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

Cheng Xu[1], Jianliang Xu[1], Haibo Hu[2], and Man Ho Au[2]

[1]Hong Kong Baptist University
{chengxu, xujl}@comp.hkbu.edu.hk

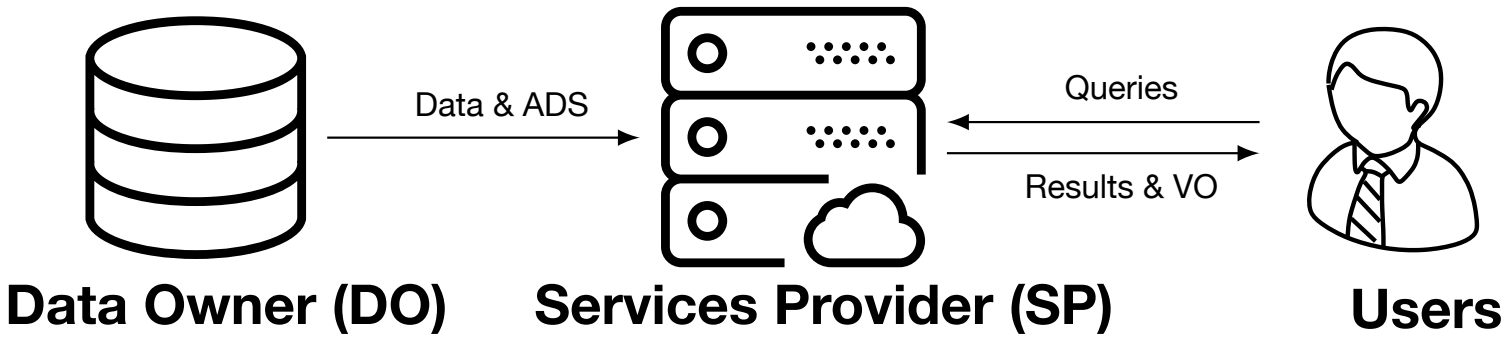
[2]Hong Kong Polytechnic University
{haibo.hu@, csallen@comp.}polyu.edu.hk

June 2018

Background

HKBU Database Group

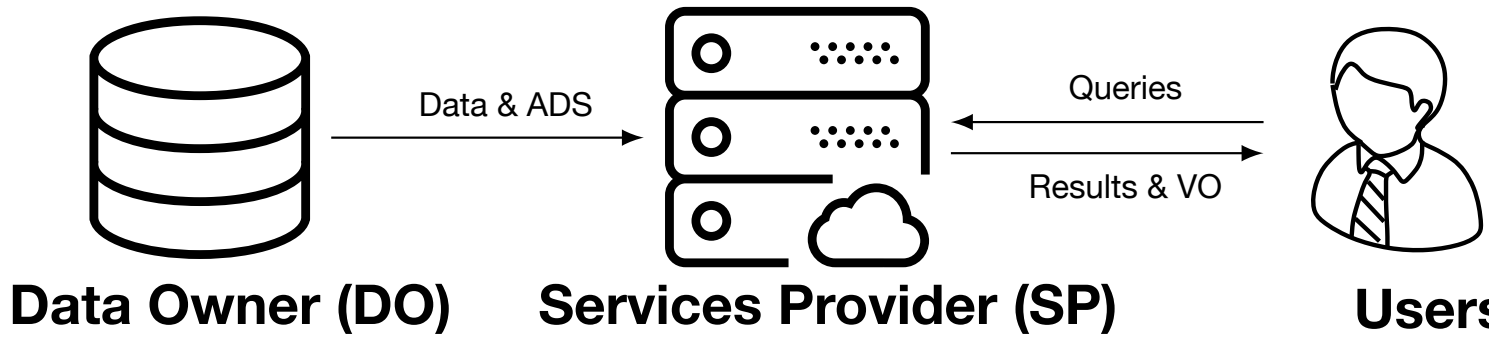
- *Data-as-a-Service* (DaaS) and cloud computing are gaining popularity for big data analytics.



