Enforcing Security Policies in Outsourced Environments

Muhammad Rizwan Asghar

PhD Supervisor: Bruno Crispo

BunnyTN3
Trento, Italy
March 12, 2012
Why Outsourcing

- Cost saving
- Scalability
- Efficiency
- Reliability
- Availability
Motivation

Issue: Policy or access request may leak sensitive information

Policy:
Only a dentist may get access from dentist-ward between duty hours (9-17 hrs)
Problem: How to evaluate encrypted policy against encrypted access request
Proposed Solution

- We name our solution ESPOON (Enforcing Security Policies in Outsourced Environments)

- In ESPOON, the Service Provider is assumed *honest-but-curious*

- ESPOON is capable of handling complex policies involving range queries

- ESPOON 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
ESPOON Architecture

Policy Store  \( \rightarrow \)  Administration Point

Policy Store  \( \rightarrow \)  Key Store

Policy Store  \( \rightarrow \)  Data Store

Policy Store  \( \rightarrow \)  PDP

Policy Store  \( \rightarrow \)  PEP

Policy Store  \( \rightarrow \)  Requester

Policy Store  \( \rightarrow \)  PIP

Policy Store  \( \rightarrow \)  Admin User

Policy Store  \( \rightarrow \)  Trusted Key Management Authority

\( (i) \)  
\( \{Policy\}_{K_{su}} \)

\( K_{ua} = (x_{A1}, s) \)

\( (ii) \)  
\( C\{Policy\} \)

\( K_{ra} = (A, x_{A2}) \)

\( K_{sr} = (R, x_{R2}) \)

\( (1) \)  
\( \{REQ\}_{K_{ur}} \)

\( (2) \)  
\( TD\{REQ\} \)

\( (3) \)  
\( C\{Policy\} \)

\( (4) \)  
\( \{Context\}_{K_{ur}} \)

\( (5) \)  
\( Yes/No \)

\( (6) \)  
\( TD\{Context\} \)

\( (7) \)  
\( Response \)

Trusted Key Management Authority

\( MSK = (x, s) \)

\( Params = (g, h = g^i, H, f) \)

\( x = x_{i1} + x_{i2} \)

Outsourced Environment

Service Provider

Admin User

Requester

Trusted but can be removed

Fully-trusted

Partially-trusted but can be removed

trusted_user
Policy Representation

Policy:
Only a dentist may get access from dentist-ward between duty hours (9-17 hrs)

AT = Access Time
Policy Evaluation

Access Time = 10hrs

TD(Requester=Dentist)
TD(Location=Dentist-Ward)

C(Requester=Dentist)
C(Location=Dentist-Ward)

AT = Access Time

C(AT:0****)
C(AT:*1***)
C(AT:**0**)
C(AT:***1*)
C(AT:****0)

No
Yes
No
Yes
No
Yes

AT = Access Time
Policy Evaluation (2)
Performance Evaluation: Requester

- **String Attribute**: $O(n)$, $n$ is the number of string attributes

- **Numerical Attribute**: $O(ns)$, $n$ is the number of numerical attributes each of size $s$
- **String Attribute:** \(O(nm)\), \(n\) is the number of string attributes and \(m\) is the number of string comparisons

- **Numerical Attribute:** \(O(nms^2)\), \(n\) is the number of numerical attributes and \(m\) is the number of numerical comparisons each of size \(s\)
Related Work

- Schemes supporting access controls in outsourced environments require re-generation of keys and re-encryption of data for any administrative changes [Vimercati et al. CSAW’07 VLDB’07]

- Schemes supporting queries on encrypted data do not support access policies [Dong et al. DBSec’08, Song et al. S&P’00, Boneh et al. EUROCRYPT’04, Curtmola et al. CCS’06, Hwang and Lee LNCS’07, Boneh and Waters TCC’07, Wang et al. SOFSEM’08, Baek et al. ICCSA’08, Rhee et al. JSS’10, Shao et al. Inf. Sci.’10]

- Encrypted data with CP-ABE policy reveals the policy structure [Narayan et al. CCSW’10]

- Hidden credentials schemes do not support complex policies and require parties to be online [Holt et al. WPES’03, Bradshaw et al. CCS’04]
Conclusions and Future Work

- **Conclusions**
  - ESPOON enforces policies in outsourced environments
  - ESPOON supports complex policies including range queries
  - ESPOON employs a multiuser scheme where users do not share keys

- **Future work**
  - *Support of full-fledged RBAC style of policies (current focus)*
  - Secure auditing mechanism in ESPOON
  - Support for negative authorisation policies
  - Dynamic updates of attributes within a request
References


Thank You!

Any Questions?

asghar@disi.unitn.it
Performance Evaluation: Policy Deployment

- **String Comparison**: For both enc and re-enc: $O(n)$, $n$ is the number of string comparisons

- **Numerical Comparison**: For both enc and re-enc $O(ns)$, $n$ is the number of numerical comparisons each of size $s$
Key Distribution

- A Trusted Key Management Authority (KMA) is initialised with security parameters to generate
  - Master secret key $x$ and $s$
  - Public parameters $(g, h=g^x, H, f)$

- For each user $i$, the KMA
  - randomly generates $x_{i1}$
  - calculates $x_{i2} = x - x_{i1}$

- Finally, the KMA securely transmits
  - $K_{U_i} = (x_{i1}, s)$ to user $i$
  - $K_{S_i} = (x_{i2}, i)$ to the Server Provider
Policy Deployment: Admin User Side

\[ e \xrightarrow{K_{U_A}} \text{PD-Condition-Enc} \xrightarrow{} \{ e \}_{K_{U_A}} = \begin{cases} \hat{c}_1 = g^{r_A + \sigma_A}, & \sigma_A = f_s(e) \\ \hat{c}_2 = \hat{c}_1^{x_{A1}} \\ \hat{c}_3 = H(h^{r_A}) \end{cases} \]
Policy Deployment: Service Provider Side

\[ c(e) = \begin{cases} 
  c_1 = (\hat{c}_1^{x_A^2}) \cdot \hat{c}_2 = (g^{r_A + \sigma_A})^x = h^{r_A + \sigma_A} \\
  c_2 = \hat{c}_3 = H(h^{r_A}) 
\end{cases} \]
Request: Requester Side

\[
e' \xrightarrow{\text{PE-Attributes-Enc}} \{ e' \} _{K_{UR}} = \begin{cases} 
t_1 = g^{-r_{e'}}^{\sigma_{e'}}, \text{ where } \sigma_{e'} = f_s (e') \\
t_2 = h^{r_{e'}}^{x_{R1}} g^{x_{R1} \sigma_{e'}} = g^{x_{R2} \sigma_{e'}} \end{cases}
\]
Request: Service Provider Side

\[ \{e'\}_{K_{UR}} \xrightarrow{\text{PE-Attributes-Re-Enc}} \text{TD} (e') = t_1^{x_{R_2}} t_2 = g^{x_{\sigma'c}} \]
Policy Evaluation

\[ TD\ (e') = T \]

\[ c (e) = (c_1, c_2) \]

\[ c_2 \leq H(c_1 T^{-1}) \]

Match

Yes or No