OAuth and ABE based Authorization in Semi-Trusted Cloud Computing

Anuchart Tassanaviboon and Guang Gong

Department of Electrical and Computer Engineering
University of Waterloo
CANADA

<http://comsec.uwaterloo.ca/~atassana, ggong>
Outline

- **Security** requirements in cloud environment
- Solutions & **challenges** in semi-trusted cloud computing (STCC)
- **Review** of OAuth and ABE based schemes
- **AAuth**: A new authenticated **authorization** scheme for securing STCC
- **Performance** evaluation and simulation
- Conclusions and remarks
Vulnerabilities

- **Web-interface flaws**, XML signature wrapping, legacy same origin policy, unsecured browser authentication
- **Leak virtual isolation**, side/covert channel, cross-tenant data access
- **Image insanity**, malicious/illegal images
- Limited **network control**, under-provisioning, limited QoS, new form of DoS
- **Weak access control**, weak credentials, weak tokens, coarse authorization
- **Lack of standards**, APIs, inter-operations
**Review of Access Control (AAA)**

**Typical** models
- Centralized server
- Client-server: Kerberos/Active Directory
- HTTP: OpenID/OAuth

**Cloud** problems and challenges
- Trust boundary is expanded to CSPs
- CSPs are untrusted or semi-trusted
- A shared trusted domain doesn’t present
- A single trusted domain is unscalable
An authorizer arbitrarily grants accesses

Cloud servers reveal sensitive data

Cloud servers disobey the access policies

Weak tokens cause fabrication, replay attacks, etc.

Lock-in vendors
Kerberos

Key Distribution Center (KDC)
Owner Client Resource Server (RS)
Authentication Server (AS)
Database
Ticket Granting Server (TGS)

IDCLI: Client ID
IDRS: Resource Server ID
IDTGS: TGS ID
TS: Timestamp
LT: Lifetime
K_A: A’s Secret key in KDC
SKB: CLI – B Session key between
TGT = (IDCLI, IDTGS, TS, LT, SKTGS)
Auth = (IDCLI, TS)
ST = (IDCLI, IDSR, TS, LT, SKRS) KOwner = Hash(password, salt)

1. (IDCLI, IDTGS, LT)
2. (IDTGS, TS, LT, SKTGS)KOwner||{TGT}KTGS
3. (IDRS, LT)||{Auth}SKTGS
   ||{TGT}KTGS
4. (IDRS, TS, LT, SKRS)SKTGS
   ||{ST}KRS
5. {Auth}SKRS||{ST}KRS
6. Resource

Tassana-Gong (University of Waterloo)
OAuth

Tassana-Gong (University of Waterloo)

November 14, 2011 7 / 1
In order to adapt to our scheme, a modified CP-ABE will be introduced later.
Design Goals

- **Data owners** contribute to token generation.
- Data is encrypted in an **end-to-end** fashion.
- **Policies** are enforced by cryptographic functions.
- **Token** knowledge is distributed among CSPs for reducing risks.
- Scheme is integrated with existing standards and cloud entities.
Data owner ($O$): (owners for short) entities, i.e., end-users or software applications, who have resource ownerships and the right to grant access to protected data.

Cloud server ($S$): (servers for short) cloud-storage or cloud-database providers that host protected data and provide basic data-services, i.e., read, write, and delete.

Consumers ($C$): web or traditional applications service provider (ASPs) that use owners’ data to provide services to the owners.

Authority ($AA$): trusted organizations or agencies who legitimately define descriptive attributes to eligible consumers.

Authorizer ($AZ$): the server who runs AAuth protocol, then issues ABE-based tokens to eligible consumers.
System Model (Example)

Owner 'Jane'

Authority 'authority.org'

Authorizer 'mail.net'

Consumer 'print.com'

Server 'photos.com'

# Confined attributes
AND [OWNER=Jane@photos.net]
AND [SEC-CLASS=3]
AND [PERMIS=r]
AND [TIMESLOT=2011/06/27/13/**]
AND # Descriptive attributes
[(OWNERe@mail.net=Jane@mail.net) OR
((NAME=printer.com) AND
(SERVICE=print) AND
(LOCAT=canada) OR
(TRUST-LEV=3))].
AAuth Authorization Scheme in a Nutshell

Diagram showing the interactions between authority, consumer, autorizer, owner, user-agent, and server, with keys and locks indicating permissions and access levels.
AAuth Components