Background

HKBU Database Group

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

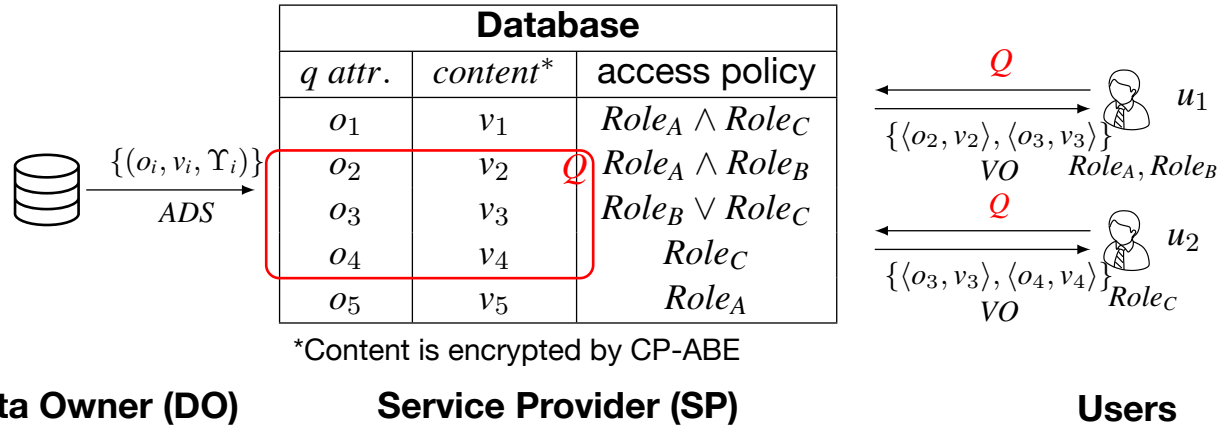


- Two challenges raised:

  Integrity *Service Provider* (SP) may be malicious.

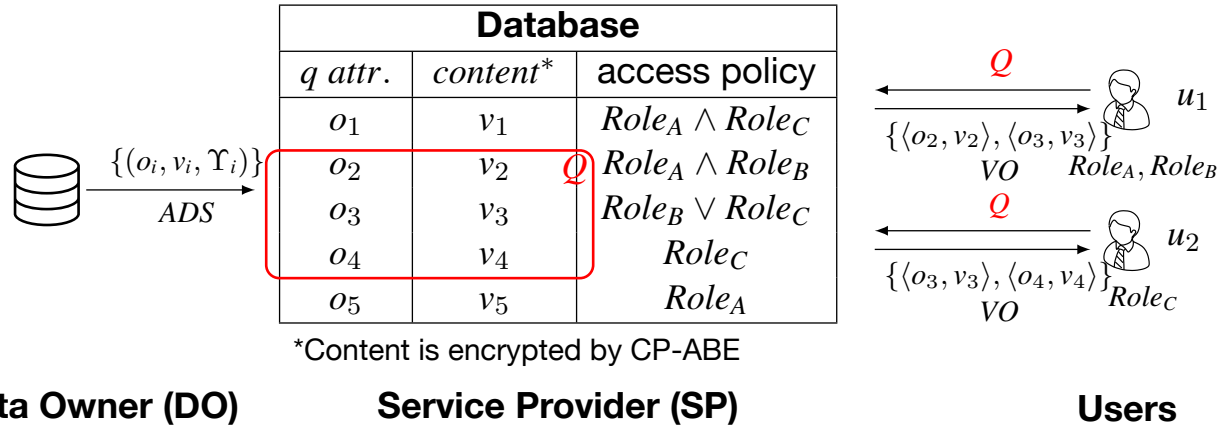
  Confidentiality *Data Owner* (DO) may want to enforce fine-grained access control on the database.

