Secure Access to Outsourced Data

Dr. Kui Ren

University at Buffalo
The State University of New York
Disclaimer!

The lecture slides are partially collected from the Internet for the educational purpose only. The lecturer does not claim any credit for them and the copyrights belong to the original authors.

Special thanks to Prof. Robert Deng at Singapore Management University for the slides used in Lecture 2.
Outline

• Access Control in Cloud Storage Systems
• Attribute-based Access Control
• Basic Construction
• Improving Granularity – Attribute Revocation
• Improving Efficiency – Decryption Outsource
• Improving Privacy – Policy Hidden
• Summary
Lecture 2: Access Control in Cloud Storage Systems
Cloud Storage Systems

Google Drive

amazon web services™

salesforce.com®

Windows Azure

facebook
Traditional Access Control Model

**Pros:** Flexible and scalable, MAC, DAC, RBAC

**Cons:** Data vulnerable to compromise

Trusted to mediate Access control

Trusted to keep data confidential
Biggest Examples of Data Breaches

http://www.identityhawk.com/biggest-examples-data-breaches

- **Bank of New York Mellon Feb 2008**: lost data storage tapes containing information of 12.5 million people, led to an undisclosed amount of stolen funds...

- **Heartland Payment Systems Early 2009**: hackers infiltrated its database and gained access to the more than 100 million credit card transactions it processes each month. The company paid more than $41.1 million to settle claims.
It’s often unrealistic to assume that servers are trusted

- Cloud computing for outsourced data storage: hardware not under direct control of data owners
- Portable devices storing electronic medical records for emergency access: devices might be lost or stolen
- Software are not guaranteed to be bug-free
- Insider attacks
- ......
Untrusted Servers

- **General solution**: Store data in encrypted form

- **Good practice even for “trusted” servers**: The principle of defense in depth
Access Control by Encryption

Idea: Need secret key to access data

- Ciphertexts stored on server;
- Each user can decrypts its own data
Sharing Encrypted Data with Others

- **Public key solution**
  - Large overhead in public key certificate management

- **Symmetric key solution**
  - Online key distribution
A Wishlist for Storing Encrypted Data on Untrusted Servers

• Key management is scalable and offline
• No need for an online trusted party to mediate access control
• Expressive and scalable access control polices

Attribute-Based Encryption (ABE) does this!
Attribute-based Access Control
Evolution of Attribute Based Encryption

- Identity-Based Encryption
- Fuzzy Identity-Based Encryption
- Key-Policy Attribute-Based Encryption
- Attribute-Based Encryption
- Ciphertext-Policy Attribute-Based Encryption
- Functional Encryption
Attribute-Based Encryption (ABE)
[Sahai, Waters CCS’05]

- Encrypt data to users with certain attributes
- One-to-many public key encryption
- Built-in access control mechanism

"All professors, CS PhD"

Alice

Bob

Charlie

This is a Key-Policy Attribute-Based Encryption!
Key-Policy Attribute-Based Encryption (KP-ABE) [Sahai, Waters CCS’05]

- Ciphertext has a set of attributes
- Keys reflect a tree access structure
- Decrypt iff attributes from Ciphertext satisfy key’s policy

“All professors, CS PhD”
Access Control via KP-ABE

SK_{Bob}: “CS Dept.”
“Professor”

SK_{Kevin}: “CS Dept.”
“Master”

CS Dept.
Professor

CS Dept.
Master
Access Control via KP-ABE

Scenarios: Database (e.g., e-Health System)
Pro: Data associated with some attributes
Con: Users hold multiple secret keys for different access policies


But...In real cloud storage systems, users may associated with some attributes.
How about defining access policy on users’ attributes?

