

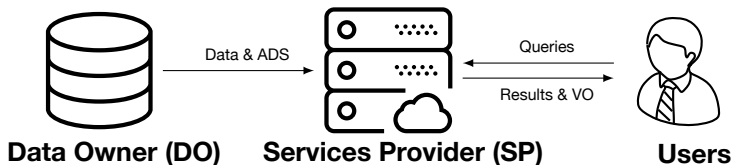
When Query Authentication Meets Fine-Grained Access Control: A Zero-Knowledge Approach

Cheng Xu¹, Jianliang Xu¹, Haibo Hu², and Man Ho Au²

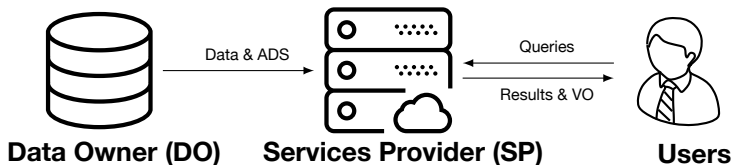
¹Hong Kong Baptist University ²Hong Kong Polytechnic University
{chengxu, xujl}@comp.hkbu.edu.hk {haibo.hu@, csallen@comp.}polyu.edu.hk

June 2018

- *Data-as-a-Service (DaaS)* and **cloud computing** are gaining popularity for big data analytics

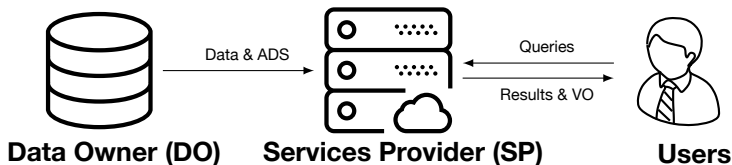


- *Data-as-a-Service (DaaS)* and *cloud computing* are gaining popularity for big data analytics

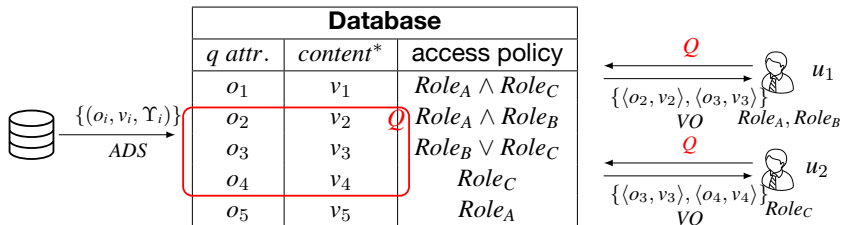


- *Fine-Grained Access Control*: enable big data sharing

- *Data-as-a-Service (DaaS)* and *cloud computing* are gaining popularity for big data analytics



- *Fine-Grained Access Control*: enable big data sharing
- *Security Threats*:
 - Query result integrity not guaranteed
 - Leaking information beyond query result may breach privacy



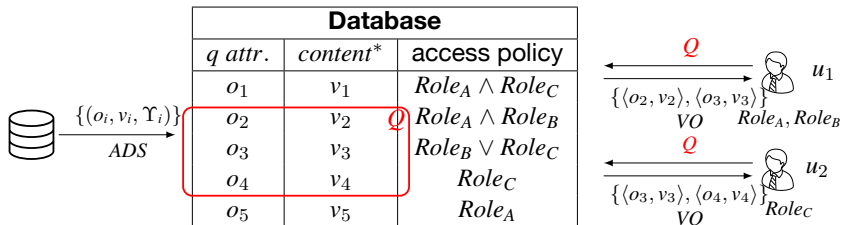
*Content is encrypted by CP-ABE

Data Owner (DO)

Service Provider (SP)

Users

- Fine-grained access policy as **monotone boolean function**



*Content is encrypted by CP-ABE

Data Owner (DO)

Service Provider (SP)

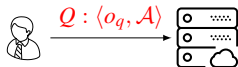
Users

- Fine-grained access policy as **monotone boolean function**
- Our solution:**
 - Integrity:** SP returns a *verification object* (VO) to prove
 - Soundness
 - Completeness
 - Zero-Knowledge Confidentiality:**

VO leaks no information beyond query results

- Develop a novel ABS-based APP signature
 - Authenticate accessible records
 - Prove inaccessibility in zero-knowledge
- Supported query types:
 - Equality queries
 - Range queries
 - Join queries
- Optimization techniques to reduce verification cost

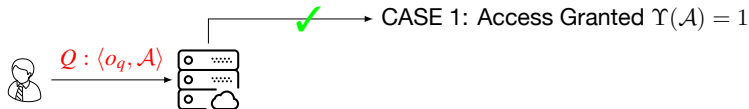
- Develop a novel ABS-based **APP signature**
 - Authenticate accessible records
 - Prove inaccessibility in zero-knowledge
- Supported query types:
 - **Equality queries**
 - Range queries
 - Join queries
- Optimization techniques to reduce verification cost