Problem Model



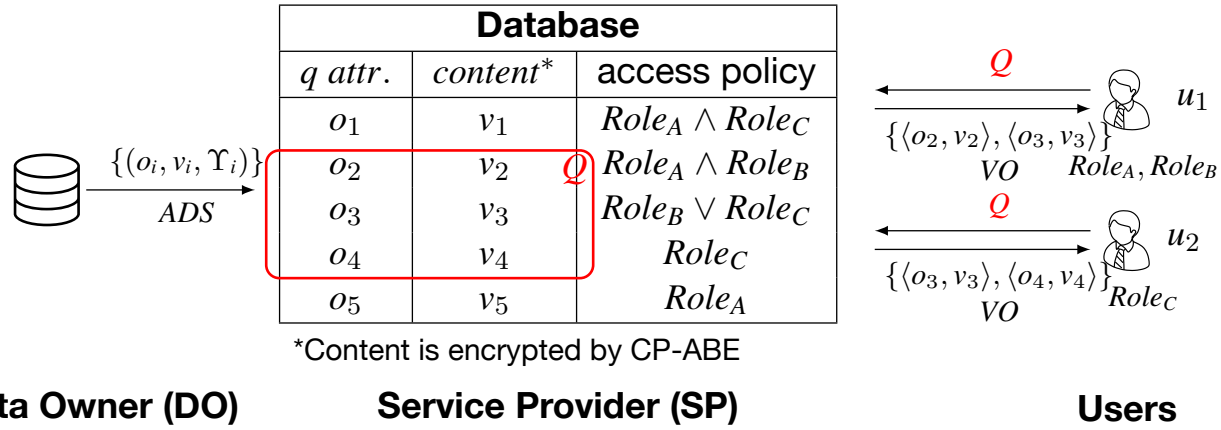
- Fine-grained access policy as monotone boolean function.

Problem Model



- Fine-grained access policy as monotone boolean function.
- Integrity: SP returns a *verification object* (VO) to prove soundness and completeness of query results.

Problem Model



- Fine-grained access policy as monotone boolean function.
- Integrity: SP returns a *verification object* (VO) to prove soundness and completeness of query results.
- Zero-Knowledge Confidentiality:
  VO leaks no information beyond query results.

Our Contributions

HKBU Database Group

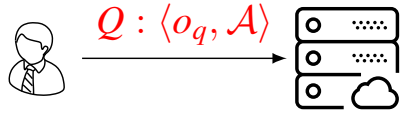
- Develop a novel ABS-based APP signature as ADS.
  - Authenticate accessible records.
  - Prove inaccessibility in zero-knowledge.
- Supported Query Types:
  - Equality queries.
  - Range queries.
  - Join queries.
- Optimization techniques to reduce verification cost.

Our Contributions

HKBU Database Group

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

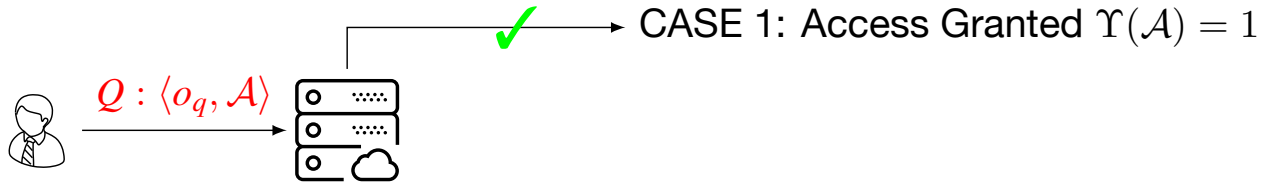
Equality Query



User **Service Provider**

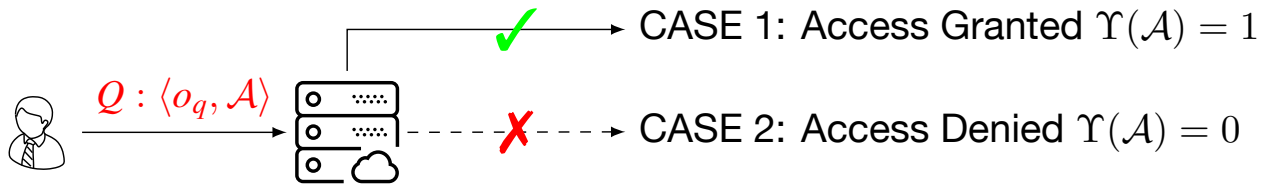
- User submits a query key o_q and a role set $\mathcal{A}$.