- Defined Attributes

  - FILE-LOC = URI
  - OWNER = ownerId
  - PERMIS = \langle r | w \rangle
  - SEC-CLASS = \langle 1 − 5 \rangle
  - TIMESLOT = yyyy/mm/dd/hh/nn

- Access Policy $\mathcal{A}$

  $\mathcal{A} = [\text{FILE-LOC}] \text{ AND } [\text{OWNER}] \text{ AND } [\text{SEC-CLASS}] \text{ AND } [\text{PERMIS}] \text{ AND } [\text{TIMESLOT}] \text{ AND } [(\text{OWNER@AUTHZ}) \text{ OR } (\text{Descriptive Boolean Algebra})].$
AAuth Components (Cont.): Access Tree $\tau$

\[
\begin{align*}
R &= q_{R}(0) = s \\
q_{R}(1) &= q_{FL}(0) \\
q_{R}(2) &= q_{OW}(0) \\
q_{R}(3) &= q_{SC}(0) \\
q_{R}(4) &= q_{PM}(0) \\
q_{R}(5) &= q_{TS}(0) \\
q_{R}(6) &= q_{DA}(0) \\
q_{DA}(7) &= q_{ow@AZ}(0) \\
q_{DA}(8) &= q_{AA}(0)
\end{align*}
\]

$R$: Root  \quad FL$: File Location  \quad OW$: Owner  
SC$: Security Class  \quad PM$: Permission  \quad TS$: Timeslot  
DA$: Descriptive Attributes  
OW@:AZ Owner at Authorizer  
AA$: Authority

\begin{itemize}
\item $s = q_{R}(0) = q_{FL}(0) + q_{OW}(0) + q_{SC}(0) + q_{PM}(0) + q_{TS}(0) + q_{DA}(0)$
\item (6,6) threshold
\item (1,2) threshold
\end{itemize}
AAuth Components (Cont.): Archive File

File Desc | Access Policy
---|---
Encry. Meth. | Encry. Key
Integ. Meth. | Integ. Key

\{Header\}_{ABE} \| \{Data File\}_{KE} \| Tail

Access Policy \| Integ. Tag
Modified CP-ABE

Setup($k$)

**Authorizer**

System parameters
Bilinear map $e : G_1 \times G_1 \to G_2$.
Generator $g$ of group $G_1$.
Hash function $H : \{0, 1\}^* \to G_1$.

Randomly selects $\beta \in \mathbb{Z}_p$.
Master Secret Key: $MSK = \langle \beta \rangle$.
Master Public Key: $MPK = \langle G_1, g, h = g^\beta, f = g^{1/\beta} \rangle$.

**Owner**

Randomly selects $\alpha \in \mathbb{Z}_p$.
Owner Secret Key: $OSK = \langle g^\alpha \rangle$.
Owner Public Key: $OPK = \langle e(g, g)^\alpha \rangle$. 
Modified CP-ABE: $\text{Encrypt}(\text{MPK}, m, \tau)$

- Randomly selects $s \in \mathbb{Z}_p$.
- Construct access tree $\tau$ according to $q_R(0) = s$ and an access policy $\mathbb{A}$.
- Let $Y$ be the leave nodes in $\tau$:

  **Ciphertext:**  
  $CT = \langle \tau, \tilde{C} = m \cdot e(g, g)^{\alpha s}, C = h^s, 
  \forall y \in Y : C_y = g^{q_y(0)}, C'_y = H(\text{att}(y))^{q_y(0)}) \rangle$.  

Assume that an attribute set $\omega = \omega' \cup \omega''$ where $\omega'$ = confined attributes, and $\omega''$ = descriptive attribute.

- **Authorizer**: $r \in R \mathbb{Z}_p$, and selects a set $\{r_i \in R \mathbb{Z}_p \mid i \in \omega'\}$ where $\omega'$: confined attributes.
- **Authority**: selects $\{r_j \in R \mathbb{Z}_p \mid j \in \omega''\}$.
- **Owner**: $a \in R \mathbb{Z}_p$.

