EURECOM
Sophia Antipolis

Multi-User Searchable Encryption in the Cloud

Cédric Van Rompay, Refik Molva, and Melek Önen

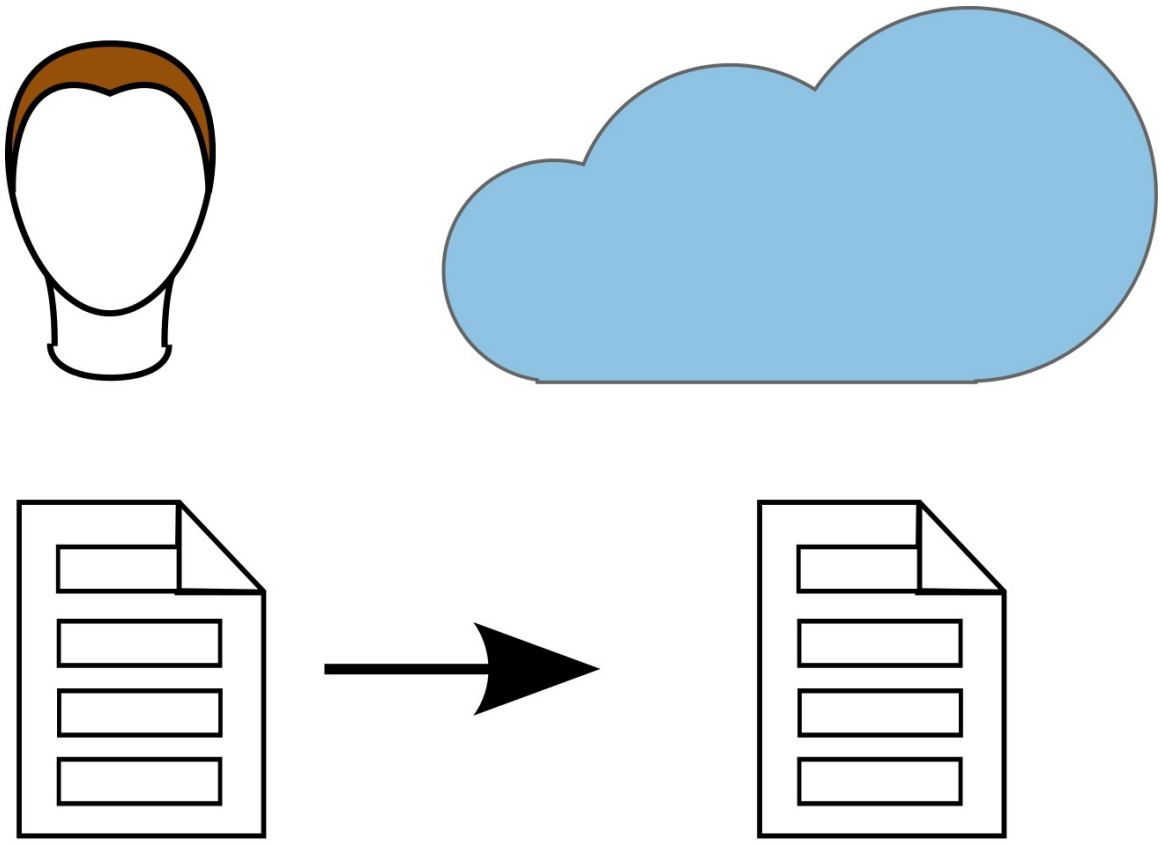
ISC 2015

September 10th, 2015 | Trondheim, Norway

Outline

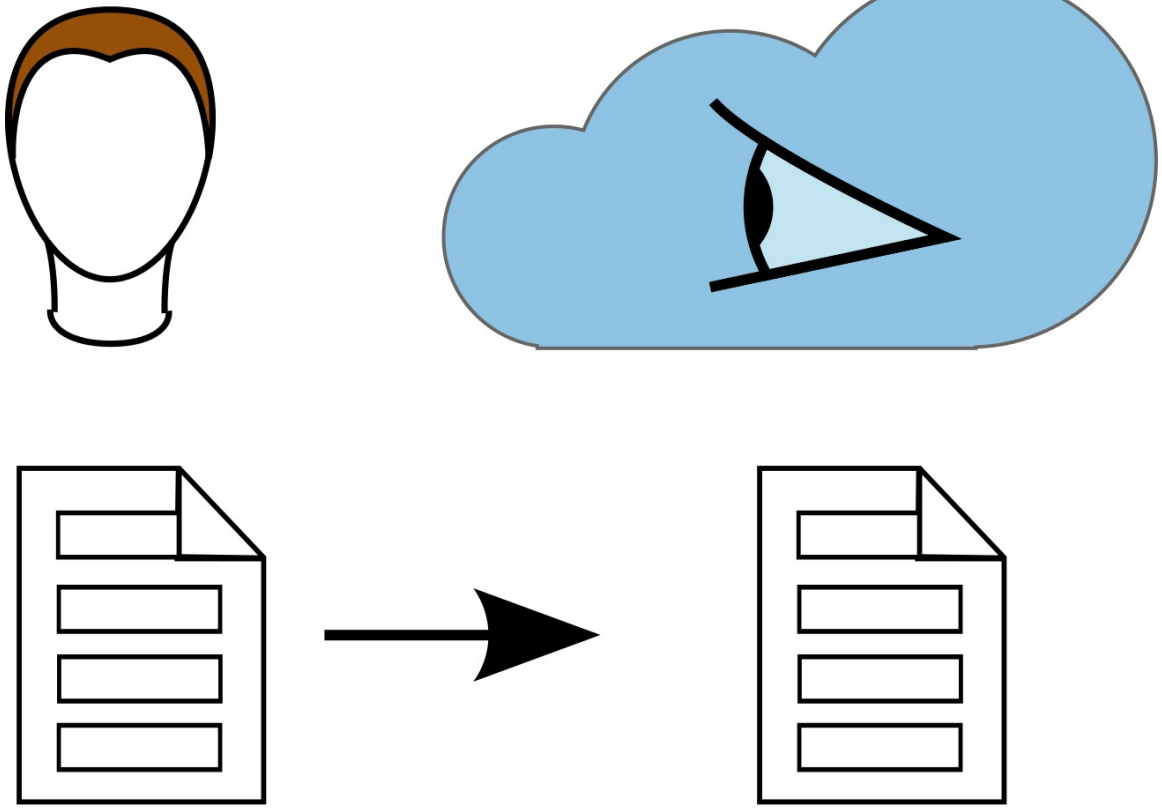
- Searchable Encryption
- Multi-User Scenarios
- State of the Art
- Privacy Issue
- Challenge
- Solution
- Conclusion

Searchable Encryption



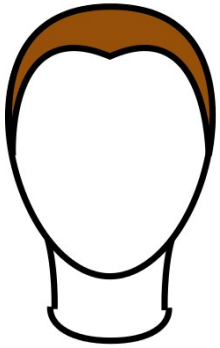
- Data upload

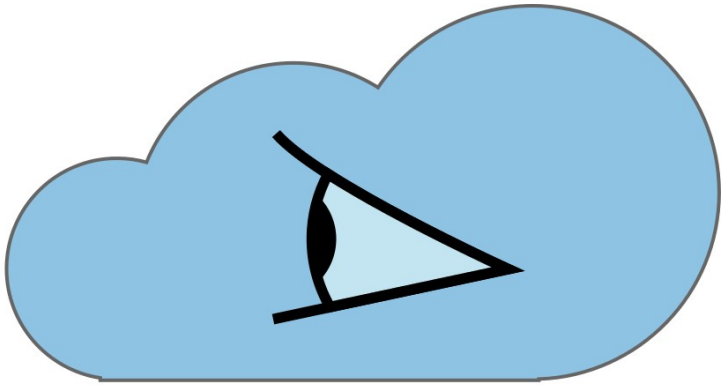
Searchable Encryption



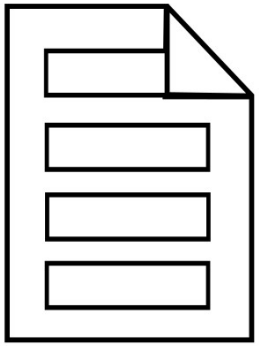
- Data upload
- Honest-but-curious Cloud