User

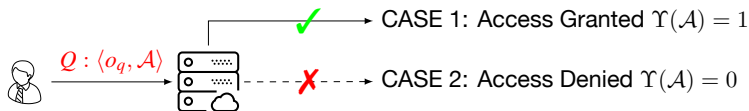
Service Provider

- User submits a query key o_q and a role set \mathcal{A}



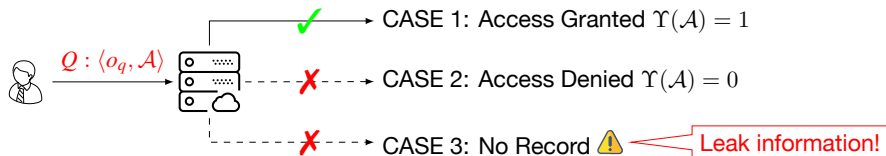
User	Service Provider	Outcomes
------	------------------	----------

- User submits a query key o_q and a role set \mathcal{A}



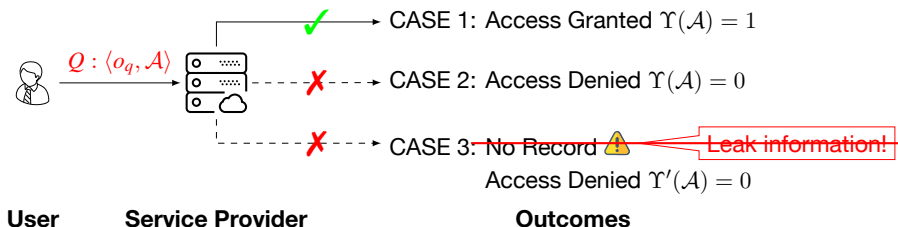
User	Service Provider	Outcomes
------	------------------	----------

- User submits a query key o_q and a role set \mathcal{A}



User	Service Provider	Outcomes
------	------------------	----------

- User submits a query key o_q and a role set \mathcal{A}
- Non-existent record will leak information

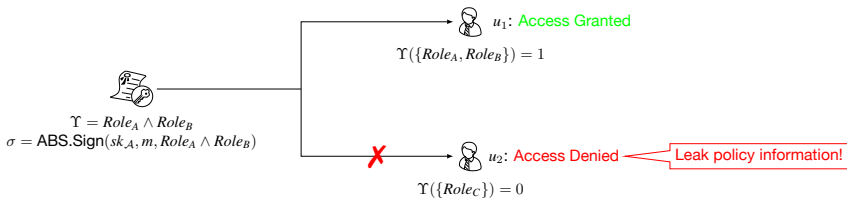


- User submits a query key o_q and a role set \mathcal{A}
- ~~Non-existent record will leak information~~
- Treat non-existent records as **inaccessible by anyone**
i.e. $\Upsilon' = Role_{\emptyset}$

ABS with Predicate Relaxation

- Attribute Based Signature (ABS)

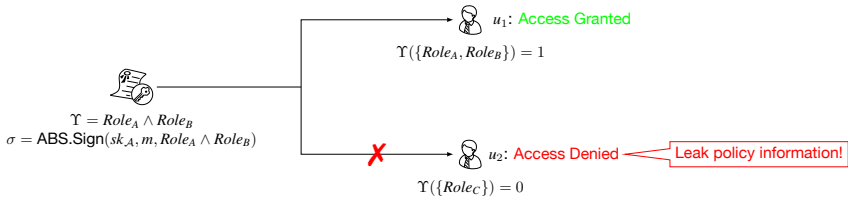
It signs a message with a monotone boolean function predicate that is satisfied by the attributes obtained from the authority



ABS with Predicate Relaxation

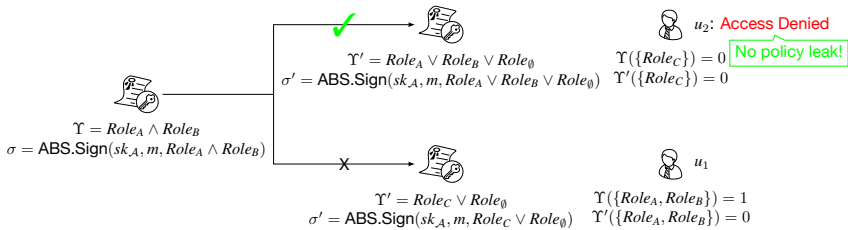
- Attribute Based Signature (ABS)

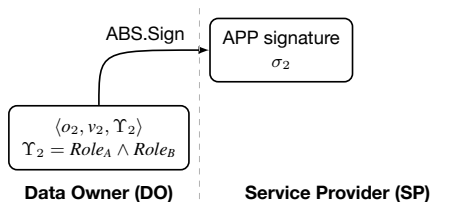
It signs a message with a monotone boolean function predicate that is satisfied by the attributes obtained from the authority