With ElGamal-like masking, the authorizer, the authority, and the owner jointly compute a private key for the consumer

**Private key:**

$$SK = \langle D = g^{(\alpha + ra)/\beta}, D_k = g^{ra} \cdot H(k)^{r_k}, D'_k = g^{r_k}, \forall k \in \omega \rangle.$$
Given a secret key $SK$ for an attribute set $\omega$.
Let $\tilde{\omega} \supseteq \omega$ denote a new attribute set.
Random value $\tilde{r}$ and random set $\{\tilde{r}_l \mid \forall l \in \tilde{\omega}\}$.
A consumer creates a new private key $\tilde{SK}$ for the attribute set $\tilde{\omega}$:

$$\tilde{SK} = \langle \tilde{D} = D \cdot f\tilde{r}, \forall l \in \tilde{\omega} : \tilde{D}_l = D_l \cdot g^{\tilde{r}_l} \cdot H(l)^{\tilde{r}_l}, \tilde{D}_l' = D'_l \cdot g^{\tilde{r}_l} \rangle.$$
Recursively computes from the root node $R$ of access tree $\tau$ by using node algorithm $\text{DecryptNode}(CT, SK, x)$:

$$F_R = \text{DecryptNode}(CT, SK, R) = e(g, g)^{ra \cdot q_R(0)} = e(g, g)^{ras}$$

If the tree $\tau$ is satisfied by $\omega$ then decryption can be computed by:

$$\text{Decrypt}(CT, SK) = \tilde{C} / (e(C, D) / F_R) = \tilde{C} / (e(h^s, g^{(\alpha + ra)/\beta}) / e(g, g)^{ras})$$

$$= m$$
A Diagram of $\text{DecryptNode}(CT, SK, x)$

\[ F_R = e(g, g)^{ras} \]

\[ F_x = \prod_{z \in \omega_x} F_{z \triangleleft k, \hat{w}_x}^{(0)} \]

\[ F_x = \frac{e(D_k, C_x)}{e(D'_k, C'_x)} \]
AAuth: Service Request Protocol

1. REQ-PRT

2. REQ-POL

3. $[A]_S$

4. RED
AAuth: Token Request Protocol

1. RED
2. HTTP Form
3. Login/password
4. REQ-DES1
5. [{\hat D}_j}]_{AA}
6. \{\hat D}_i, [{\{D}_i]}_AZ, [{\{\hat D}_j}\}_{AA}, g^r
7. g^{(\alpha + ra)}
8. RED
9. \{D}_i, [{\{D}_i}\}_{AZ}, \{D}_j
10. REQ-DES2
11. \{D}_j'}
AAuth: File Access Protocol

1. 

Server S

Consumer C

1.REQ-FILE

2. Chall

3. Resp

4. Archive
<table>
<thead>
<tr>
<th>Timeslot</th>
<th>0</th>
<th>1</th>
<th>...</th>
<th>(n-1)</th>
<th>(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random value, (\bar{s}) Share, (q_{TS}(0))</td>
<td>(\bar{s}(0))</td>
<td>(\bar{s}(1))</td>
<td>...</td>
<td>(\bar{s}(n-1))</td>
<td>(\bar{s}(n))</td>
</tr>
<tr>
<td>(q_{TS}(0,0))</td>
<td>(q_{TS}(0,1) = q_{TS}(0,0) + \bar{s}(1))</td>
<td>...</td>
<td>(q_{TS}(0,n-1))</td>
<td>(q_{TS}(0,n) = q_{TS}(0,n-1) + \bar{s}(n))</td>
<td></td>
</tr>
<tr>
<td>Component, (C_{ST})</td>
<td>(C_{ST}(0))</td>
<td>(C_{ST}(1) = g^{q_{TS}(0,1)})</td>
<td>...</td>
<td>(C_{ST}(n-1))</td>
<td>(C_{ST}(n) = g^{q_{TS}(0,n)})</td>
</tr>
<tr>
<td>Component, (C'_{ST})</td>
<td>(C_{ST}(0))</td>
<td>(C'<em>{ST}(1) = H(Att</em>{ST}(1))^{q_{TS}(0,1)})</td>
<td>...</td>
<td>(C_{ST}(n-1))</td>
<td>(C'<em>{ST}(n) = H(Att</em>{ST}(n))^{q_{TS}(0,n)})</td>
</tr>
<tr>
<td>Component, (C)</td>
<td>(C(0))</td>
<td>(C(1) = C(0) \cdot h^{\bar{s}(1)})</td>
<td>...</td>
<td>(C(n-1))</td>
<td>(C(n) = C(n-1) \cdot h^{\bar{s}(n)})</td>
</tr>
<tr>
<td>Component, (\bar{C})</td>
<td>(\bar{C}(0))</td>
<td>(\bar{C}(1) = \bar{C}(0) \cdot e(g,g)^{\alpha \bar{s}(1)})</td>
<td>...</td>
<td>(\bar{C}(n-1))</td>
<td>(\bar{C}(n) = \bar{C}(n-1) \cdot e(g,g)^{\alpha \bar{s}(n)})</td>
</tr>
<tr>
<td>Secret mask, (s)</td>
<td>(s(0))</td>
<td>(s(1) = s(0) + \bar{s}(1))</td>
<td>...</td>
<td>(s(n-1))</td>
<td>(s(n) = s(n-1) + \bar{s}(n))</td>
</tr>
</tbody>
</table>
AAuth: Token Delegation

