
    - merchants querying each others' inventories and prices
    - communicating payment info with unauthorized parties
- **Social network services, outsourced data storage…**

Overview

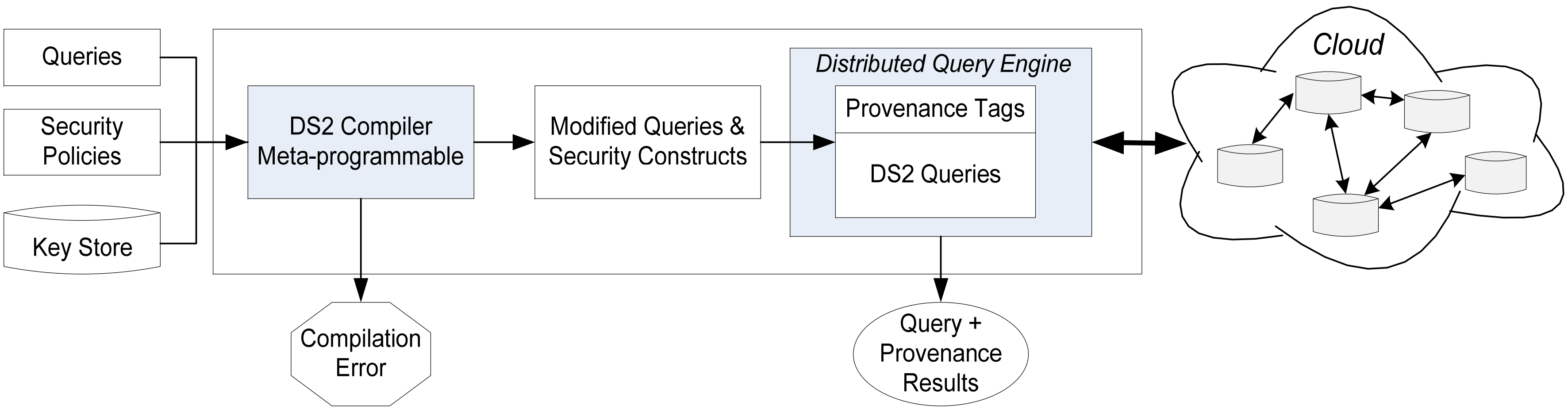
- **A comprehensive solution should …**
  - go beyond OS and VM-centric security solutions
  - securely share, verify, and trace data between applications

- **DS2 (Declarative Secure Distributed Systems)**
  http://netdb.cis.upenn.edu/ds2/
  - Secure querying processing
  - Declarative access control polices for data sharing
  - System analysis and forensics using distributed provenance
  - End-to-end verification of data partitioned across users

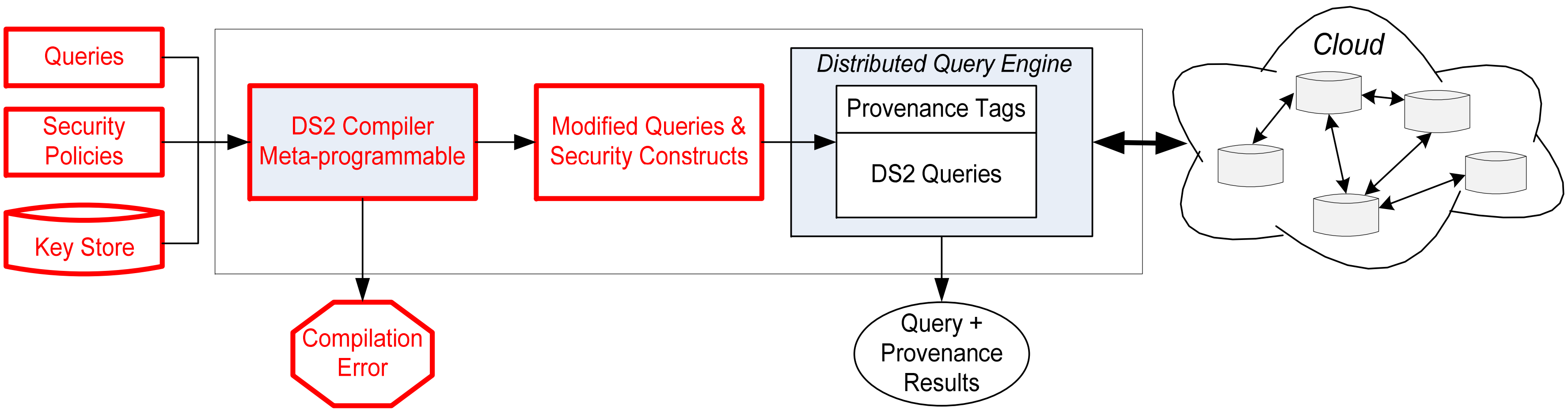
Outline

- Introduction
- Motivation
- DS2 Platform
  - Secure Data Processing
  - Declarative Access Control
  - Distributed Provenance
  - End-to-end query verification
- Conclusion