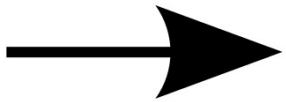
Searchable Encryption



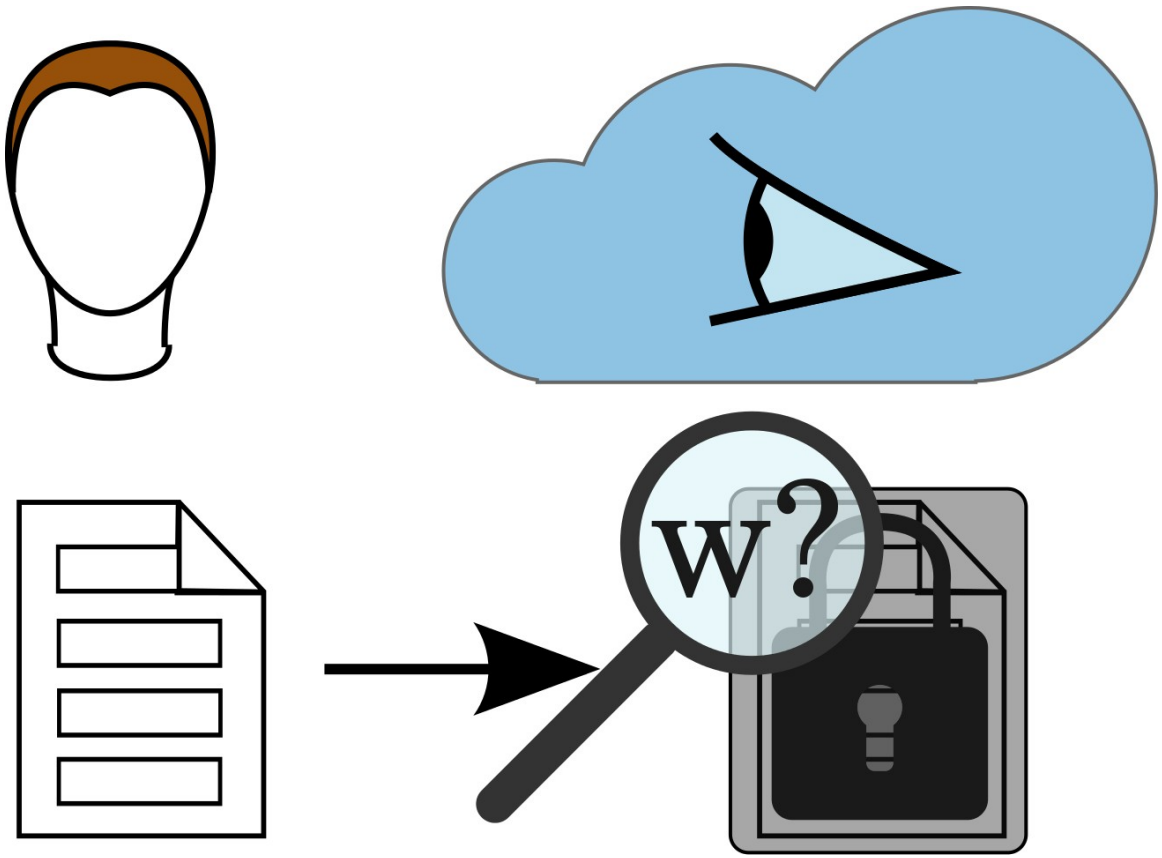


- Data upload
- Honest-but-curious Cloud
- Encryption for Privacy





Searchable Encryption

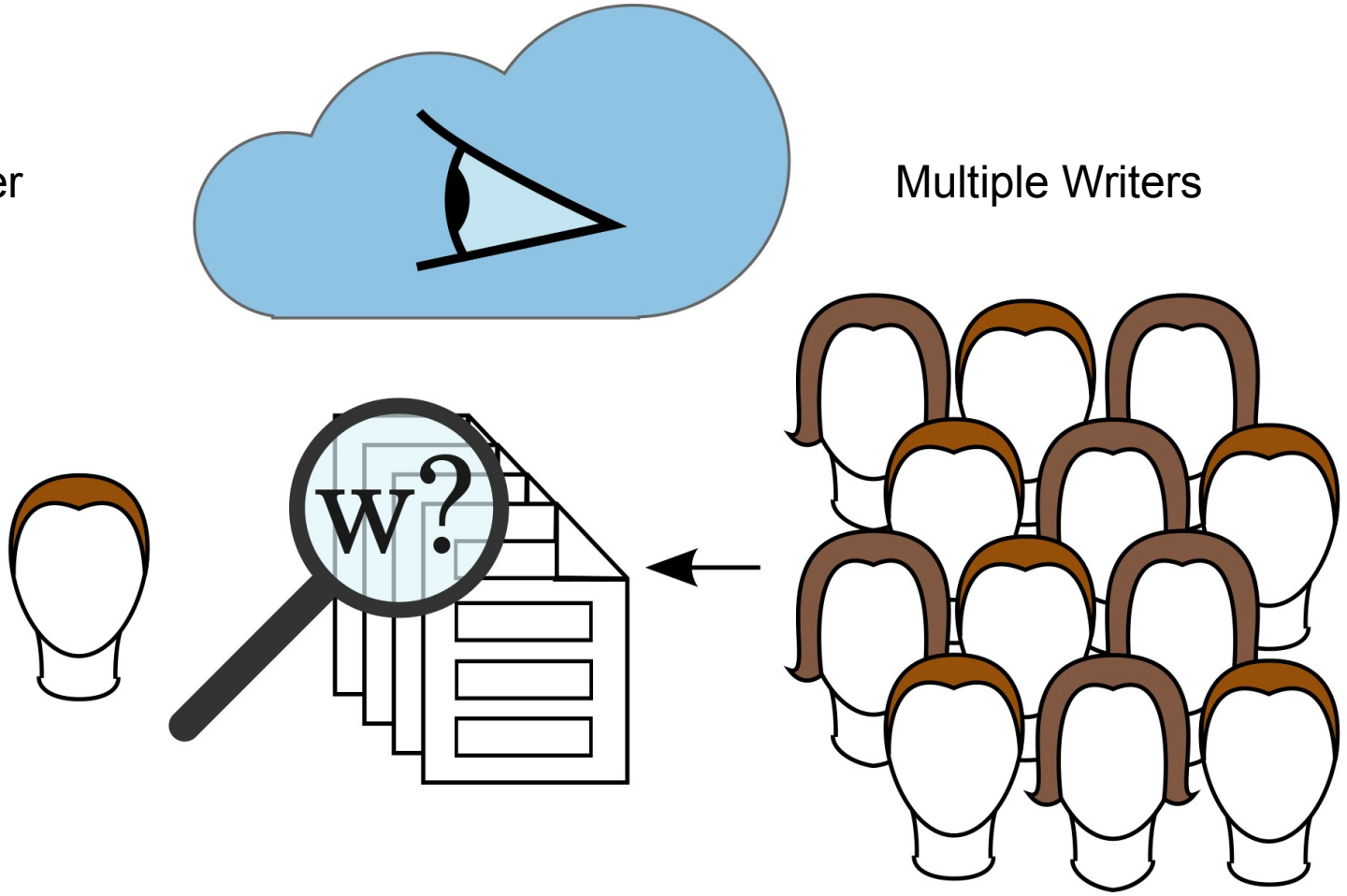


- Data upload
- Honest-but-curious Cloud
- Encryption for Privacy
- Remote word search queries