Equality Query



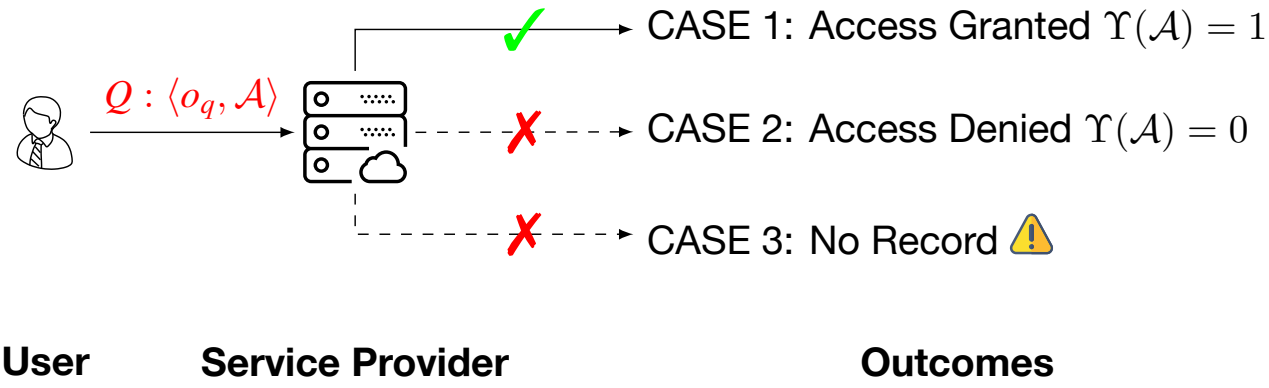
- User submits a query key o_q and a role set $\mathcal{A}$.

Equality Query



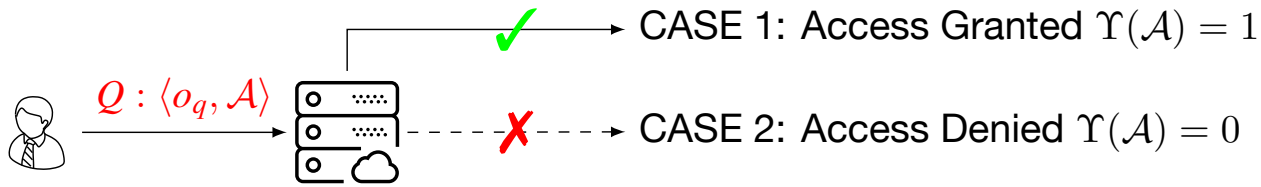
- User submits a query key o_q and a role set $\mathcal{A}$.

Equality Query



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

Equality Query

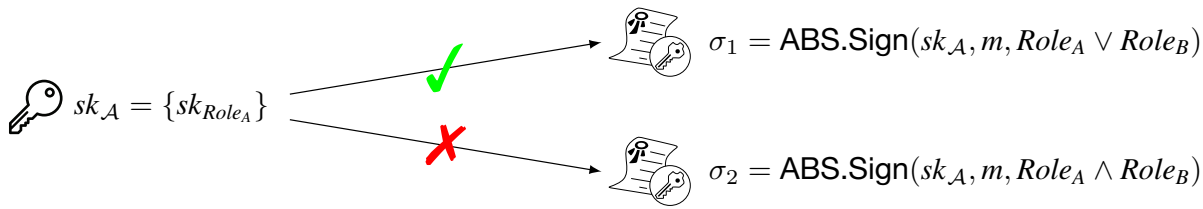


- User submits a query key o_q and a role set $\mathcal{A}$.
- ~~Non-existent record will leak information.~~
- Treat non-existent records as inaccessible by anyone. i.e. $\Upsilon = Role_{\emptyset}$.

ABS with Predicate Relaxation

HKBU Database Group

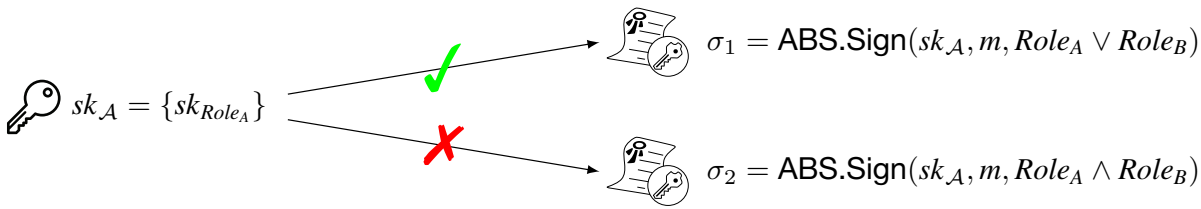
- Attribute Based Signature (ABS)
  It signs a message with a monotone boolean function predicate that is satisfied by the attributes obtained from the authority.



