Fine-grained Data Access Control Systems with User Accountability in Cloud Computing

Jin Li[1], Gansen Zhao[2], and Xiaofeng Chen [3]
Chunming Rong[4], Yong Tang[2]

1 Guangzhou University, China
2 South China Normal University
3 Xidian University
4 Stavenger University, Norway

Outline

- Background
- Our Approach
- Conclusion and Future Work

Background

- Cloud computing is an emerging computing paradigm in which IT resources and capacities are provided as services over the Internet.

- Challenging issues:

  Data Storage and *Access Control*

Background

- Security Concern: *Data access control*
  - ❒ Only authorized users can access data.
  - ❒ Different type of data should be accessible to different category of users, i.e., *fine-grained* data access control.

Background

- Challenges for achieving fine-grained access control
  - Strong attacks are possible
    -user collusion
    - key abuse
    - …
  - User/attribute add and revocation

Background

- Our Design Goals
  - Fine-grained access control over data stored in cloud computing
  - Collusion resistance
  - User accountability
- Design Tool
  - Attribute-based Encryption (ABE): one-to-many encryption, PKC

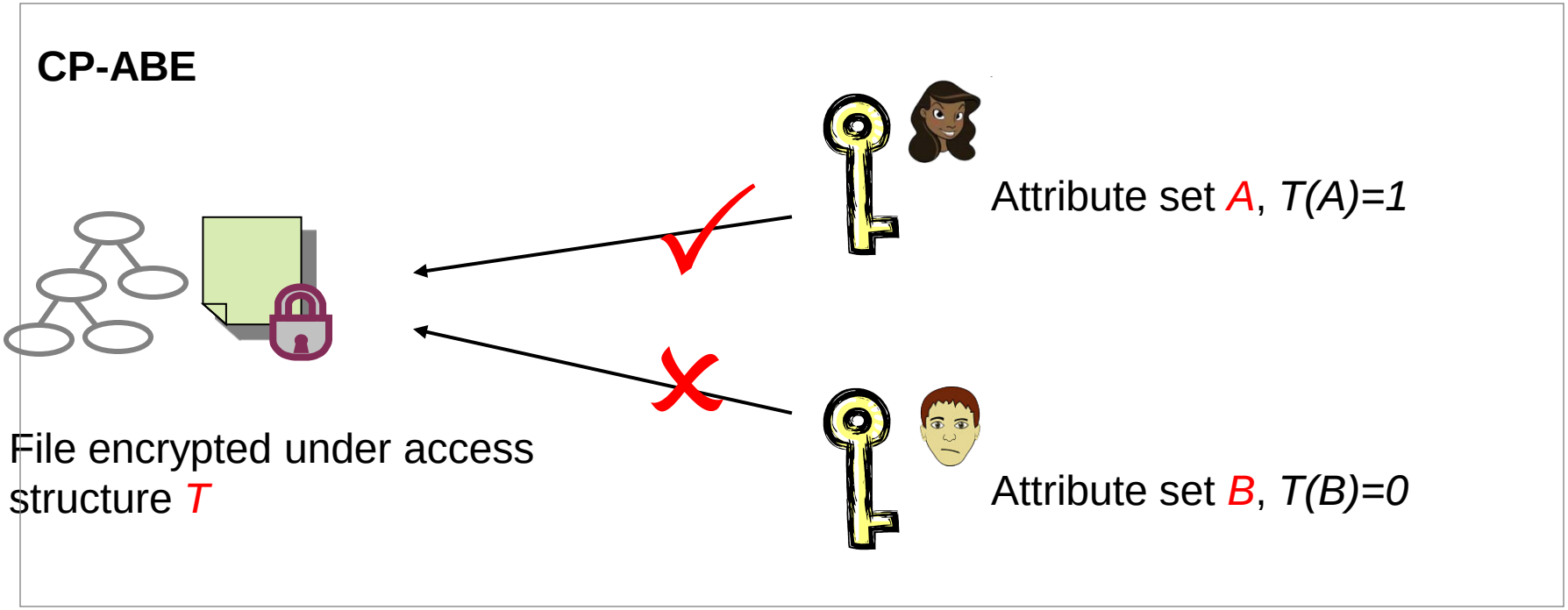
Problem Description

- Illegal key sharing among users
  - How to prevent users from sharing their attribute private keys?
    - Some users may have common attributes.

Outline

- Background
- Our Approach
- Conclusion And Future Work

- ABE is developed into two branches
  - Ciphertext Policy ABE (*CP-ABE*) and Key Policy ABE (*KP-ABE*)
  - Both are powerful tools for fine-grained access control



Idea

- Each user is associated with a set of attributes.
- User secret key is associated with an access structure.
- Data are encrypted over a set of attributes.
- Decrypt of data requires data attributes satisfy user access structure.

- Our observation
  - To detect illegal user, their identities IDs should be included in the private key of attribute list L.
  - There is no user ID information in the ciphertext.

Scheme Description

- **System Setup**
  - Public parameters as well as a master key for the attribute authority are chosen.
- **New User Grant**
  - Assume that the attributes of user ID are $L=(L_1, L_2, \ldots, L_k)$.
  - The authority computes the key for $L \| ID$ with the technique of hierarchical identity-based encryption, where ID is viewed as another default attribute.

- **New File Creation**

  - Assume that a message is encrypted with ciphertext-policy W.

  - The sender computes a ciphertext with policy W ‖*, such that any user with attribute list R(L,W)=1 can decrypt, regardless of the identity ID.

- **File Access**
  - Suppose that a message is encrypted with W ‖*.
  - Assume the user's secret key is for L ‖ ID, where R(L, W)=1.

    The user can only decrypt the ciphertext with attribute private key of L and the secret key of ID.

- **Trace**

  Suppose that a given pirate device can decrypt the ciphertext under ciphertext-policy W. To pinpoint the exact identity: The authority just computes the hash values of identities for all authorized users and find the identity if its hash value is the same with the one in the decryption key.

Main Idea in our construction

In normal encryption algorithm, a message is encrypted under ciphertext-policy $W=W' \| *$ such that any user with $L \| ID$ satisfying $R(L \| ID, W)=1$ is able to decrypt.

In tracing algorithm, the well-formed private key is extracted and pinpointed from the pirate device.

Main Idea in our construction

To achieve efficient user revocation, a broadcast encryption algorithm is used such that any revoked user cannot get the secret encrypted by the broadcast encryption.

Security Analysis

- Fine-grained access control
  - User access structure is able to describe sophisticated logics over attributes. We can enforce complex access policies.
- Collusion resistance
  - Each user’s secret key has a unique secret sharing scheme. Secret keys from different users do not “match” each other.

Outline

- Background
- Our Approach
- Conclusion And Future Work

- We presented a construction of ABE with user accountability.
- We showed how to use the ABE construction in Cloud computing to achieve access control.
- How to achieve more fine-grained access structure is our future work.

Thank You!