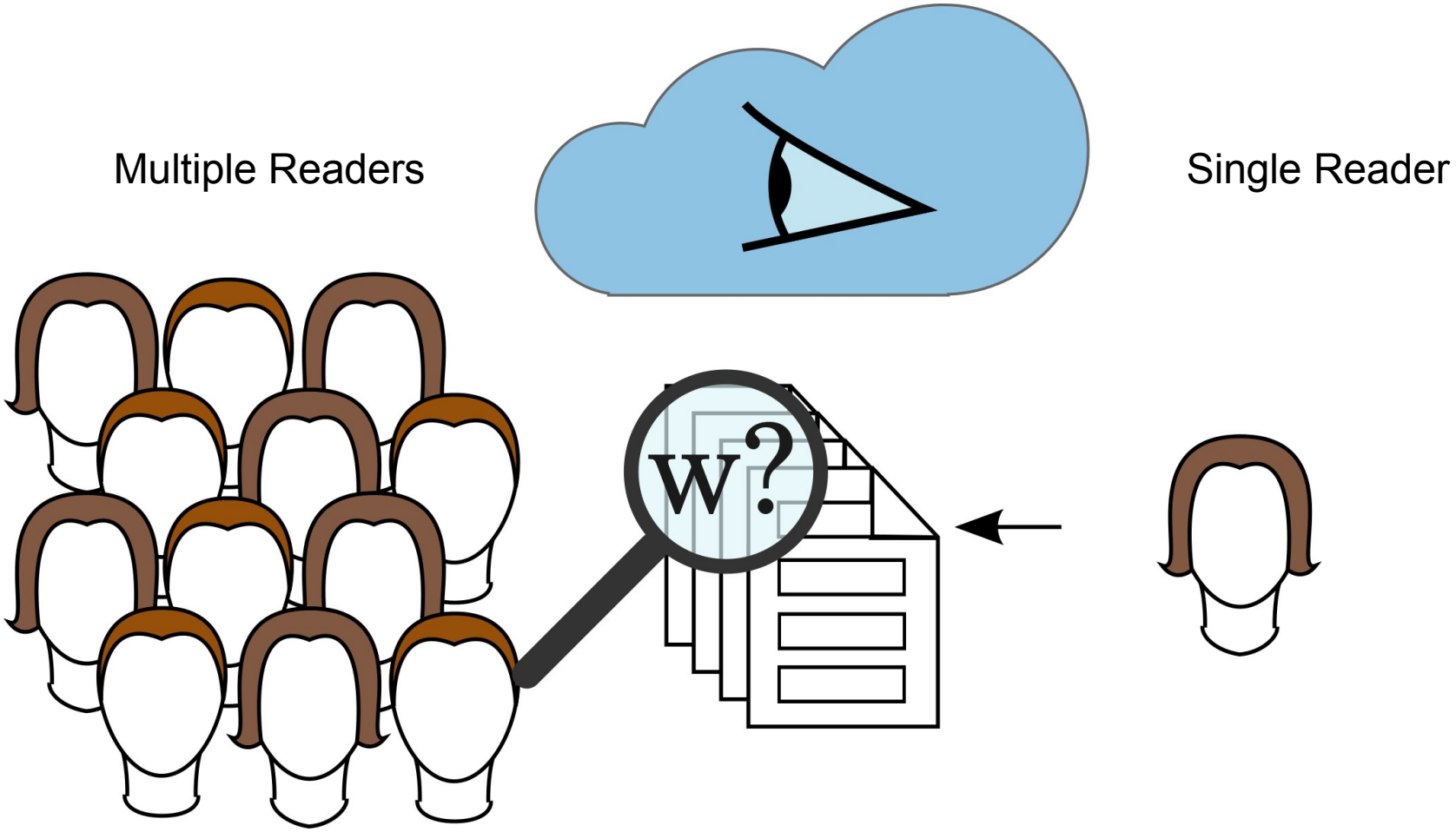
Multi-User Scenarios - PEKS

Single Reader

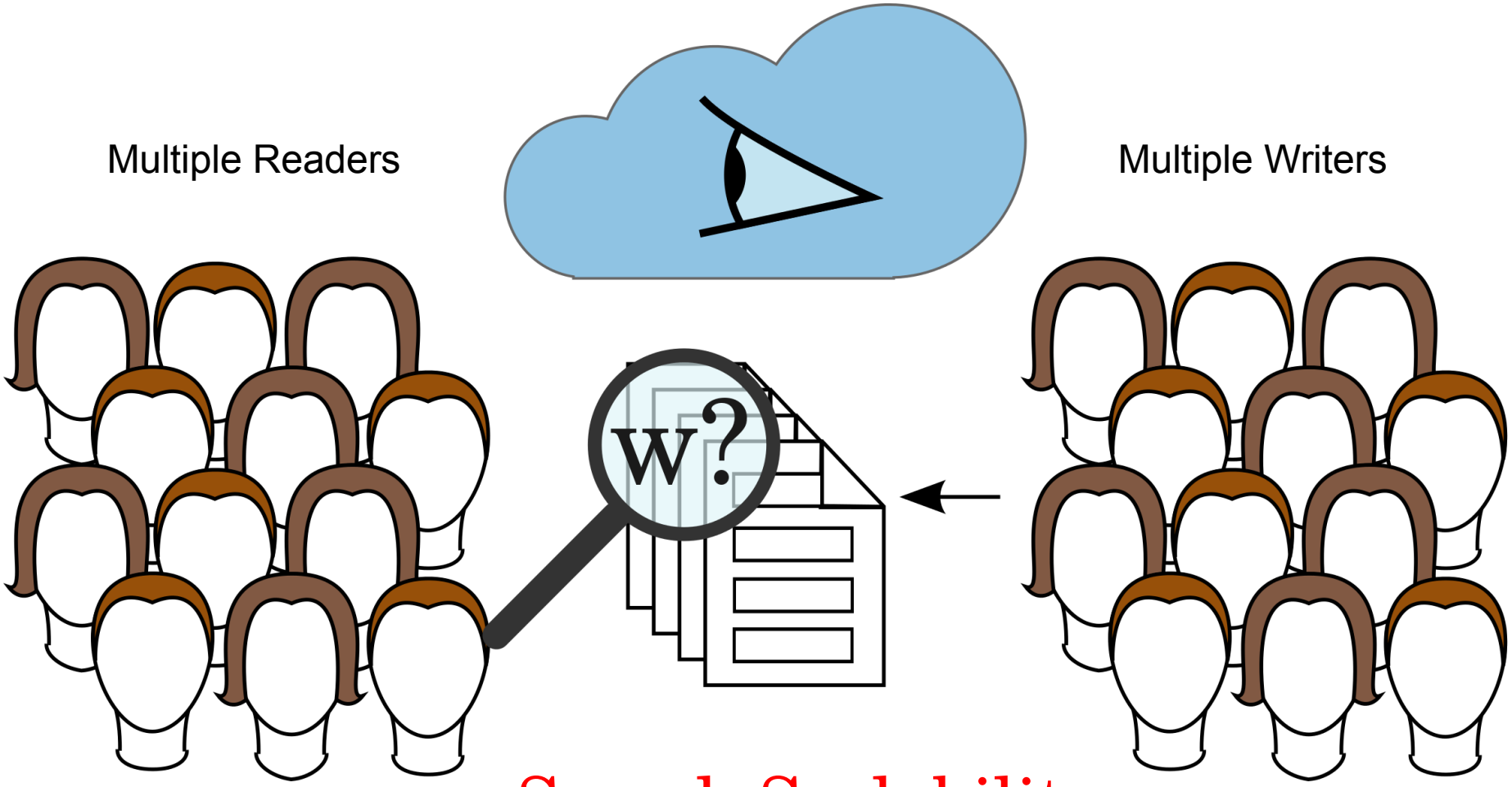
Multiple Writers



Multi-User Scenarios – Delegated SE



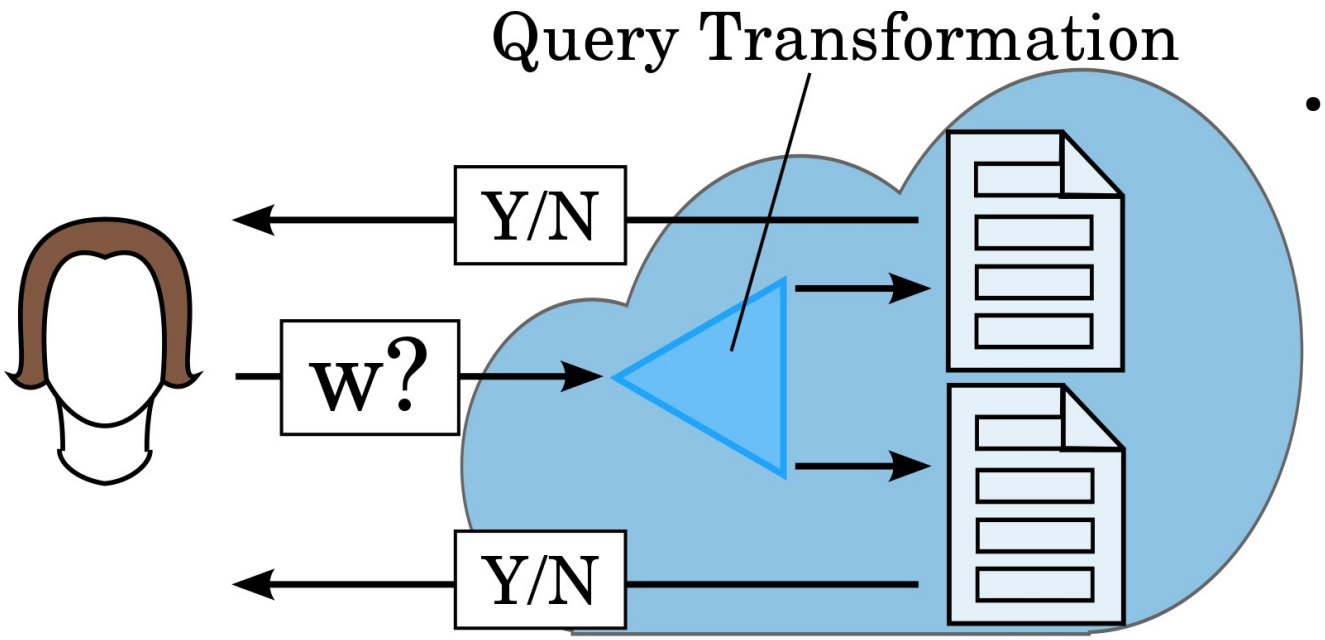
Multi-User Scenarios – MUSE



+ Search Scalability

State of the Art for MUSE

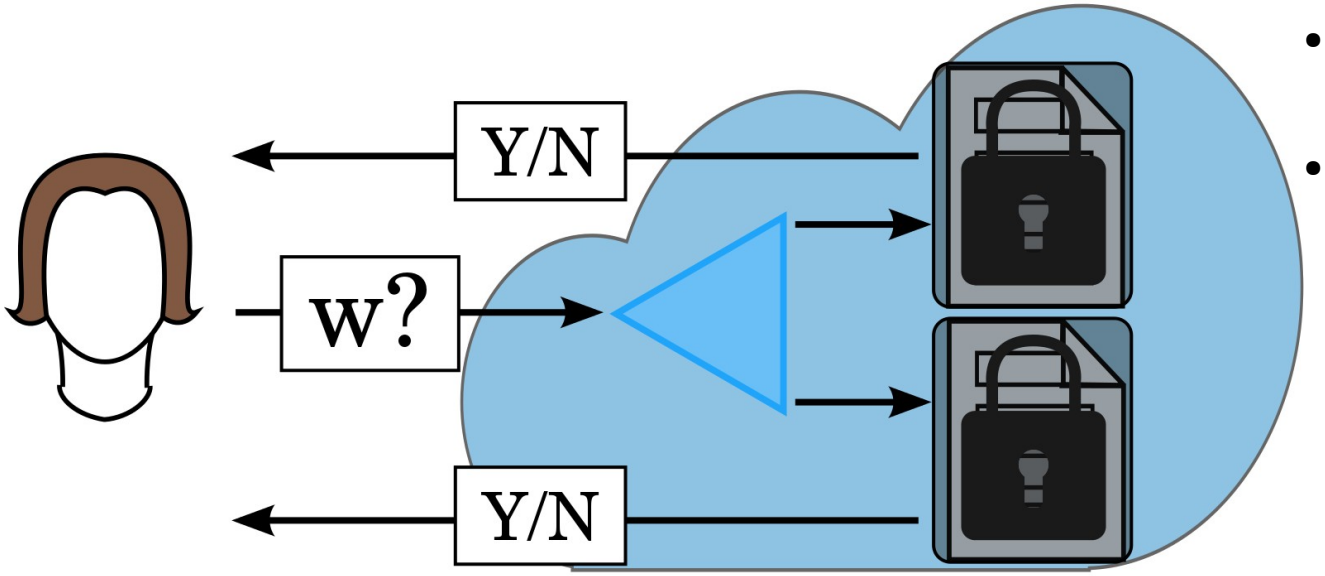
[Bao2008], [Yang2011], [Popa2014]