DS2 Platform

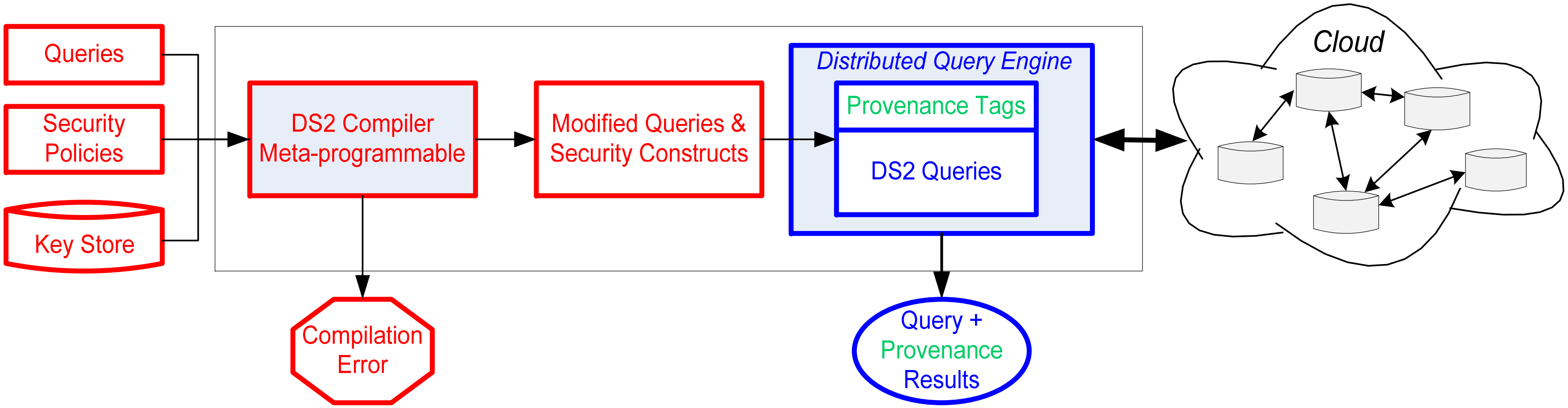


DS2 Platform



- Integration of access control policies
  - Meta-programmability

DS2 Platform



- Integration of access control policies
  - Meta-programmability
- Provenance-aware secure query processing
- End-to-end verification

Secure Query Processing

Zhou et al. **Unified Declarative Platform for Secure Networked Information Systems**, *ICDE09*

- **Compact specification of network protocols**
- ***Secure Network Datalog (SeNDlog)***
  - A distributed variant of *Datalog*
  - Continuous recursive queries over network state
  - Security Primitives
    - Rules within a *context*
    - Authenticated communication
- **A variety of secure distributed systems**
  - Secure network routing (S-BGP), DHTs, p2p query processing

Example: Authenticated Map-Reduce

At MW:

m1 map(ID,Content) :- file(MW,ID,Content).

m2 MW says emits(MW,Word,Num,Offset)@RW :-
word(Word,Num,Offset),
reduceWorker(RID,RW), RID=f_SHA1(Word).

- In the context of Map Worker
  - m1: Perform map operation on each file
  - m2: For each word in the document, pass it to the reducer according to the mapper-reducer mapping.
- Authenticate outgoing tuples by tagging signatures

Example: Authenticated Map-Reduce

At RW:

r1 reduceTuple(Word,a_LIST<Num>) :-
    MW says emits(MW,Word,Num,Offset).

r2 reduce(Word,List) :- reduceTuple(Word,List),
    Master says rBegin(RW).

- In the context of Reduce Worker
  - r1: Group the received words, and maintain a list for each word
  - r2: Perform reduce operation once received signal from Master
- Verify the signatures of the incoming tuples

Example: Authenticated Map-Reduce

At RW:

r1 reduceTuple(Word,a_LIST<Num>) :-

MW says emits(MW,Word,Num,Offset).