CP-ABE (Ciphertext-Policy Attribute-Based Encryption) !!!
Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [Bethencourt, Sahai, Waters S&P’07]

- Ciphertext is associated with an access policy

(CS AND PhD) OR Prof

- Secret key is associated with attributes
  - Attributes are mathematically incorporated into the key
Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [Bethencourt, Sahai, Waters S&P’07]

- Decrypt iff attributes in the key satisfy the policy of Ciphertext

{EE, Prof} \textbf{Satisfies} \hspace{1cm} (CS AND PhD) OR Prof

Alice \hspace{1cm} \textbf{Message}

- No 3\textsuperscript{rd} party explicitly evaluates the policy and makes access decision. Policy checking done inside the crypto
Access Control via CP-ABE

Access is determined by the set of attributes on the policy and the corresponding keys in the ciphertext.

- **SK_{Bob}:** "CS Dept." "Professor"
- **SK_{Kevin}:** "CS Dept." "Master"

Policy Example:
- Professor
- AND
- CS Dept.
- PhD

Access determined by whether the attributes match the policy.
Advantages of Attribute-Based Access Control

Access policy is defined by owners

Access policy is enforced by the cryptography

• nobody explicitly evaluates the policies and makes an access decision

Only one copy of ciphertext is generated for each file
Access Control in Cloud Storage Systems

**CP-ABE** is more **suitable** than **KP-ABE** for access control in cloud storage systems. Because:

- Owners can define access policy for each file based on user’s attributes
- Users only hold one secret key
- Owners can change the access policies without changing public keys and secret keys
Basic Construction
Elliptic Curve Techniques

- $G$: multiplicative of prime order $p$. (Analogy: $\mathbb{Z}_q^*$)

- Intuitive Hardness Discrete Log:
  Given: $g, g^a$ Hard to get: $a$

- Bilinear map $e: G \times G \rightarrow G_T$
  Def: An admissible bilinear map $e: G \times G \rightarrow G_T$ is:
    - Non-degenerate:
      $g$ generates $G$ $\Rightarrow$ $e(g,g)$ generates $G_T$.
    - Bilinear: $e(g^a, g^b) = e(g,g)^{ab}$ $\forall a,b \in \mathbb{Z}_p$, $g \in G$
    - Efficiently computable
CP-ABE Algorithms

Setup(\λ) \rightarrow MSK, PK

KeyGen(MSK,Attrs.) \rightarrow SK

Encrypt(PK,M,Access policy) \rightarrow CT

Decrypt(SK,CT) \rightarrow M
System Setup [Bethencourt, Sahai, Waters S&P’07]

Authority

\[ a, b \in \mathbb{Z}_p \]

MSK

\[ \text{MSK} = a \]

Public Key

\[ \text{PK} = ( g, g^b, e(g, g)^a, H: \{0,1\}^* \rightarrow G ) \]
Key Generation [Bethencourt, Sahai, Waters S&P’07]

Authority

Authority issues secret keys for users who have attributes

Bob
“CS Dept.”
“Professor”

Kevin
“CS Dept.”
“Master”

James
“EE Dept.”
“PhD”
Central Issue: Prevent User Collusions

Users must not be able to collude by combining their attributes
Key Generation [Bethencourt, Sahai, Waters S&P’07]

Authority

\[ SK = (g^{a+\mathbf{t}}, g^{\mathbf{t}}, H(\text{"PhD"})^{\mathbf{t}}, H(\text{"CS Dept."})^{\mathbf{t}}, H(\text{"TA"})^{\mathbf{t}}) \]

‘\( \mathbf{t} \)’: random number in \( \mathbb{Z}_p \)

‘\( \mathbf{t} \)’ ties components together

Bob has attributes:

\{“PhD”, “CS Dept.”, “TA”\}

Personalization!

Collusion Resistance
Key Personalization (Intuition)

Kevin: “CS Dept.”