- **Scalability** through query transformation

State of the Art for MUSE

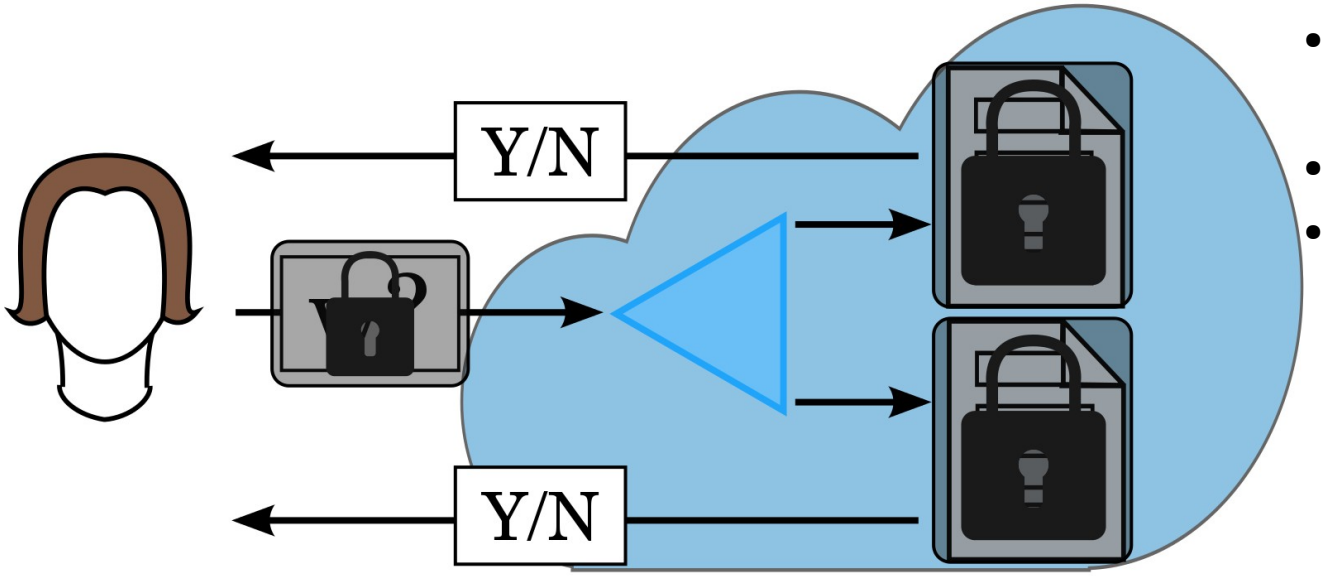
[Bao2008], [Yang2011], [Popa2014]



- **Scalability** through query transformation
- **Data Privacy**

State of the Art for MUSE

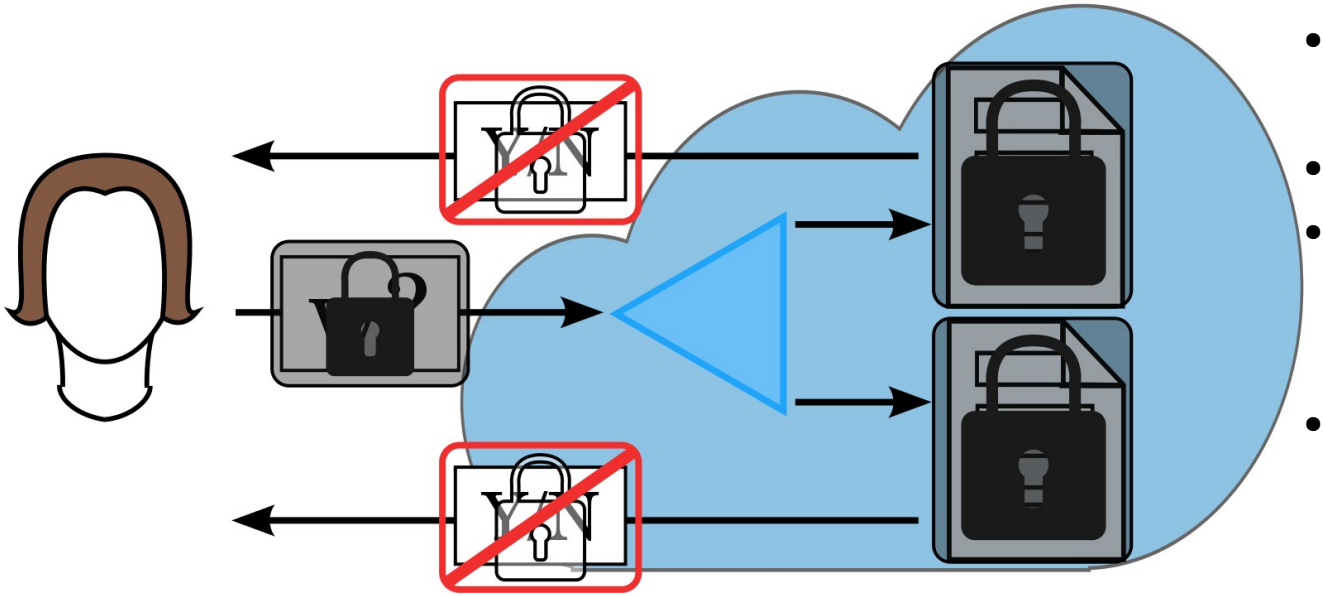
[Bao2008], [Yang2011], [Popa2014]



- **Scalability** through query transformation
- **Data Privacy**
- **Search Pattern Privacy** (query indistinguishability)

State of the Art for MUSE

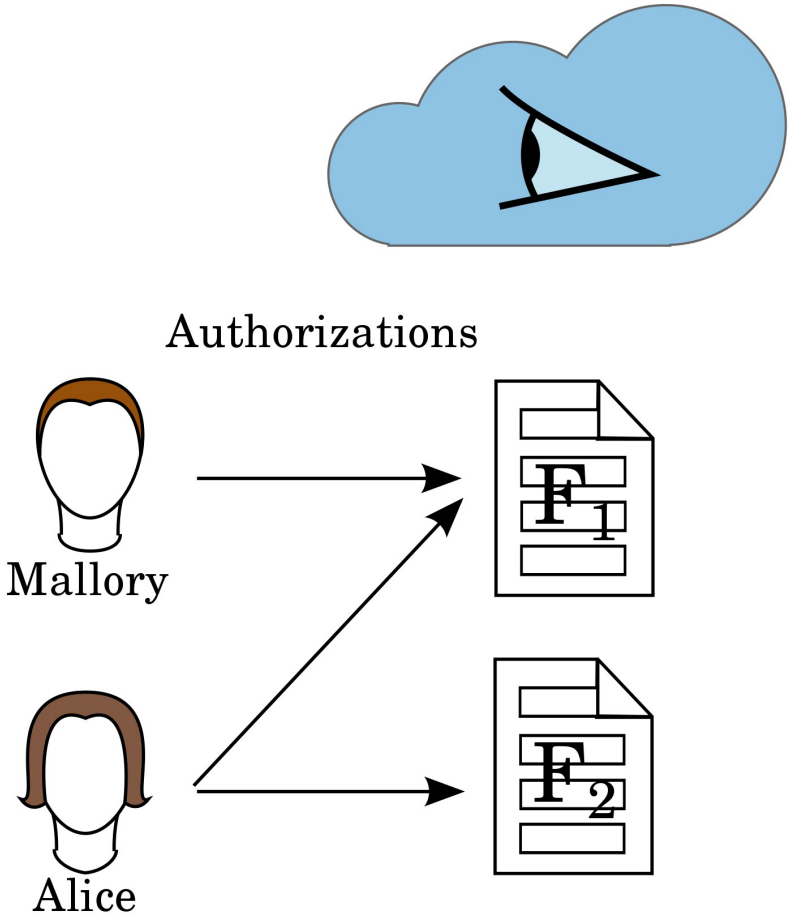
[Bao2008], [Yang2011], [Popa2014]



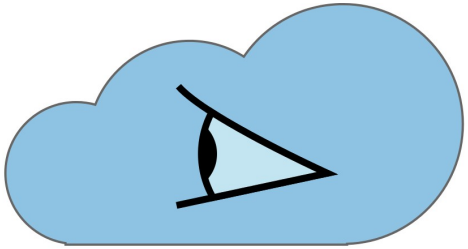
- **Scalability** through query transformation
- **Data Privacy**
- **Search Pattern Privacy** (query indistinguishability)
- **No Access Pattern Privacy** (result privacy)