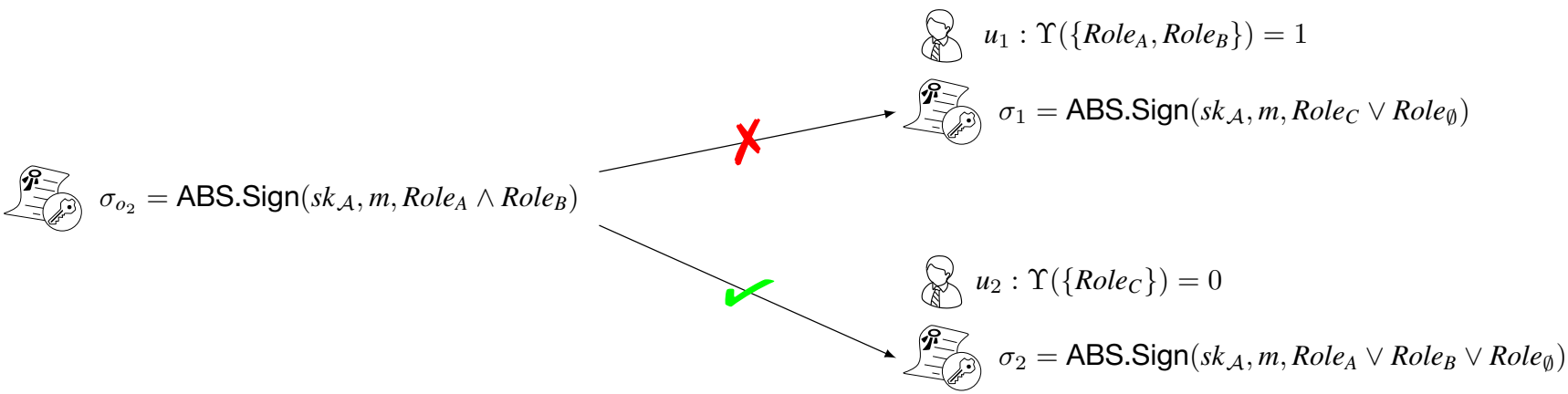
ABS with Predicate Relaxation

HKBU Database Group

- 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.



Authenticated Data Structures (ADS)

HKBU Database Group

- 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

$$\text{Record}_2 \leftarrow \langle o_2, v_2, \Upsilon_2 = Role_A \wedge Role_B \rangle$$

$$\sigma_2 \leftarrow \text{ABS.Sign}(sk_{\text{DO}}, hash(o_2)|hash(v_2), Role_A \wedge Role_B)$$

Authenticated Data Structures (ADS)

HKBU Database Group

- 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

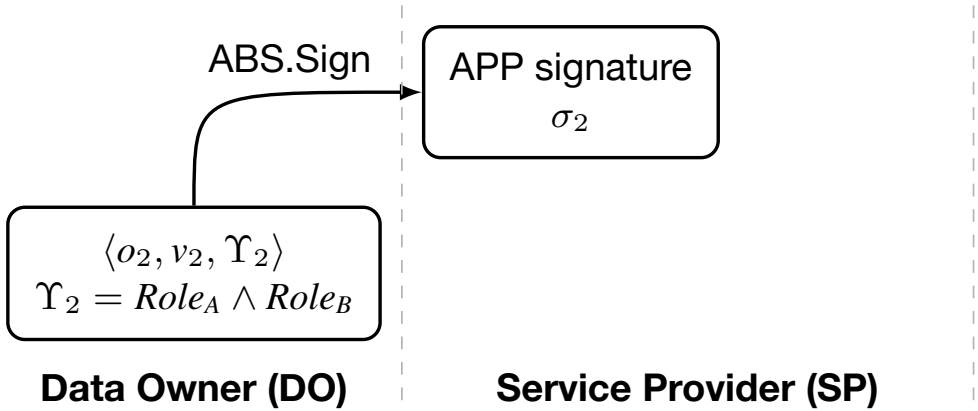
$$\text{Record}_2 \leftarrow \langle o_2, v_2, \Upsilon_2 = Role_A \wedge Role_B \rangle$$
$$\sigma_2 \leftarrow \text{ABS.Sign}(sk_{\text{DO}}, hash(o_2)|hash(v_2), Role_A \wedge Role_B)$$