SK
Random \( t \)

James: “PhD”

SK
Random \( t' \)

\[ g^{a+b^t}, g^t, H("CS Dept.")^t, \]

\[ g^{a+b^{t'}}, g^{t'}, H("PhD")^{t'} \]

Components are incompatible
(Formal security proofs in papers)
Data Owner

PK = ( g, g^b, e(g, g)^a, H: \{0,1\}^* \rightarrow G )

Given a file $M$ and an access policy, data owner will perform the following:

- **CS Dept.**
- **PhD**
- **Prof**

**OR**
Encryption [Bethencourt, Sahai, Waters S&P’07]

Data Owner generates random $s$, then computes

$$CT = (M \cdot e(g,g)^{as}, g^s),$$

$$C_1 = (g^{bs_1}H(“Prof“)^{r_1}, g^{r_1}), \quad C_2 = (g^{bs_2}H(“PhD“)^{r_2},$$

$$C_3 = (g^{bs_3}H(“CS Dept.“)^{r_3}, g^{r_3})$$
Decryption [Bethencourt, Sahai, Waters S&P’07]

Ciphertext CT

\[
CT = ( M, e(g, g)^{as}, g^s, C_1 = (g^{bs_1}H(“Prof”)r_1, g^{r_1}), \\
C_2 = (g^{bs_2}H(“PhD”)r_2, g^{r_2}), C_3 = (g^{bs_3}H(“CS Dept.”)r_3, g^{r_3}) )
\]

Secret Key SK

\[
SK = ( g^{a+b}, g^t, H(“Prof”)^t, H(“PhD”)^t, H(“CS Dept.”)^t )
\]

\[
e(g^{a+b}, g^s) = e(g, g)^{as} e(g, g)^{b+s}
\]

\[
\begin{array}{c|c}
\text{“Prof”} & \text{OR} & \text{“PhD” AND “CS Dept.”} \\
\hline
\frac{e(g^{bs_1}H(“Prof”)r_1, g^t)}{e(g^{r_1}, H(“Prof”)^t)} & \frac{e(g^{bs_2}H(“PhD”)r_2, g^t)}{e(g^{r_2}, H(“PhD”)^t)} & \frac{e(g^{bs_3}H(“CS Dept.”)r_3, g^t)}{e(g^{r_3}, H(“CS Dept.”)^t)} \\
\end{array}
\]

\[
e(g, g)^{b+s} = e(g, g)^{b+s2} e(g, g)^{b+s3} = e(g, g)^{b+s}
\]
Security [Bethencourt, Sahai, Waters S&P’07]

Theorem: System is (semantically) secure under chosen key attack

Number Theoretic Assumption:

Bilinear Diffie-Hellman Exponent [BBG05]

Given $g^a, g^b, g^c$ distinguish $e(g,g)^{abc}$ from random
Improving Granularity – Attribute Revocation
Efficient Revocation

**User Revocation**
When one attribute is revoked, the user loses all the decryption privilege of all the ciphertexts (e.g., a user is leaving a company)

**Attribute Revocation**
When one attribute is revoked, the user still can use its other attribute to decrypt ciphertexts (e.g., a user is degraded from PM to Developer)
Requirements of Efficient Revocation

• **Protecting Previous Encrypted Data**
  - Once an attribute is revoked from the user, it cannot use this revoked attribute to decrypt the previous encrypted data

*Dynamic Credentials and Ciphertext Delegation for Attribute-Based Encryption* [Sahai, Seyalioglu, Waters CRYPTO’12]

  - Assume the user may access the files necessary for his work and not download all files he has access to (e.g., enforced by access logs).
  - Ciphertexts Update
  - Key Update
Requirements of Efficient Revocation

- Protecting Previous Encrypted Data
  - Once an attribute is revoked from the user, it cannot use this revoked attribute to decrypt the previous encrypted data

Attribute based data sharing with attribute revocation [Yu, Wang, Ren, Lou AsiaCCS'10]

- Re-generate Secret Keys
- Re-encrypt Ciphertexts (Proxy Re-encryption)
Requirements of Efficient Revocation

• **Protecting Newly Encrypted Data**
  - Once an attribute is revoked from the user, it cannot use this revoked attribute to decrypt the newly encrypted data
  - **Key Update**

• **Guaranteeing Newly Joined Users**
  - The newly joined users should still be able to decrypt previous encrypted data, if they have sufficient attributes
  - **Ciphertexts Update**
Attribute-based Access Control with Efficient Revocation

YANG, JIA, REN (AISACCS'13)

- Each attribute is assigned a version number
- To revoke an attribute, the authority updates the version number and generates an update key
- Secret key update only by non-revoked users (Protecting newly encrypted data)
- Ciphertext update by cloud servers (Guaranteeing newly joined users)
Revocation for Multiple Authorities Systems

AND

CS dept.