Unified platform: protocol specs & security enforcement

Building blocks for more complex security policies

- r1: Group the received words, and maintain a list for which word
- r2: Perform reduce operation once received signal from Master

- Verify the signatures of the incoming tuples

Access Control

Marczak et al. **SecureBlox: Customizable Secure Distributed Data Processing**, *SIGMOD10*

- **View-based Access Control**
  - Horizontal and vertical partition of relational table
  - Authorization + authentication
  - Access ONLY to the secure views
  - ***How can we enforce this?***

At alice:

sv1 sview(Name,Dept) :- employee(Name,Dept,Salary), Salary < 5K.

sv2 predsecview("employee","sview",U) :- authority says good(U).

sv3 ret(Name,Dept)@U :- U says query("***sview***"), sview(Name, Dept).

Access Control

Marczak et al. **SecureBlox: Customizable Secure Distributed Data Processing**, *SIGMOD10*

- **Enforcement: meta-constraints**
  - Meta-model – rules as data
  - Check the query format against schema constraints
  - says(U,R), body(R,A), functor(A,P) -> predsecview(_,P,U)
- **Code Generation**
  - Automatic rewrite of queries to refer to security views
  - Updates in the meta-model
  - Customizable security constructs according to policy changes

Outline

- Introduction
- Motivation
- DS2 Platform
  - Secure Data Processing
  - Declarative Access Control
  - Distributed Provenance
  - End-to-end query verification
- Conclusion

Distributed Provenance

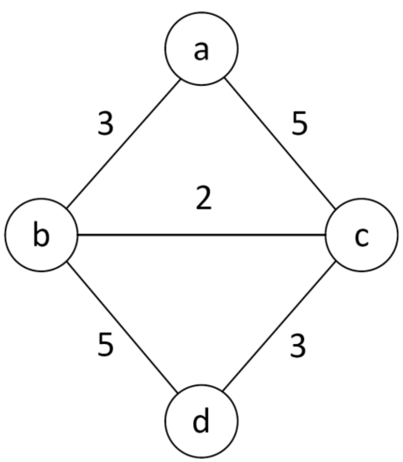
Zhou et al. **Efficient Querying and Maintenance of Network Provenance at Internet-Scale**, *SIGMOD10*

- **Distributed provenance (or lineage)**
  - Explains the existence and derivation of any network state
  - Maps naturally into various applications
- **Applications in cloud**
  - Error detection, diagnosis, and forensics
  - Mitigation: propagating corrections only to affected applications
  - History-based trust management

Distributed Provenance

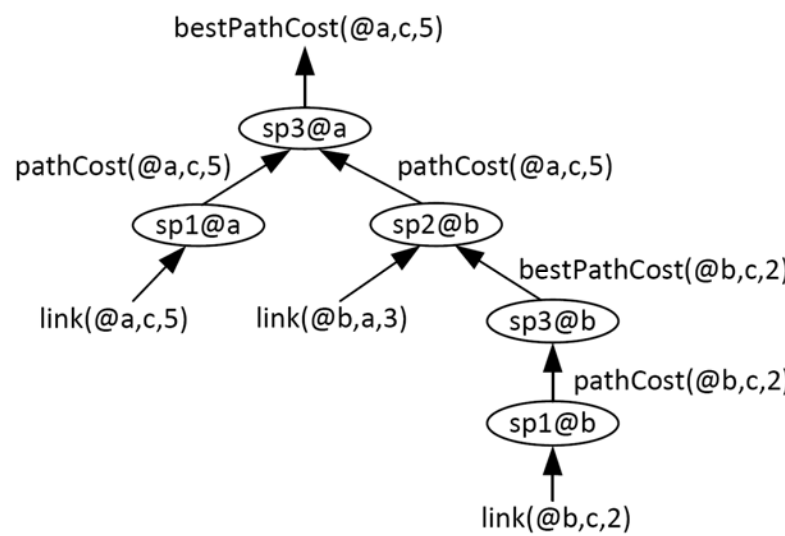
Zhou et al. **Efficient Querying and Maintenance of Network Provenance at Internet-Scale**, *SIGMOD10*

- **Data model – a directed graph**
  - Tuple and rule execution vertices
  - Edges represent dataflows



link (@src, dest, cost)

src	dest	cost
a	b	3
a	c	5
b	a	3
b	c	2
b	d	5
c	a	5
c	b	2
c	d	3
d	b	5
d	c	3



Distributed Provenance

Zhou et al. **Efficient Querying and Maintenance of Network Provenance at Internet-Scale**, *SIGMOD10*