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

Example 2

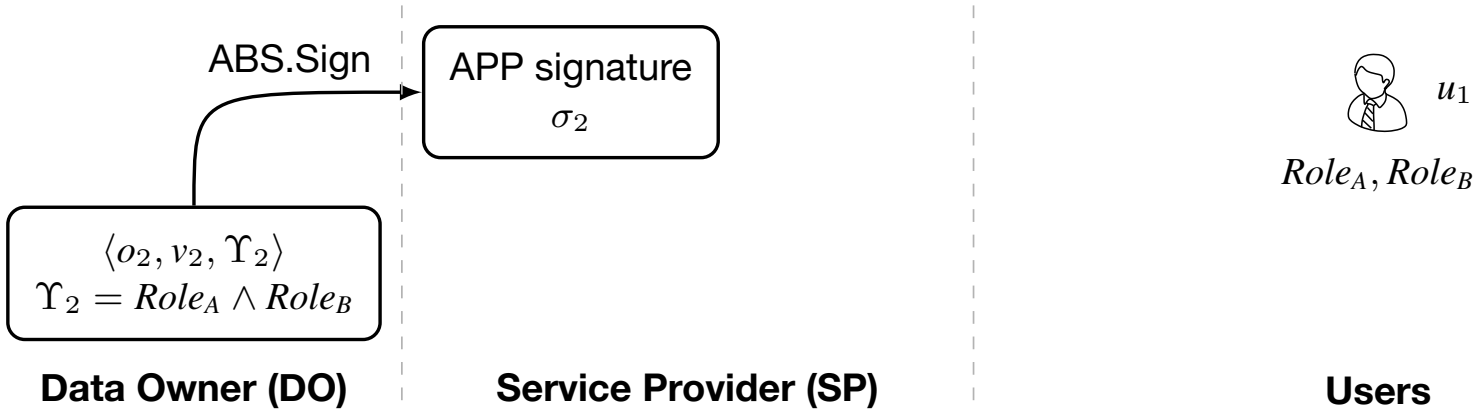
$$\mathbb{A} = \{Role_A, Role_B, Role_C, Role_\emptyset\}, \hat{\Upsilon}_{\{Role_C\}} = Role_A \vee Role_B \vee Role_\emptyset$$
$$\hat{\sigma}_2 \leftarrow \text{ABS.Sign}(sk_{\text{DO}}, hash(o_2)|hash(v_2), Role_A \vee Role_B \vee Role_\emptyset)$$

Query Processing



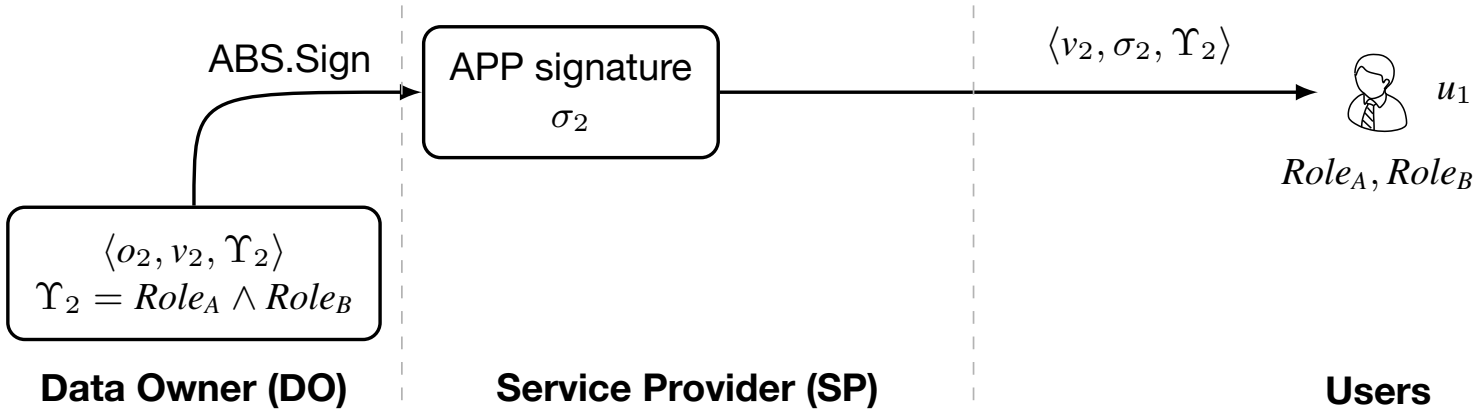
- DO generates ADS and sends to the SP.

