Enforcing Encrypted Dynamic Security Constraints in the Cloud

Muhammad Rizwan Asghar, Giovanni Russello, Bruno Crispo

University of Trento, Italy

August 28, 2015
WHY CLOUD STORAGE

- Cost saving
- Scalability
- Efficiency
- Availability
APPLICATION
PROBLEM

Data in cleartext raises privacy issues
A POSSIBLE SOLUTION

Access policies specify who can gain access to the data

Encrypted databases [Asghar’13 CCSW]
A POSSIBLE SOLUTION, BUT

Access policies may leak sensitive information

Access policies specify who can gain access to the data

Encrypted databases [Asghar’13 CCSW]
OUR APPROACH

What kind of access policies?
ROLE-BASED ACCESS CONTROL (RBAC) POLICIES

- **RBAC\(_0\)**
  - Permissions are assigned to roles while roles are assigned to users
  - Encrypted RBAC\(_0\) [Asghar’13 COSE]

- **RBAC\(_1\)**
  - Role hierarchies
  - Encrypted RBAC\(_1\) [Asghar’13 COSE]

- **RBAC\(_2\)**
  - Separation of duties and Chinese wall constraints
  - **Focus of this work!**

- **RBAC\(_3\) = RBAC\(_1\) + RBAC\(_2\)**
SEPARATION OF DUTIES

- Separation of Duties (SoD) constraints aim at providing multiuser control over the resources when there is any conflict-of-interest for completing a business process

- E.g., a clerk issues the purchase order while a manager approves it

- Types
  - Static SoD
    - A user cannot be active in two mutually exclusive roles
  - Dynamic SoD (DSoD)
    - A user can be active in two mutually exclusive roles but …
    - Simple DSoD (SDSoD) – not in the same session
    - Object-Based DSoD (ObDSoD) – not the same object
    - Operational DSoD (OpDSoD) – not all actions in a workflow
    - History-Based DSoD (HBDSoD) = ObDSoD + OpDSoD
HBDSoD

- HBDSoD is the most fine-grained category of DSoD

- A user active in both clerk and manager roles can either *issue* or *approve* a particular instance of the *purchase order*
CHINESE WALL

- It aims at providing confidentiality by preventing illegitimate information flow between domains that are in conflict-of-interest.
- Imagine a consultant organisation that provides services to companies that are in conflict-of-interest, say Google and Microsoft.
E-GRANT

- E-GRANT protects queries and policies stored in outsourced environments
- Our scheme is based on El-Gamal proxy encryption
- An encrypted session is maintained
- It is a multiuser scheme in which entities do not share any encryption keys
- A compromised user can be removed without requiring re-encryption of policies
E-GRANT ARCHITECTURE

Cloud Service Provider

Policy Store

Key Store

Data Store

PEP

Session

Outsourced Environment

Honest-but-curious

Requests

Policy

Session Information

Yes/No

Data

Response

Admin User

Requester

Trusted Key Management Authority

Fully-trusted

MSK = (x, s)

x = x_{i1} + x_{i2}

K_{UA} = (x_{A1}, s)

K_{RA} = (x_{R1}, s)

K_{SA} = (A, x_{A2})

K_{SR} = (R, x_{R2})

\{Policy\}_{K_{UA}}

\{Policy\}_{K_{RA}}

\{Policy\}

\{Policy\}

\{Policy\}_C

\{Policy\}_C

\{REQ\}_{K_{SR}}

\{REQ\}_{K_{RA}}
HBDSoD EVALUATION

**Session Policy Store**

1. **Match(c(.), TD(.)) = YES/NO**
   - c(Action=Issue)
   - c(Action=Approve)
   - c(Object-Type=Purchase-Order)

2. **YES AND YES**
   - TD(REQ) = <..., TD(Action=Approve), TD(Object-Type=Purchase-Order), ... >

3. **NO**
   - Requester
   - REQ = <..., Action=Approve, Object-Type=Purchase-Order, ... >
E-GRANT PROTOTYPE

- We developed a prototype of E-GRANT in Java

- We tested our prototype using a standard machine
  - Microsoft XP Professional version 2002 (SP3)
  - Intel Core2 Duo 2.2 GHz
  - 2 GB RAM
DEPLOYMENT OF CONSTRAINTS

a denotes actions: Clerk or manager ...

Clerk or manager ...

Clerk or manager ...

Clerk or manager ...

Clerk or manager ...

Clerk or manager ...

Clerk or manager ...

d denotes domains: Google/Marketing/Project ...

Google/Marketing/Project ...

Google/Marketing/Project ...

Google/Marketing/Project ...

Google/Marketing/Project ...

Google/Marketing/Project ...

Google/Marketing/Project ...

o represents object (or instance)
REQUEST GENERATION

REQ = <R, A, O, ...>
R is role
A is action  t is time
t is time l is location
O is object  l is location
d denotes (an object with) domains
EVALUATION OF HBDSoD

Time complexity per constraint: number of actions * number of records

a denotes actions
**EVALUATION OF CHINESE WALL**

Time complexity per constraint: number of domains * number of records

d denotes domains
COST OF UPDATING SESSION

REQ = <R, A, O, ...>
R is role
A is action
O is object
t is time
l is location
d denotes domains
CONCLUSIONS AND FUTURE WORK

- E-GRANT enforces separation of duties and Chinese wall constraints in an encrypted manner

- We are capable of providing full-fledged encrypted RBAC style of policies [Asghar’13 PhD-Thesis]

- It is a multiuser scheme where each user has her own key, i.e., removing a user does not require re-encryption of stored policies

- As future work, exploring how encrypted RBAC could be made accountable would be an interesting direction
REFERENCE

- Muhammad Rizwan Asghar, Giovanni Russello, Bruno Crispo,
  **E-GRANT: Enforcing Encrypted Dynamic Security Constraints in the Cloud**, Pages 135-144,
The 3rd International Conference on Future Internet of Things and Cloud (FiCloud) – Special Track on Security, Privacy and Trust,
Rome, Italy, August 2015
(Acceptance rate: 55/178≈30%).
Best Paper Award!
r.asghar@auckland.ac.nz
https://www.cs.auckland.ac.nz/~asghar/