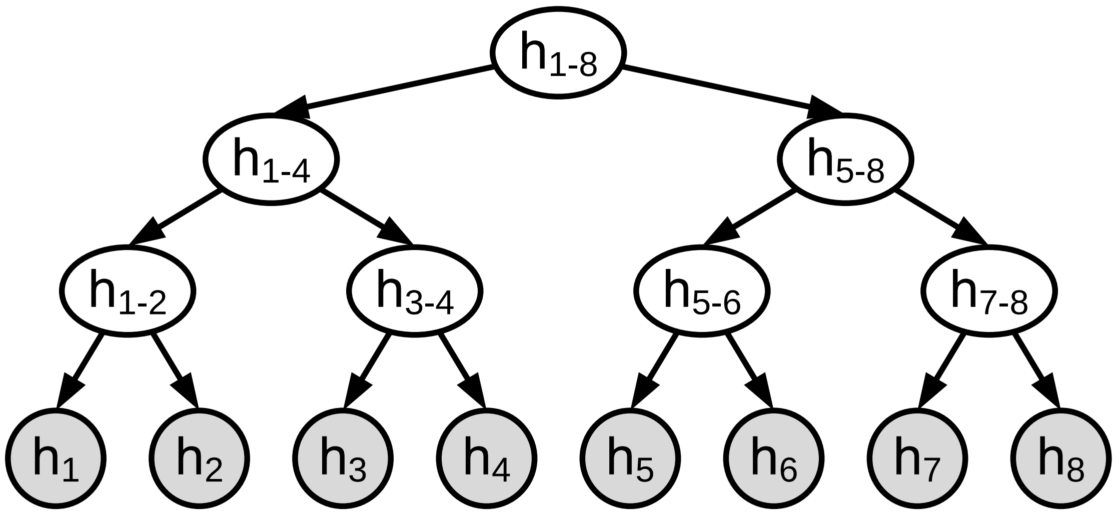
- **Data model – a directed graph**
  - Tuple and rule execution vertices
  - Edges represent dataflows
- **Maintenance and querying**
  - Maintained as distributed relational tables
  - Views of base and derived tuples
  - Querying performed as graph traversal
- **Reasonable overhead for distributed provenance**

End-to-end Query Verification

- **Threat Model**
  - The owner of the data is trustworthy
  - Some fraction of the cloud that host the data could be malicious

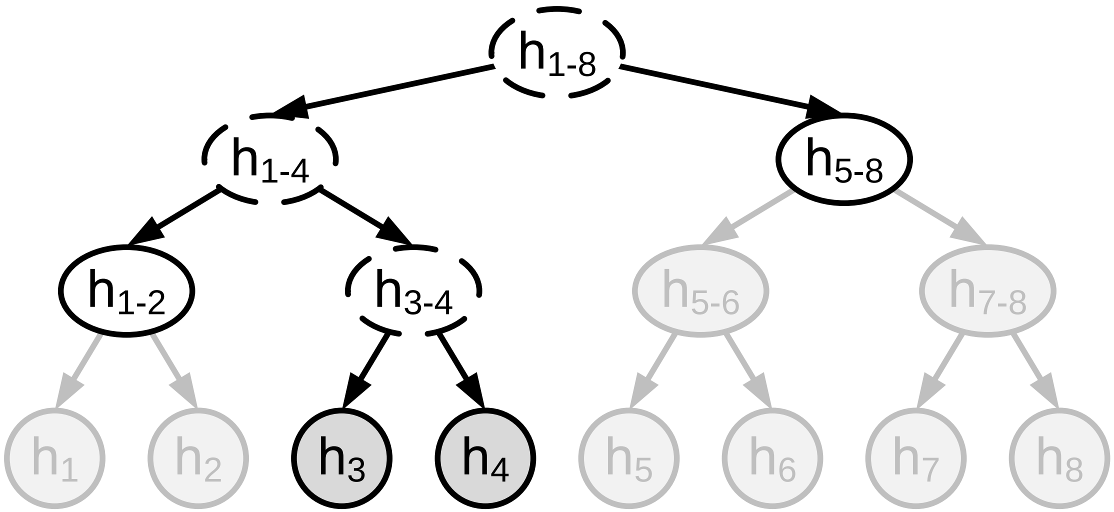
- **Verification with MHT (Merkle Hash Tree)**
  - Previously used to check correctness of outsourced databases
  - Maintain hash hierarchy (MHT) on pre-sorted data
  - VOs (verification objects) attached to query results
    - Signature over the root of MHT
    - Hash values required for re-computing the root of MHT