➡ Privacy Issue if collusions

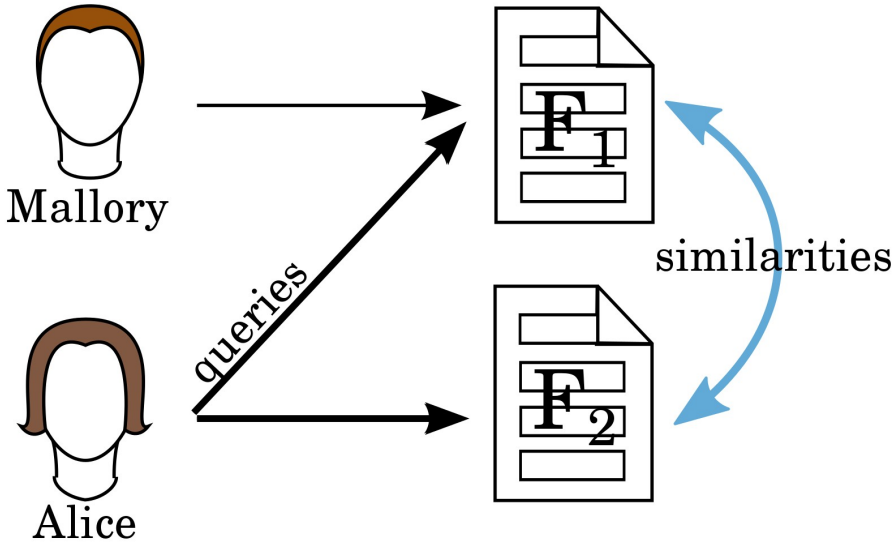
Privacy Issue



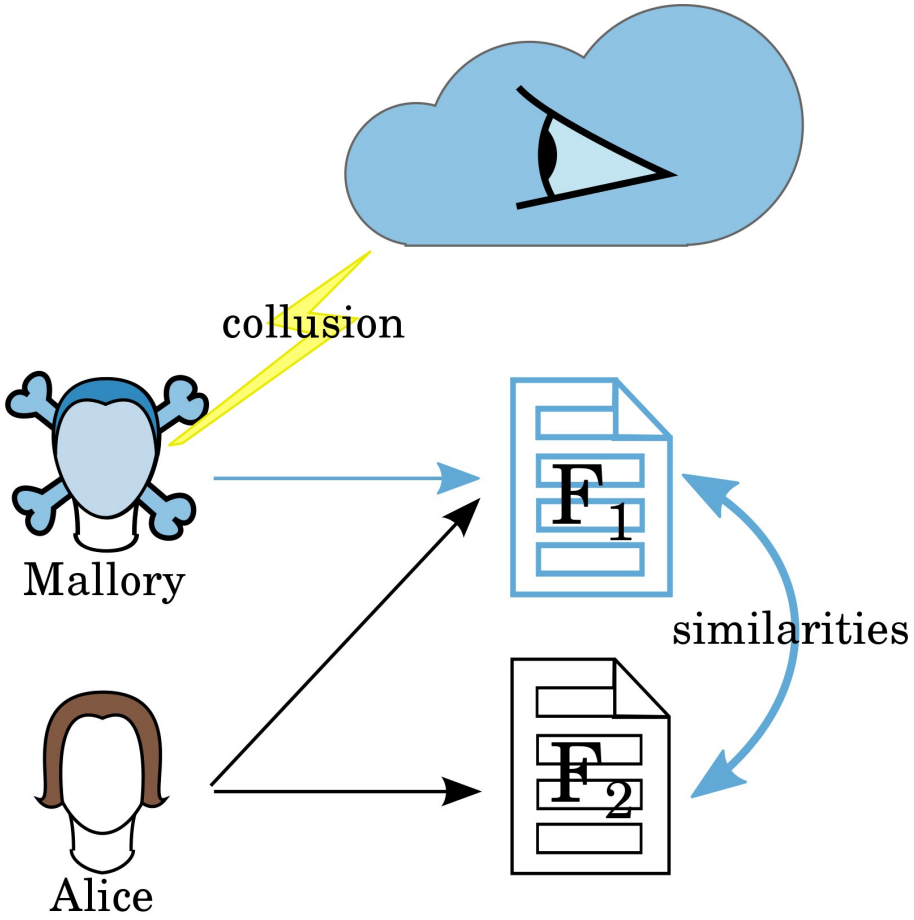
Privacy Issue



- Access Pattern➡Similarities

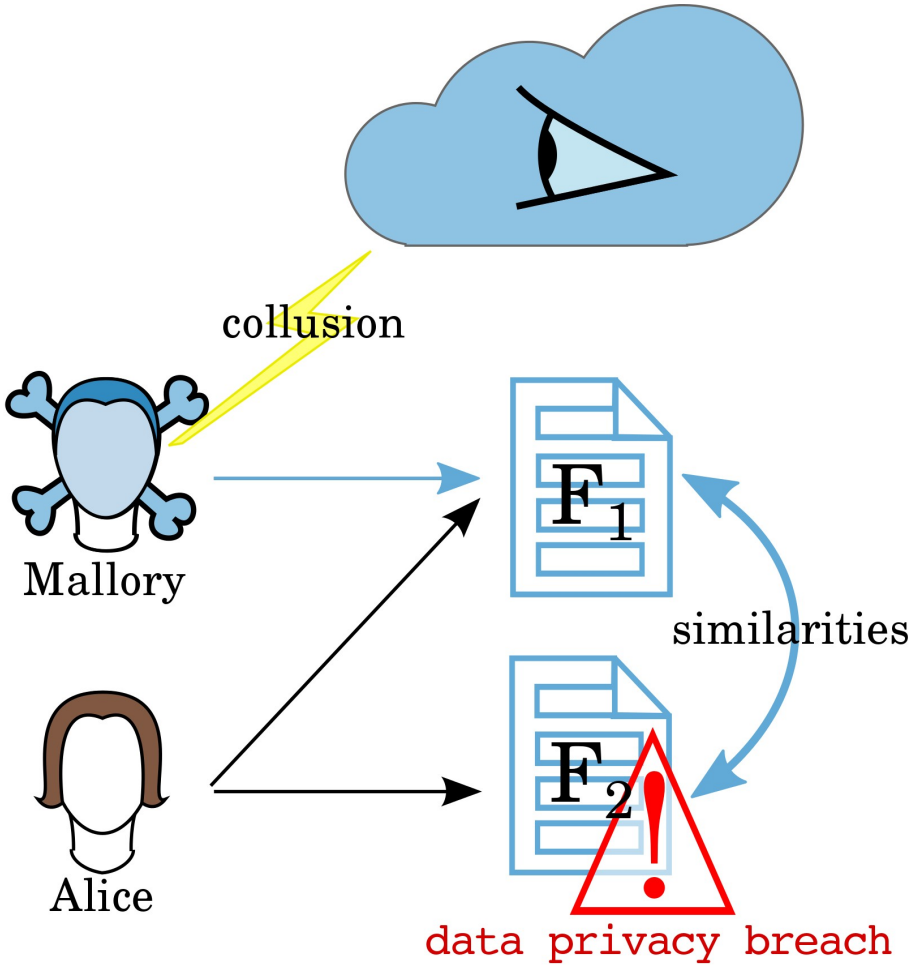


Privacy Issue



- Access Pattern➡Similarities
- Collusion➡Data leakage

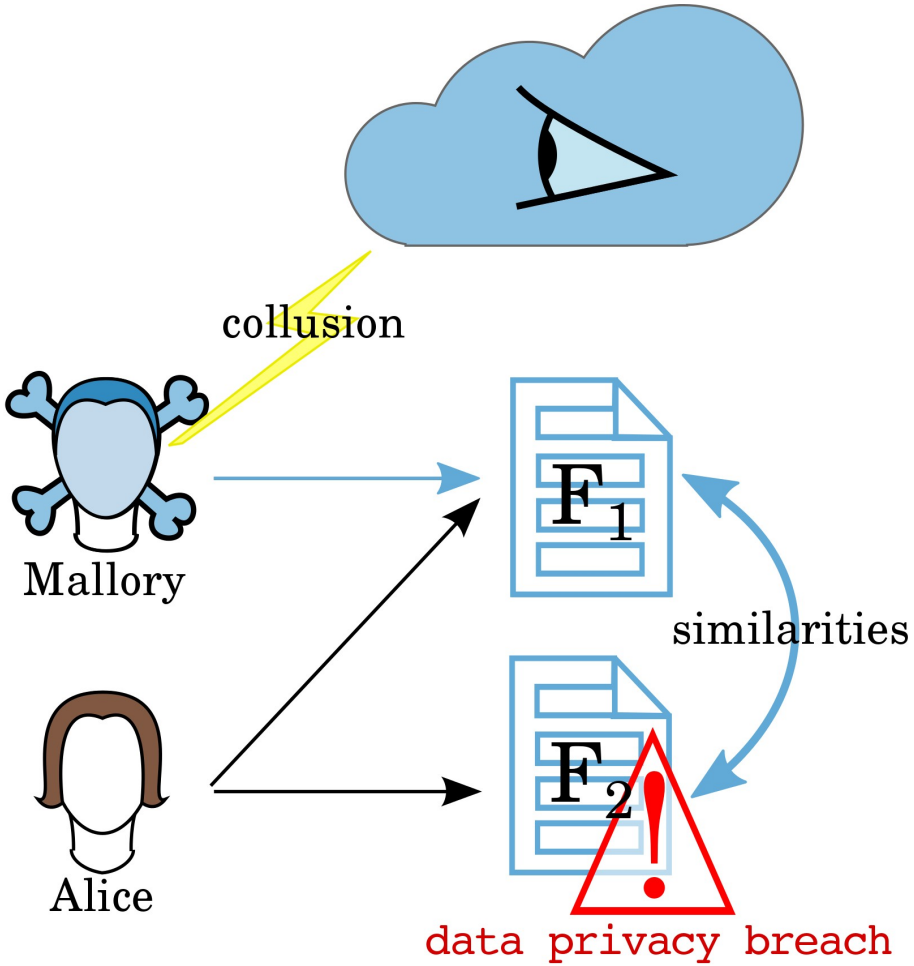
Privacy Issue



- Access Pattern➡Similarities
- Collusion➡Data leakage

➡Privacy Breach:
Data Privacy Broken
Search Privacy Broken

Challenge



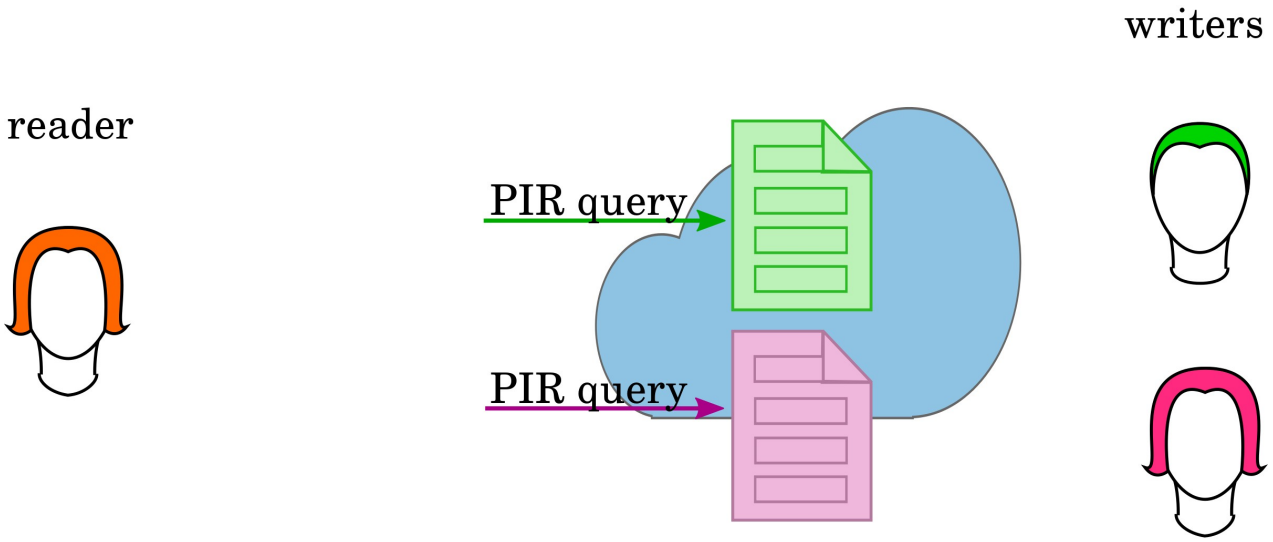
- Access Pattern➡Similarities
- Collusion➡Data leakage