The web site ‘printer.com’ can ask the website ‘poster.com’ to print a poster for a file ‘pic-1’ in the time slot ‘2011|06|27|13|**’

‘printer.example.com’

FILE-LOC = http://photos.com/2010/brunce/pic-1,
FILE-LOC = http://photos.com/2010/brunce/pic-2,
SEC-CLASS = 3, PERMIS=r,
/* current time slot */
TIMESLOT = 2011|06|27|13|**,
/* future time slot(s)*/
TIMESLOT = 2011|06|27|14|**.

‘poster.com’

FILE-LOC = http://photos.com/2010/brunce/pic-1,
SEC-CLASS = 3, PERMIS = r,
/* current time slot */
TIMESLOT = 2011|06|27|13|**.

Tassana-Gong (University of Waterloo)
## Recap: The procedures and protocols in AAuth

<table>
<thead>
<tr>
<th>Procedures/Protocols</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Setup procedure</strong></td>
<td>1. A bilinear group $G_1, G_2$</td>
</tr>
<tr>
<td></td>
<td>2. A bilinear map $e$</td>
</tr>
<tr>
<td></td>
<td>3. A generator $g$ of $G_1$</td>
</tr>
<tr>
<td></td>
<td>4. hash function $H$</td>
</tr>
<tr>
<td><strong>File encapsulation procedure</strong></td>
<td>1. An access policy $\mathcal{A}$ from both confined and descriptive attributes</td>
</tr>
<tr>
<td></td>
<td>2. An access tree $\tau$</td>
</tr>
<tr>
<td></td>
<td>3. An archive file</td>
</tr>
<tr>
<td><strong>Service request protocol</strong></td>
<td>An access policy $\mathcal{A}$</td>
</tr>
<tr>
<td><strong>Token request protocol</strong></td>
<td>An ABE-token</td>
</tr>
<tr>
<td><strong>File access protocol</strong></td>
<td>An archive file</td>
</tr>
<tr>
<td><strong>File decapsulation procedure</strong></td>
<td>1. A header in plaintext form</td>
</tr>
<tr>
<td></td>
<td>2. An integrity tag</td>
</tr>
<tr>
<td></td>
<td>3. A data file in plaintext form</td>
</tr>
<tr>
<td><strong>Time slot synchronization protocol</strong></td>
<td>1. Two ciphertext components</td>
</tr>
<tr>
<td></td>
<td>2. Two update values</td>
</tr>
<tr>
<td></td>
<td>3. A new time slot header</td>
</tr>
</tbody>
</table>
Security Analysis

i With **end-to-end encryption** and signature, a cloud server cannot subvert the confidentiality and integrity of the data it is hosting.

ii With **end-to-end authorization**, the access policy is enforced by the encryption algorithm, not by a cloud server.

iii Without **cooperation** between owners and the authority, none of them can individually generate ABE-tokens.

iv Since owners can verify confined keys before combining, the **authorizer** cannot fake keys to owners.

v Separating keys to two parts, each of which is individually sent to consumer, to fabricate keys, owners face DLP while consumers face DBDH problems.

vi The scheme can prevent eavesdropping, active, MITM, off-line attacks from external adversaries.
## On-line Cryptographic Cost

<table>
<thead>
<tr>
<th>Role</th>
<th>Signing</th>
<th>Verify</th>
<th>Exponent</th>
<th>Paring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td></td>
<td>1</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>Consumer</td>
<td></td>
<td>2</td>
<td></td>
<td>$2(</td>
</tr>
<tr>
<td>Authorizer</td>
<td>2</td>
<td></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Authority</td>
<td>1</td>
<td></td>
<td>$2</td>
<td>I - 5</td>
</tr>
<tr>
<td>Server</td>
<td>1</td>
<td></td>
<td>$2</td>
<td>L</td>
</tr>
</tbody>
</table>
### Additional Communication Cost