Query Processing



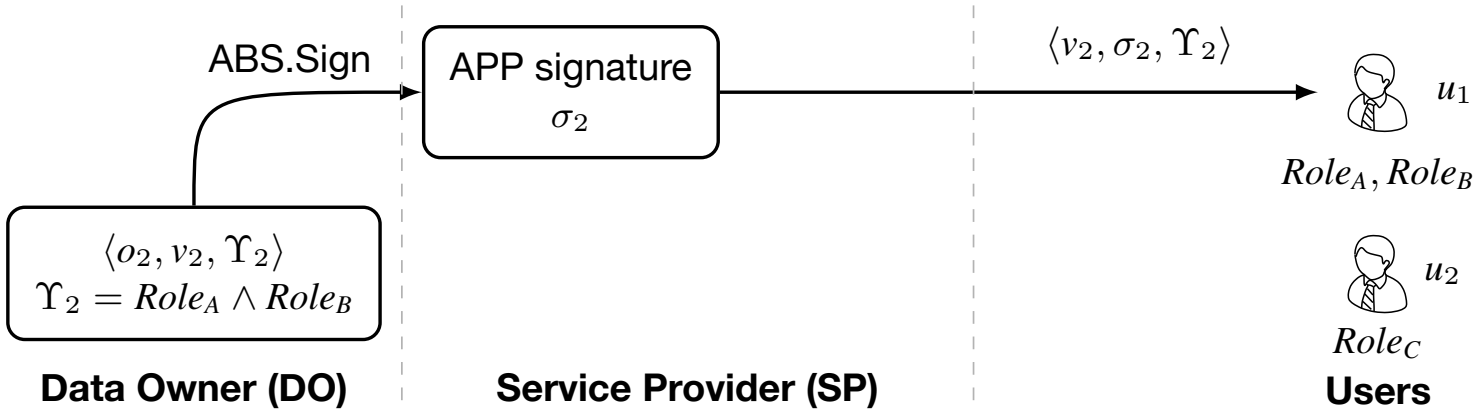
- DO generates ADS and sends to the SP.
- u_1 can access the data,

Query Processing



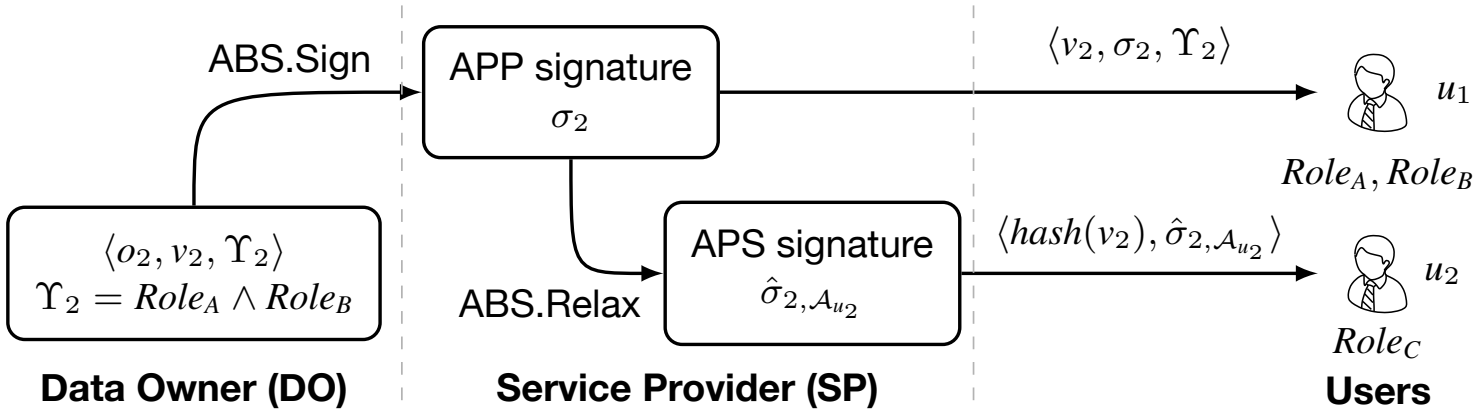
- DO generates ADS and sends to the SP.
- u_1 can access the data, APP signature is the VO.