➡Privacy Breach:
Data Privacy Broken
Search Privacy Broken

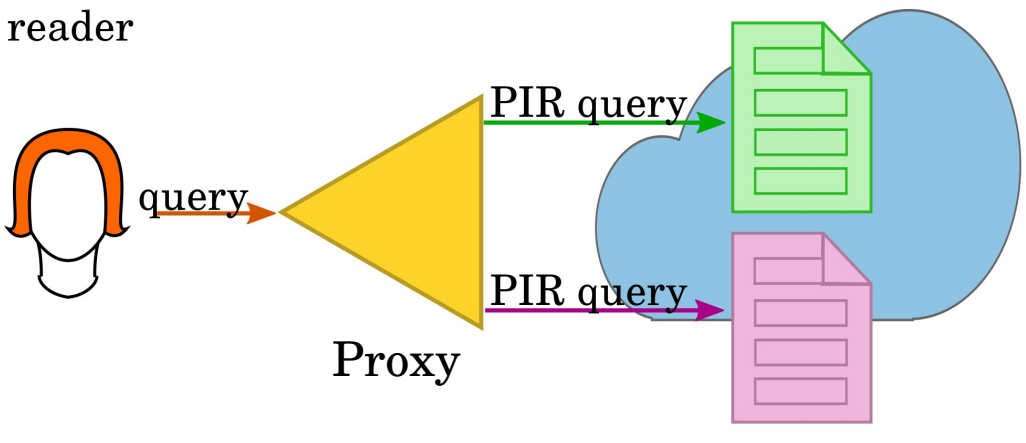
New Adversary Model

Idea of Solution

- **Private Information Retrieval (PIR)** for Privacy

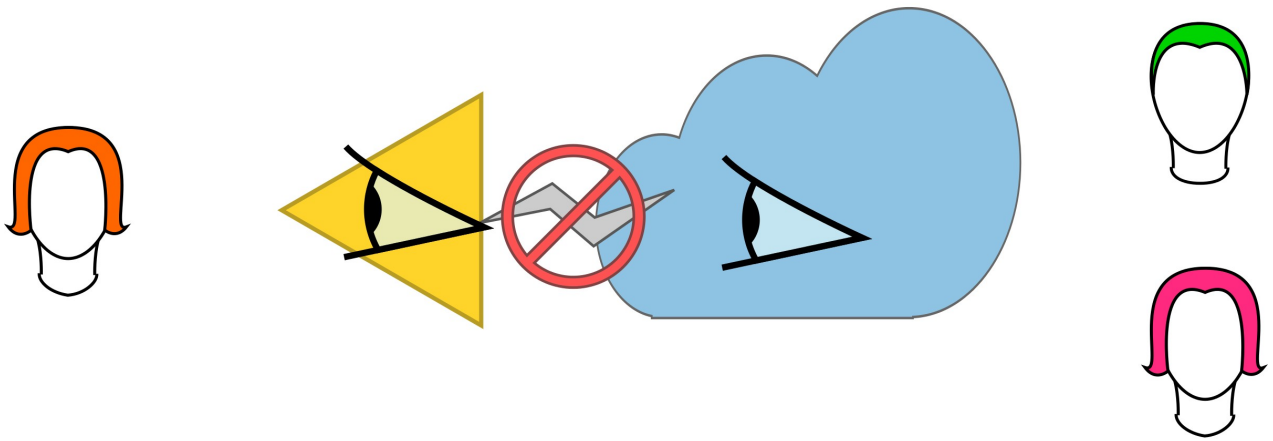


Idea of Solution



- **Private Information Retrieval (PIR)** for Privacy
- Query Transformation through **Proxy**

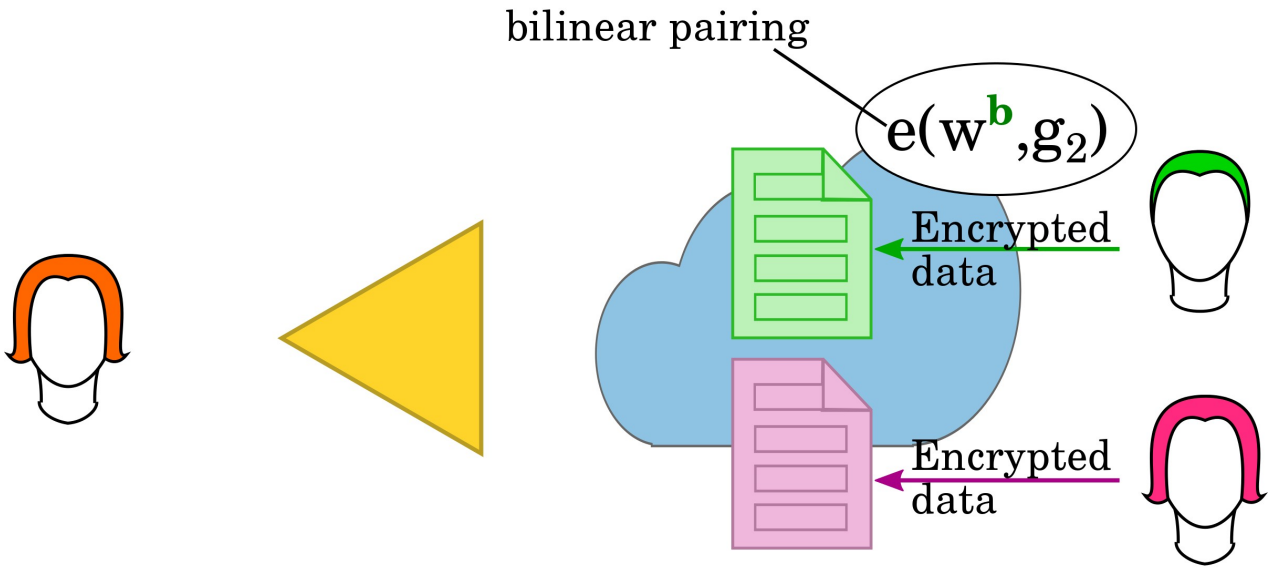
Idea of Solution



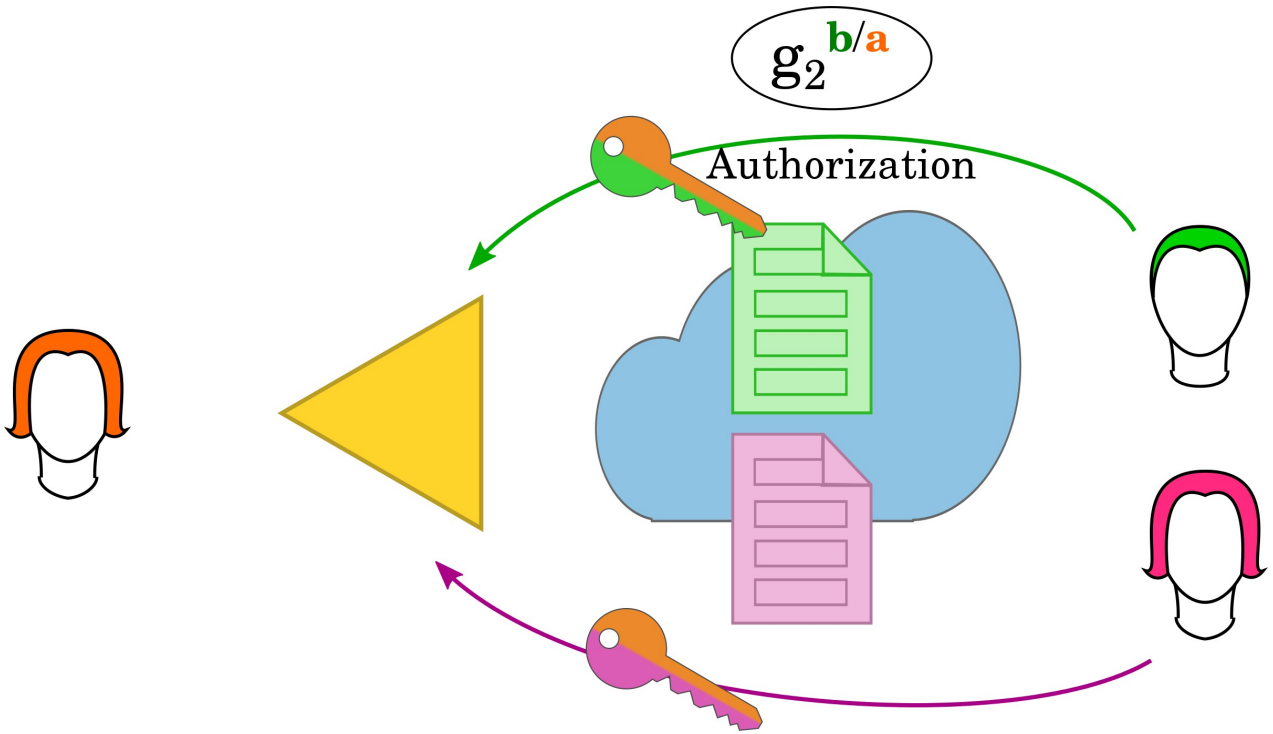
- **Private Information Retrieval (PIR)** for Privacy
- Query Transformation through **Proxy**
- Proxy **not Trusted**
- Single Assumption: **No collusion between Proxy and Cloud**