OR

manager

marketing

Authority in UB

Authority in Google
Revocation for Multiple Authorities Systems

Main Challenge:
Revocation of attributes from one authority should not affect attributes from other authorities

DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems
[Yang, Jia, Ren, Zhang, INFOCOM 2013]

- Idea similar to [Yang, Jia, Ren AisaCCS'13]
- But deal with the multi-authority scenario
Improving Efficiency - Decryption Outsource
Naïve Approach

We have to trust the cloud!
Access Policies in ABE

- May use arbitrary numbers of AND, OR, and t-out-n Threshold gates

- May support integer comparison operators $<$, $>$, $\geq$, $\leq$ by converting them into a Boolean circuit composed of OR and AND gates

- Comparing an attribute to a fixed n-bit integer adds about n components to the policy
  - Key_Expiry_Date $> X$ (Unix time) increases policy size by about 32 components

- Decryption with 100 policy leaves on iPhone 3G (412Mhz ARM) takes 30s
Outsourcing ABE Decryption

[Green, Hohenberger, Waters, UNSNIX Security'11]

Authority issues a Transform Key (TK) and a Secret Key (SK) to Alice

(SK, TK)
Outsourcing ABE Decryption (2)

[Green, Hohenberger, Waters, UNSNIX Security'11]

\[
\text{SK} \quad \rightarrow \quad (TK, CT) \quad \rightarrow \quad \text{CT'} \leftarrow \text{Transform}(TK, CT) \quad \rightarrow \quad \text{Dec(SK, CT')} \rightarrow \text{Data}
\]

Most computation done here

Little computation done here

CT

Storage

Proxy

Cloud
How It Works?

Ciphertext CT

\[ CT = (M \cdot e(g,g)^a, g^s, C_1 = (g^{bs_1}H("Prof"), r_1, g^{r_1}), \]

\[ C_2 = (g^{bs_2}H("PhD"), r_2, g^{r_2}), C_3 = (g^{bs_3}H("CS Dept."), r_3, g^{r_3}) ) \]

Alice: SK = \( (g^{a+b^*}, g^t, H("PhD")^t, H("CS Dept."))^t \)

Proxy: Transform(TK, CT) = CT' = \( (M \cdot e(g,g)^a, e(g,g)^{as/z}) \)

Alice computes: \( M \cdot e(g,g)^{as/(e(g,g)^{as/z})}z = M \)
No Verifiability in Green et al’s Scheme

\[(TK, CT) \rightarrow Proxy\]

\[SK \quad CT' \leftrightarrow \text{Transform}(TK, CT)\]

\[
\text{Dec}(SK, CT') \rightarrow \text{Data. Is the decryption correct?}
\]

Verifiable Outsourced ABE Decryption

[Lai, Deng, Guan, Weng, to appear in IEEE TIFS]

Ability for user to verify the decryption is correct, i.e., \text{Data} is indeed decryption of \text{CT}

Necessary condition: \[
\text{Dec}(SK, CT, CT') \rightarrow \text{Data}
\]
Experiment Results

- 224-bit MNT ECC
- 2.53GHz Intel Core Duo, 4GB RAM, Linux
- 800Mhz ARM-based, 278MB RAM, Android
Decryption Outsourcing for Multi-Authority Cloud Storage Systems

Authority in UB → CS dept. → OR → Authority in Google

DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems
[Yang, Jia, Ren, Zhang, INFOCOM 2013]

Token-based Decryption Outsourcing Mechanism for Cloud Storage Systems with Multiple Authorities
Improving Privacy – Policy Hidden
E-health System

Access policies may leak lots of sensitive Information!!
CP-ABE with Fully Hidden Access Policy

• One can obtain CP-ABE with fully hidden access policy from inner-product predicate encryption (IPE).