Query Processing



- 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,

Query Processing

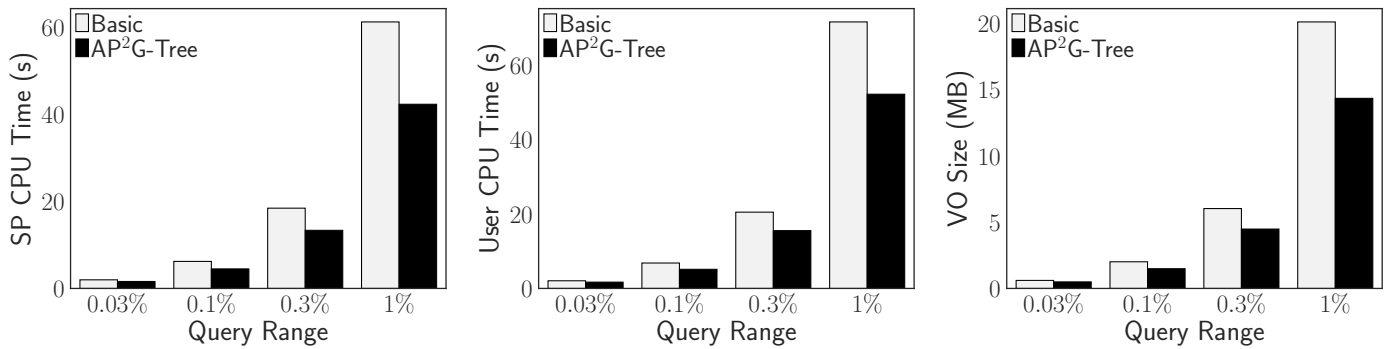


- DO generates ADS and sends to the SP.
- u_1 can access the data, APP signature is the VO.
- u_2 cannot access the data, SP generates an APS signature as VO.

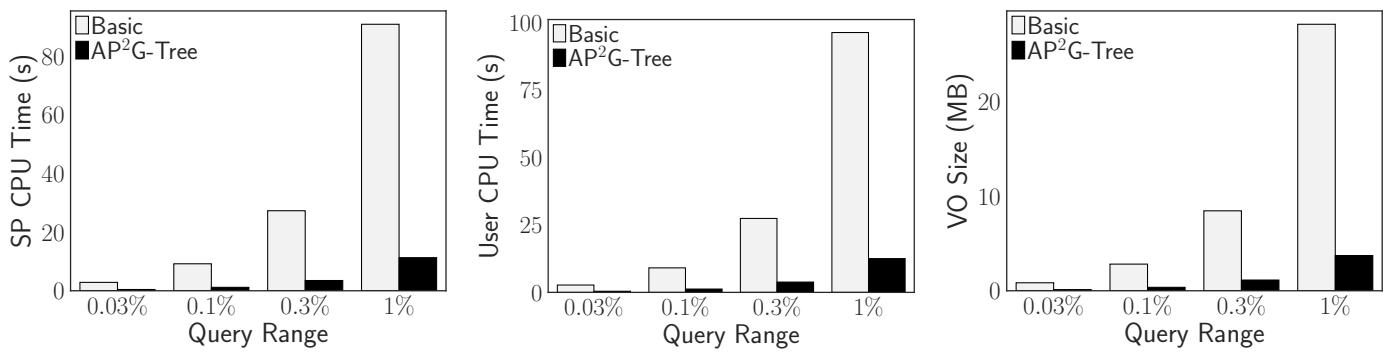
Performance Evaluation

HKBU Database Group

- 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