Protocol Description

- Encrypted words upload



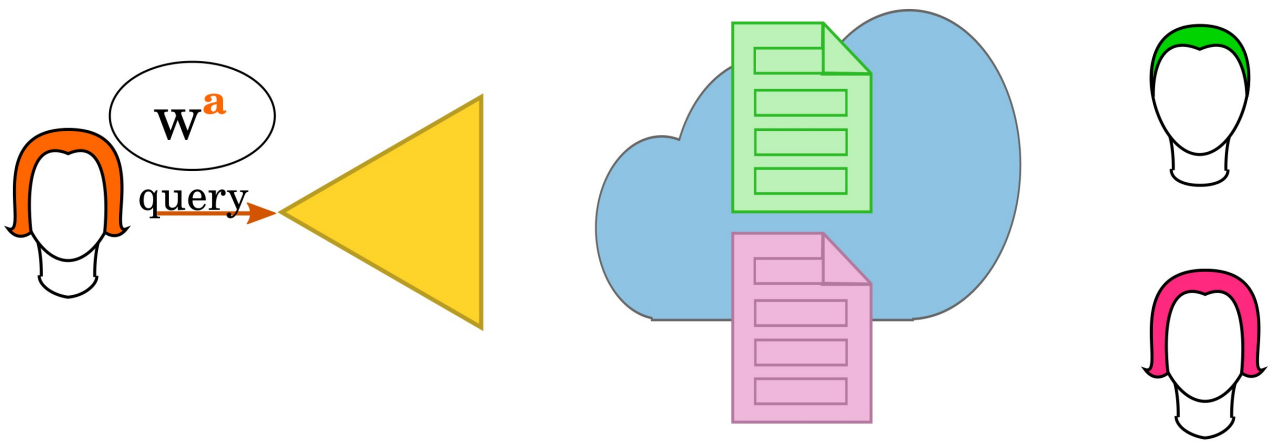
Protocol Description



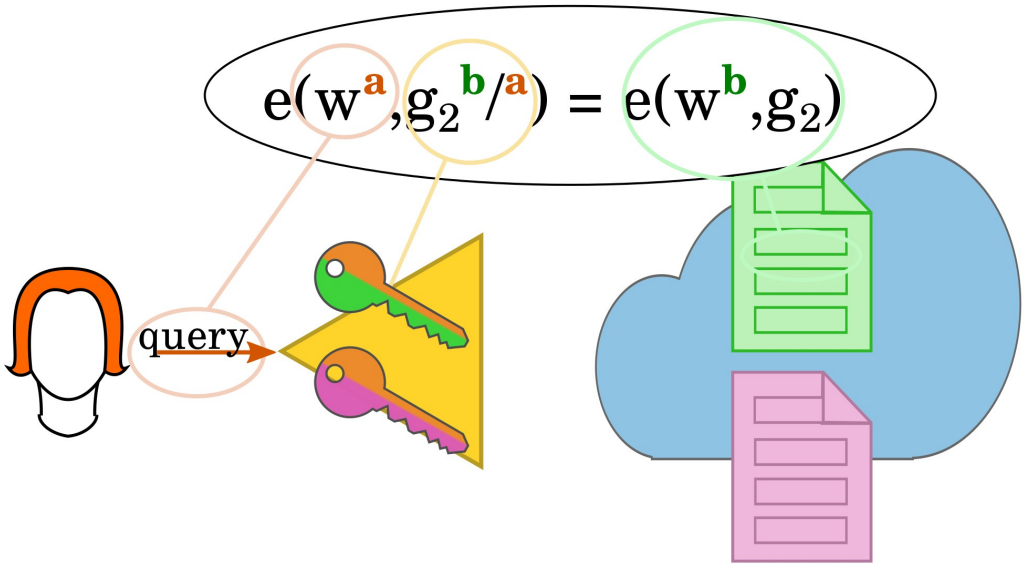
- Encrypted words upload
- Authorization

Protocol Description

- Encrypted words upload
- Authorization
- Querying

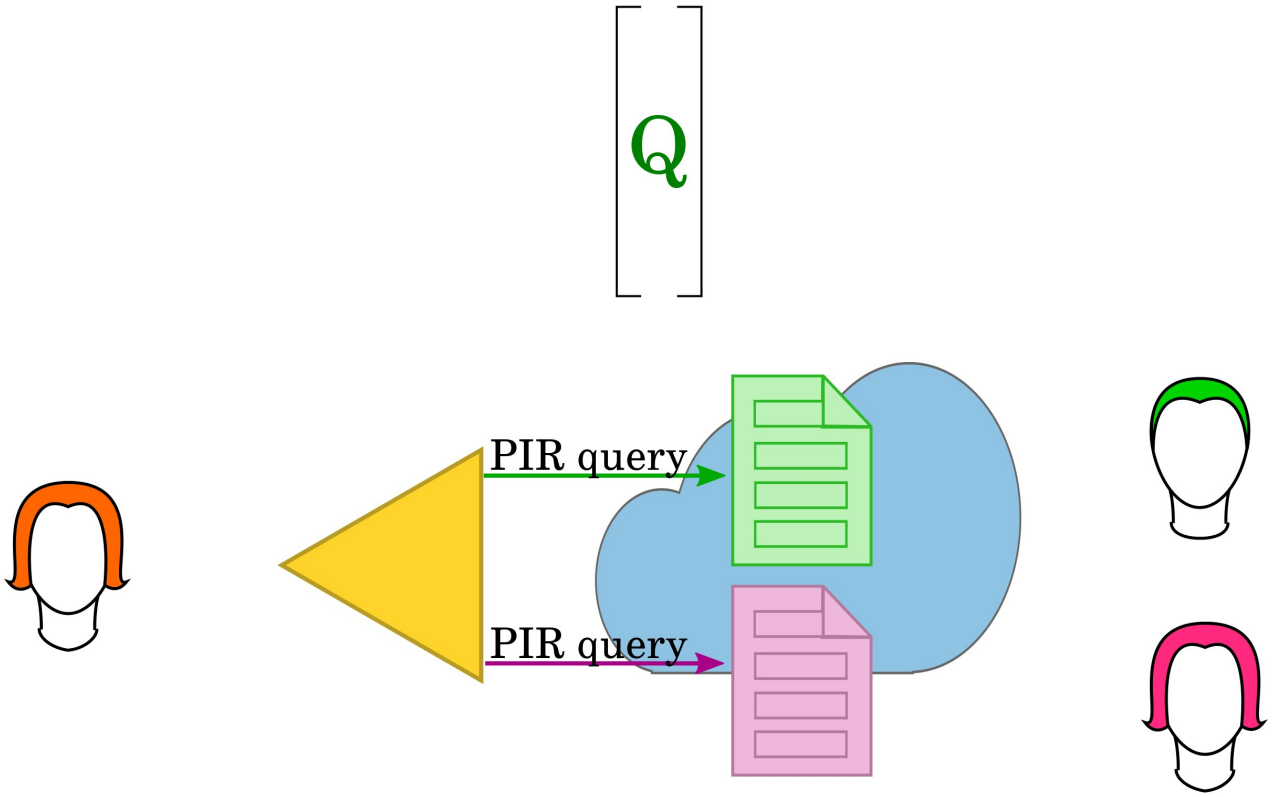


Protocol Description



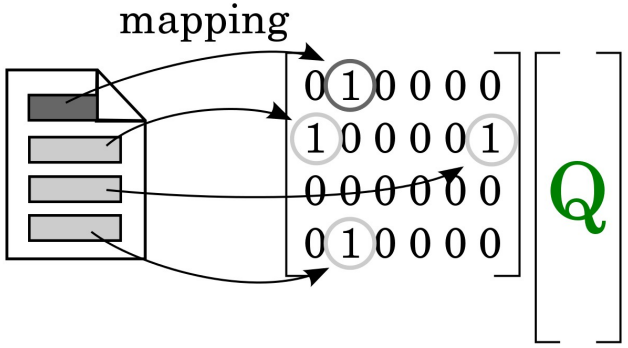
- Encrypted words upload
- Authorization
- Querying
- Query Transformation with **bilinear pairings**

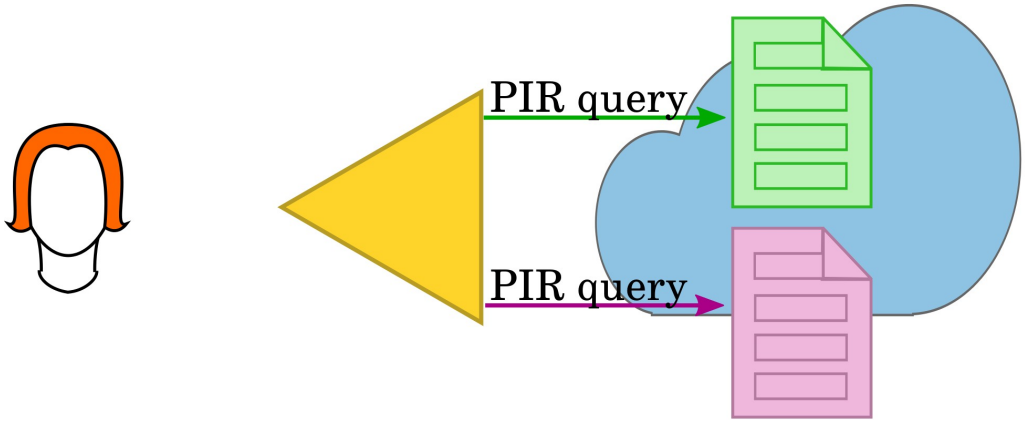
Protocol Description



- Encrypted words upload
- Authorization
- Querying
- Query Transformation with **bilinear pairings**
- **PIR** Querying