• Supporting access policies written in CNF or DNF form, which can result in a super-polynomial blowup in size for arbitrary formulas.

Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products

[Katz, Sahai, Waters, J. Cryptology 2013]

- Each attribute includes two parts: attribute name and attribute value

**Access Policy**

- **OR**
  - **AND**
    - **SS#**: 123-45-6789
    - Affiliation: *University Hospital*
    - Occupation: *Cardiologist*

**Public:**

- **OR**
  - **AND**
    - **SS#**: *
    - Affiliation: *
    - Occupation: *

**Hidden:**

- “123-45-6789”, “University Hospital”, “Cardiologist”
How about simply don’t release the attribute values in the access policy in standard CP-ABE?

Public Params

\[ e: G \times G \rightarrow G_T \]
\[ g, g^b, e(g,g)^a, H: \{0,1\}^* \rightarrow G \]

Access Policy

\[ CT = (M.e(g,g)^{as}, g^s, C_1 = (g^{bs_1}H("123-"))^{r_1}, g^{r_1}), \]
\[ C_2 = (g^{bs_2}H("UH"))^{r_2}, g^{r_2}), C_3 = (g^{bs_3}H("Cardio"))^{r_3}, g^{r_3}) \]
Dictionary Attack on Attribute Values

PP: \( g, (g^b, e(g, g)^a), F: \{0, 1\}^* \rightarrow G \)

Ciphertext:
\[
CT = ( M \cdot e(g, g)^{a_s}, g^s, C_1 = (g^{b_{s1}} H("123-"), r^1, g^{r^1}), C_2 = (g^{b_{s2}} H("UH"), r^2, g^{r^2}), C_3 = (g^{b_{s3}} H("Cardio"), r^3, g^{r^3}) )
\]

The guessed values \textbf{UH \& Cardio} can be verified as above.
Main Idea in [Lai, Deng, Li AsiaCCS'12]
Using *composite order bilinear group* to hide attribute values in ciphertext

\[ G, G_T \text{ are cyclic groups of order } N = p_1p_2p_3p_4 \]
\[ e: G \times G \rightarrow G_T \]

- **Bilinear**: \( \forall a, b \in \mathbb{Z}_N, \forall g \in G, e(g^a, g^b) = e(g, g)^{ab} \)
- **Non-degenerate**: \( \exists g \in G \) such that \( e(g, g) \) has order \( N \) in \( G_T \)

**Orthogonality**: \( e(h_i, h_j) = 1, h_i \in G_{p_i} \) and \( h_j \in G_{p_j} \) for \( i \neq j \)
Construction in [Lai, Deng, Li 2012]

Based on $e: G \times G \rightarrow G_T$, composite order $p_1p_2p_3p_4$ with $G_{p_1}$ as the main working group

**Secret Key** $g^{a+b^t}R$, $g^{t^t}R'$, $(H(\text{Value1})^tR_1, \ldots,$

**Ciphertext** $M \cdot e(g,g)^as$, $g^s$, $(g^{bs1H(“123..“)}Zh)^{r_1}Z_1$, $g^{r_1}Z_1'$, ...

where $g, h, u \in G_{p_1}$, $R, R' \in G_{p_3}$, $Z, Z_1, Z_1' \in G_{p_4}$,

$Z_i \in G_{p_4}$ are used to hide attribute values in ciphertext to prevent dictionary attack

**Orthogonality** property cancels effects of $Z_i$ in decryption
Summary

• Traditional access control to data relies on trusted servers

• Attribute-based Access control of encrypted data on untrusted server
  - Expressive policy and scalable (one-to-many encryption)
  - Fine-grained (attribute revocation)
  - Efficient (decryption outsource)
  - Privacy-preserved (policy hidden)
References


References


[Yang, Jia, Ren AsiaCCS’13] Kan Yang, Xiaohua Jia, Kui Ren: Attribute-based fine-grained access control with efficient revocation in cloud storage systems. ASIACCS 2013: 523-528


References


Acknowledgement of Contributors

Kan Yang
City University of Hong Kong

Prof. Robert H. Deng
Singapore Management University
Thank You!