MHT Example



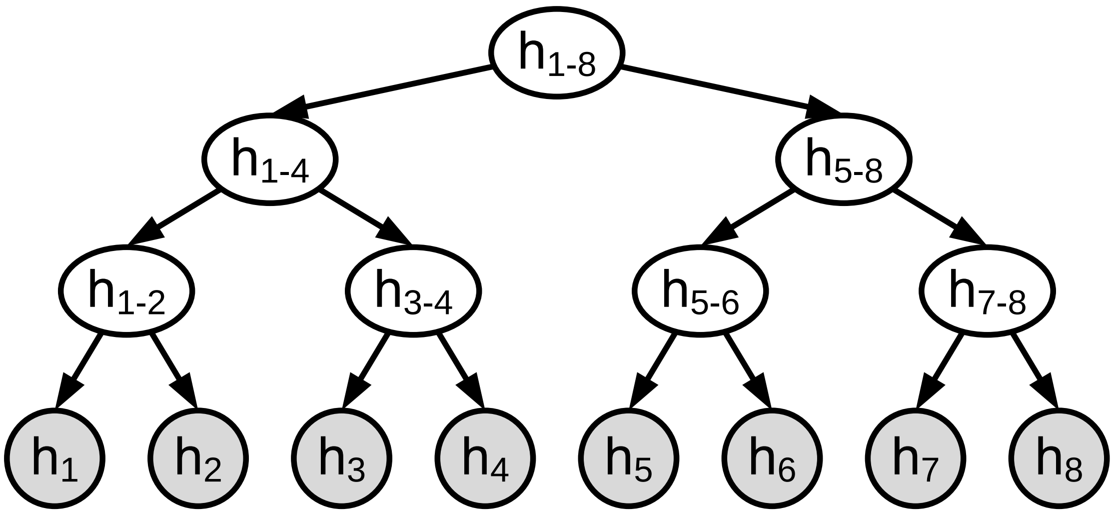
- Query Result = {x3}

MHT Example



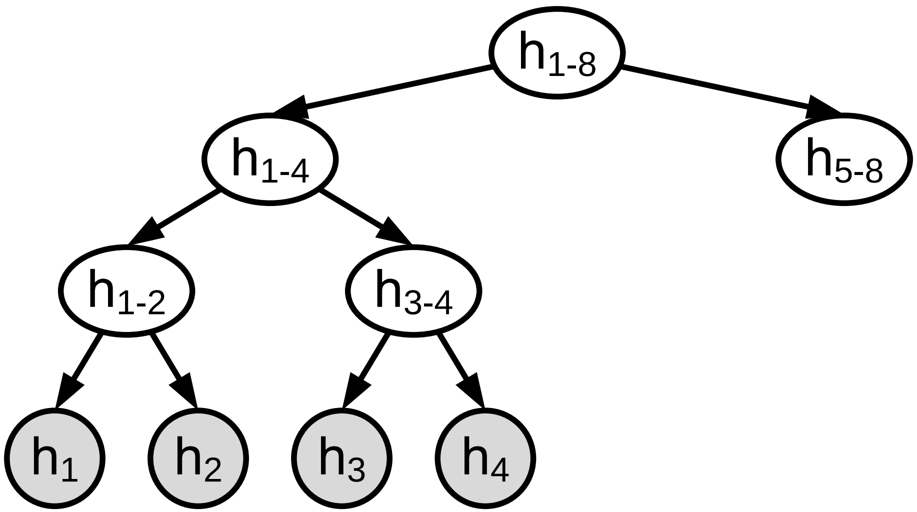
- Query Result = {x3}
- VO = $\{SIG(h_{1-8}), h_4, h_{1-2}, h_{5-8}\}$
- $hash(x_3) \mid h_4 \mid h_{1-2} \mid h_{5-8} == h_{1-8}$?

P-MHT Example



- Table is partitioned across three nodes
  - X1 = {x1, x2, x3}, X2 = {x4, x5, x6}, X3 = {x7, x8}
- Each node maintain a portion of MHT
  - Sufficient to generate the VOs for the tuples located on the node.

P-MHT Example



- Table is partitioned across three nodes
  - X1 = {x1, x2, x3}, X2 = {x4, x5, x6}, X3 = {x7, x8}
- Each node maintain a portion of MHT
  - Sufficient to generate the VOs for the tuples located on the node.

Conclusion and Future Work

- Data-centric: go beyond OS/VM-centric solutions
- Security challenges faced by data-centric cloud security
  - Secure query processing and data sharing
  - Analysis and tracing of data flowing across applications
  - End-to-end verification
- Preliminary design of the DS2 Platform

- Future work
  - Close integration with cloud applications
  - Security guarantees for distributed provenance

Thank You…

Towards a Data-centric View of Cloud Security

http://netdb.cis.upenn.edu/ds2/