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Additional messages</th>
<th>Message flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service request</td>
<td>2</td>
<td>( C \rightarrow S )</td>
</tr>
<tr>
<td>Token request</td>
<td>2</td>
<td>( AZ \rightarrow AA )</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>( O \rightarrow AZ )</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>( C \rightarrow O )</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>( C \rightarrow AA )</td>
</tr>
<tr>
<td>File access</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>
Simulations

- Tool: OMNet++
- Settings: the cloud network has a bandwidth at 400 packets/s, each owner continuously requests services in exponential distribution, each service request transfers three 256 KB-files as a dummy load, the number of owners (users) starts from 100 to 700.

### Average Latency vs. Crypt Computing Time

<table>
<thead>
<tr>
<th># Owner</th>
<th>Crypt time</th>
<th>Random number</th>
<th>Rand between Crypt</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>6.6515116</td>
<td>0.1</td>
<td>0.099090576</td>
</tr>
<tr>
<td>200</td>
<td>6.7936554</td>
<td>0.3235168457</td>
<td>0.132351685</td>
</tr>
<tr>
<td>300</td>
<td>10.169864</td>
<td>0.4959411621</td>
<td>0.1</td>
</tr>
<tr>
<td>400</td>
<td>16.5296745</td>
<td>0.0099487305</td>
<td>0.100994873</td>
</tr>
<tr>
<td>500</td>
<td>22.9078254</td>
<td>0.20703125</td>
<td>0.1</td>
</tr>
<tr>
<td>600</td>
<td>29.271162</td>
<td>0.3104248047</td>
<td>0.13104248</td>
</tr>
<tr>
<td>700</td>
<td>35.6323535714</td>
<td>0.9291992188</td>
<td>0.192919922</td>
</tr>
</tbody>
</table>

### OAuth-AAuth

![Graph showing Latency vs. Number of Owners](image)

- **Latency (Second)**
- **Number of Owners**
- **Oauth**
- **Aauth**
Conclusions & Remarks

1. **ABE-tokens** for each authorization grant.
2. A **user-centric** system in which an owner controls the authorization system to protect her resources.
3. **End-to-end** cryptographic functions from an owner to a consumer.
4. A light-weight encryption for time slot **synchronization**.
5. No significant computation cost for users.
6. AAuth’s cost is independent of the number of users in the system.
7. An acceptable increasing cost is compensated by achieving better security than OAuth.
8. **AAuth** is as secure as the original CP-ABE scheme and can resist both internal and external adversaries.
## The comparison of Kerberos, OAuth, and AAuth

<table>
<thead>
<tr>
<th></th>
<th>Kerberos</th>
<th>OAuth</th>
<th>AAuth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust platform</td>
<td>Client</td>
<td>Browser</td>
<td>Browser</td>
</tr>
<tr>
<td>SSO</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Key management</td>
<td>No</td>
<td>No</td>
<td>Integrated &amp; distrib-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>uted</td>
</tr>
<tr>
<td>Data-at-rest</td>
<td>Plaintext</td>
<td>Plaintext</td>
<td>Ciphertext</td>
</tr>
<tr>
<td>Policy mechanism</td>
<td>ACL / capabili-</td>
<td>ACL / capabil-</td>
<td>ABE attributes</td>
</tr>
<tr>
<td></td>
<td>ties</td>
<td>ties</td>
<td></td>
</tr>
<tr>
<td>Policy enforced by</td>
<td>server</td>
<td>server</td>
<td>ABE decryption</td>
</tr>
<tr>
<td>Token generation</td>
<td>AS &amp; TGS</td>
<td>OAuth provider</td>
<td>Owner, Author-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>izer, and Autho-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>rity(s)</td>
</tr>
<tr>
<td>Ext. attacks resisted by</td>
<td>Time synch.</td>
<td>SSL/TLS</td>
<td>multi SSL/TLS</td>
</tr>
<tr>
<td>Int. attacks resisted by</td>
<td>No</td>
<td>No</td>
<td>modified CP-ABE</td>
</tr>
</tbody>
</table>