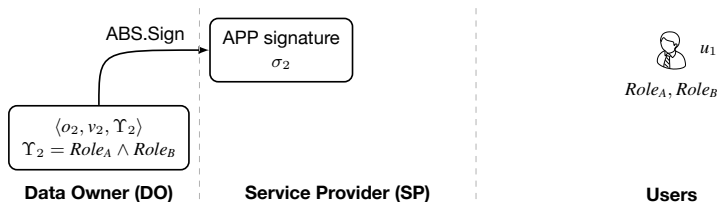
- Predicate Relaxation

Derive a **weaker** ABS signature without knowing secret key





- DO generates ADS and sends to the SP



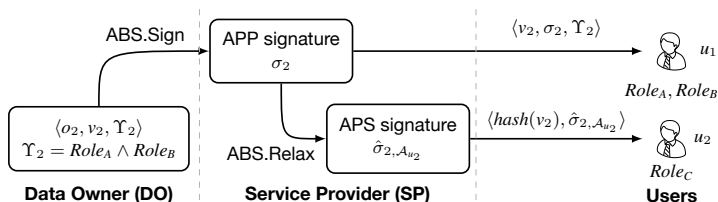
- DO generates ADS and sends to the SP
- u_1 can access the data,



- DO generates ADS and sends to the SP
- u_1 can access the data, APP signature is the VO



- DO generates ADS and sends to the SP
- u_1 can access the data, APP signature is the VO
- u_2 cannot access the data,



- DO generates ADS and sends to the SP
- u_1 can access the data, APP signature is the VO
- u_2 cannot access the data, SP generates an APS signature as VO

- Access-Policy-Preserving (APP) signature
 - Signed by DO and used as ADS
 - It captures three parts of information:
query attribute o_i , data content v_i , and access policy Υ_i

Example 1

$$\text{Record}_2 \leftarrow \langle o_2, v_2, \Upsilon_2 = \text{Role}_A \wedge \text{Role}_B \rangle$$
$$\sigma_2 \leftarrow \text{ABS.Sign}(sk_{\text{DO}}, \text{hash}(o_2) \parallel \text{hash}(v_2), \text{Role}_A \wedge \text{Role}_B)$$

Authenticated Data Structures (ADS)

- Access-Policy-Preserving (APP) signature
 - Signed by DO and used as ADS
 - It captures three parts of information:
query attribute o_i , data content v_i , and access policy Υ_i

Example 1

$$\begin{aligned} \text{Record}_2 &\leftarrow \langle o_2, v_2, \Upsilon_2 = \text{Role}_A \wedge \text{Role}_B \rangle \\ \sigma_2 &\leftarrow \text{ABS.Sign}(sk_{\text{DO}}, \text{hash}(o_2) | \text{hash}(v_2), \text{Role}_A \wedge \text{Role}_B) \end{aligned}$$

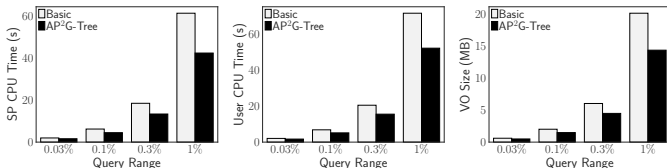
- Access-Policy-Stripped (APS) signature
 - Replace Υ_i to $\hat{\Upsilon}_{\mathcal{A}} = a_1 \vee a_2 \vee \dots \vee a_n, a_i \in \mathbb{A} \setminus \mathcal{A}$
 - Be used to prove inaccessibility in zero-knowledge

Example 2

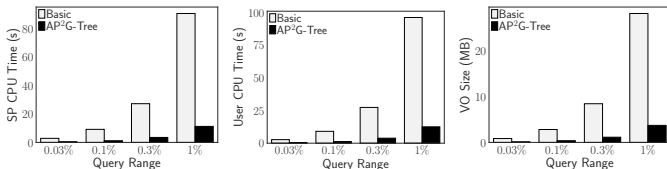
$$\begin{aligned} \mathbb{A} &= \{\text{Role}_A, \text{Role}_B, \text{Role}_C, \text{Role}_\emptyset\}, \hat{\Upsilon}_{\{\text{Role}_C\}} = \text{Role}_A \vee \text{Role}_B \vee \text{Role}_\emptyset \\ \hat{\sigma}_2 &\leftarrow \text{ABS.Sign}(sk_{\text{DO}}, \text{hash}(o_2) | \text{hash}(v_2), \text{Role}_A \vee \text{Role}_B \vee \text{Role}_\emptyset) \end{aligned}$$

Performance Evaluation

- TPC-H dataset (1 800 000 records)
- 10 distinct policies (10 global roles, max policy length is 6)



Range Query Performance vs. Range



Join Query Performance vs. Range

Thanks

Q&A