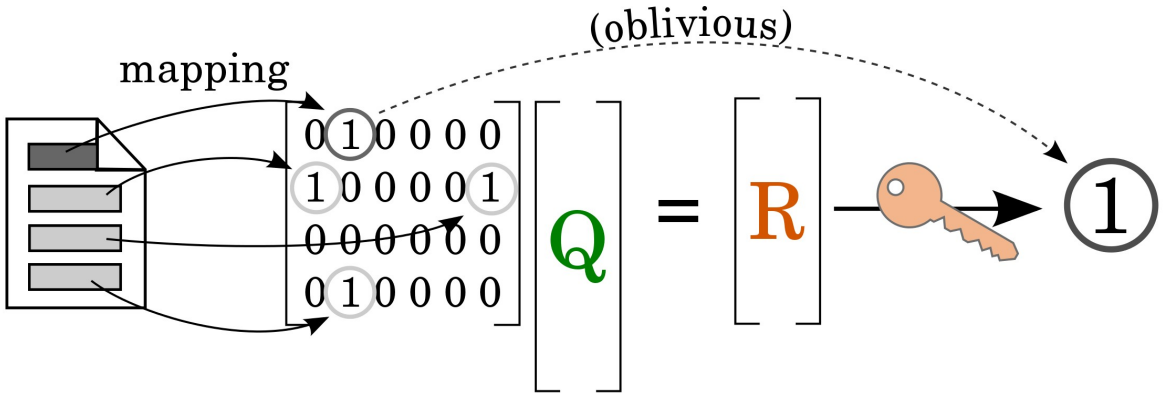
Protocol Description

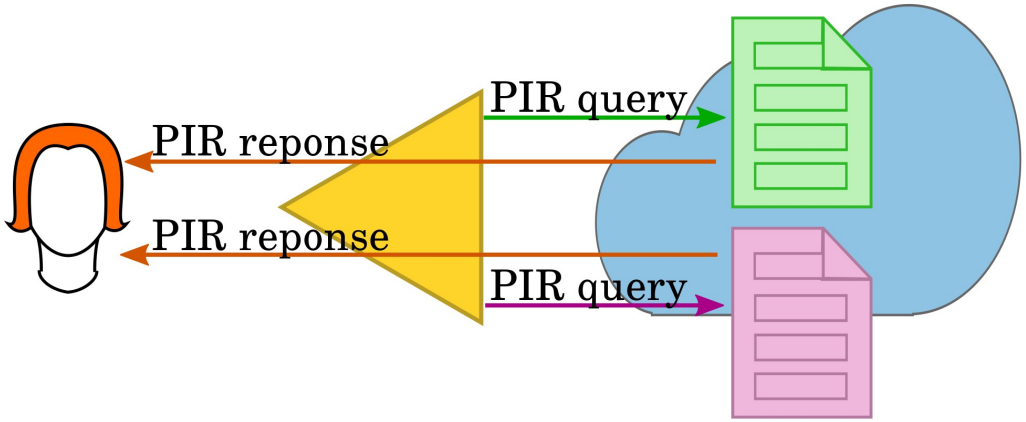


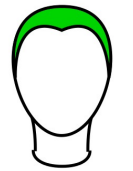


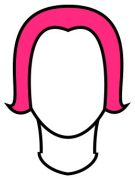
- Encrypted words upload
- Authorization
- Querying
- Query Transformation with **bilinear pairings**
- **PIR** Querying
- (Cheap) **PIR** Processing

Protocol Description





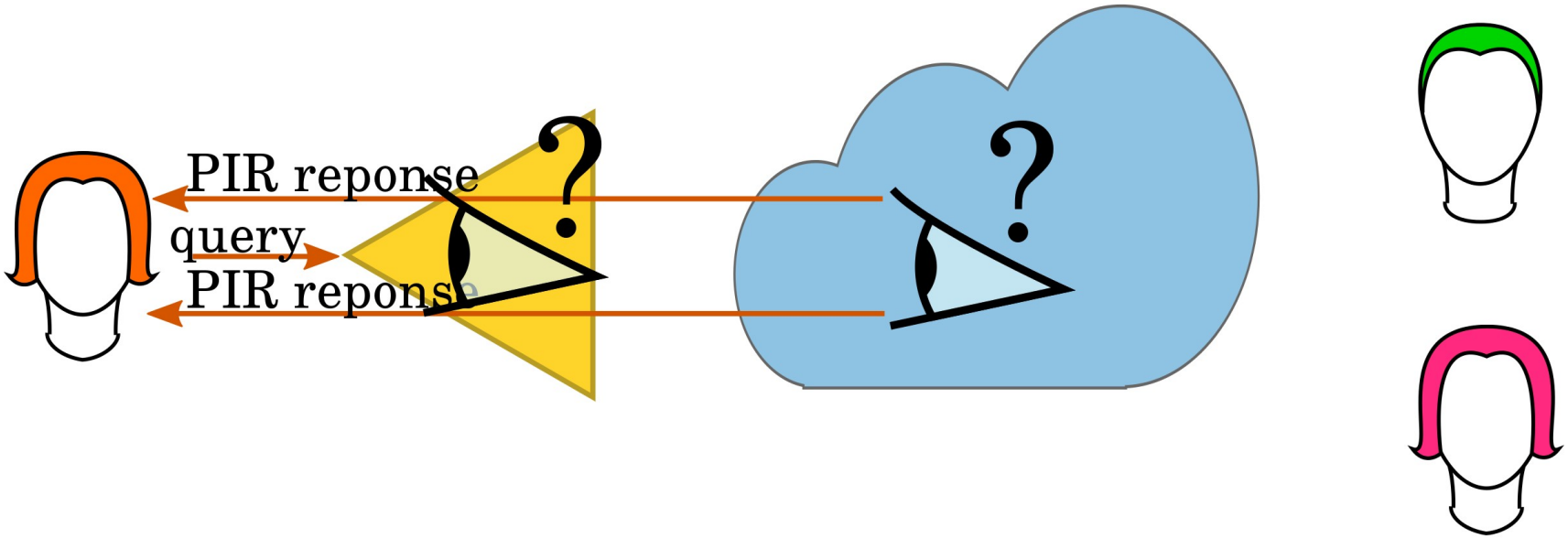




- Encrypted words upload
- Authorization
- Querying
- Query Transformation with **bilinear pairings**
- **PIR** Querying
- (Cheap) **PIR** Processing
- **PIR** Response

Achievements

- ✓ Scalable Search
- ✓ New Adv. Model
- ✓ Privacy in new model
  - ✓ ⊕ Access Pattern Privacy
  - ✓ ⊕ Safe against collusion



Conclusion

Multi-User Searchable Encryption with:

Extended Adversary Model: Collusions among Cloud (resp. Proxy) and any number of (non authorized) users.

Privacy Properties: Data Privacy, Search Pattern Privacy, Access Pattern Privacy

✓**Proved** eXternal Diffie-Hellman (XDH), PIR Security (IND-CCA2), Random Oracle Model

Questions ?

References

[Bao2008] Bao, F., Deng, R.H., Ding, X., Yang, Y.: Private query on encrypted data in multi-user settings. In: Information Security Practice and Experience, pp. 71–85. Springer (2008)

[Yang2011] Yang, Yanjiang, Haibing Lu, and Jian Weng. "Multi-user private keyword search for cloud computing." Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on. IEEE, 2011.

[Popa2014] Popa, R.A., Zeldovich, N.: Multi-Key Searchable Encryption (2013), http://people.csail.mit.edu/nickolai/papers/popa-multikey-eprint.pdf

Privacy Definitions Games

```
/* Setup phase                                                    */
A ← Setup();
for i = 1 to N do
    (γ_i, ρ_i, P_i, K_i) ← KeyGen(κ) ;
    A ← (i, P_i, K_i) ;
end
/* First learning phase                                           */
for j = 1 to a polynomial number l_1 do
    A → query ;
    switch query do
        case Index for word w and user u_i
            A ← Index(w, u_i);
        case Corrupt user u_i
            A ← (ρ_i, γ_i, K_i)
        case Delegation of user u_i by user u_j
            /* A does not receive any value, but the delegation will
               modify the set D_i used in QueryTransform           */
        end
        case Queries for word w from user u_i
            /* D_i comes from the Delegations queried by A         */
            A ← QueryTransform(QueryCreate(w, ρ_i), D_i);
            /* A also receives the randomization factor ξ          */
            A ← ξ;
        case Queries for user query Q from corrupted user u_i
            A ← QueryTransform(Q, D_i);
        case Filtered response for response R from corrupted user u_i
            A ← ResponseFilter(R)
    endsw
end
```

Algorithm 1: Setup and learning phases of both index privacy and query privacy games, whereby $\mathcal{A}$ is the CSP

Privacy Definitions Games

/* Challenge phase */
$\mathcal{A} \rightarrow (u_{chall}, w_0^*, w_1^*)$;
$b \xleftarrow{\$} \{0,1\}$;
$\mathcal{A} \leftarrow Index(w_b^*, u_{chall})$;
/* Restriction phase */
if *u_{chall} is corrupted OR Index for w_0^* or w_1^* for user u_{chall} has been previously queried OR a corrupted user has been delegated by u_{chall}* **then**
  HALT;
end

Algorithm 2: Challenge and restriction phases of the index privacy game whereby $\mathcal{A}$ is the CSP

/* Challenge phase */
$\mathcal{A} \rightarrow (u_{chall}, w_0^*, w_1^*)$;
$b \xleftarrow{\$} \{0,1\}$;
$\mathcal{A} \leftarrow QueryTransform(QueryCreate(w_b^*, \rho_{chall}), D_{chall})$;
/* Restriction phase */
if *u_{chall} is corrupted* **then**
  HALT;
end

Algorithm 3: Challenge and restriction phases of the query privacy game whereby $\mathcal{A}$ is the CSP

Transition Game 0→1

$\mathcal{A}_1$ receives data from $game_1$ setup phase;
/* $\mathcal{A}_1$ simulates some users to $\mathcal{A}_0$ */
$Sim \xleftarrow{\$} \mathcal{P}([1..N])$;
for $i \in Sim$ **do**
  $(\gamma'_i, \rho'_i, P'_i, K'_i) \leftarrow KeyGen(\kappa)$
end
for *i from* 1 *to* N **do**
  if $i \in Sim$ **then**
    $\mathcal{A}_0 \leftarrow (i, P'_i, K'_i)$;
  else
    $\mathcal{A}_0 \leftarrow (i, P_i, K_i)$
  end
end
/* Learning phase 1 */
for *a polynomial number* l_1 *of times* **do**
  $\mathcal{A}_0 \rightarrow$ query;
  if $\mathcal{A}_1$ *knows all the input values for the corresponding algorithm* **then**
    $\mathcal{A}_1$ runs the algorithm locally and sends back the answer;
  else
    if *query was for corruption* **then**
      /* exit with random guess */
      $b^* \xleftarrow{\$} 0, 1$;
      $\mathcal{A}_1 \rightarrow b^*$;
      HALT;
    else
      $\mathcal{A}_1$ forwards the call to $game_1$ and forwards the answer to $\mathcal{A}_0$;
    end
  end
end
/* Challenge phase */
$\mathcal{A}_1$ forwards everything from $\mathcal{A}_0$ to $game_1$ and back.
/* Learning phase 2 */
Same as learning phase 1;
/* Response phase */
$\mathcal{A}_1$ forwards the bit b^* outputted by $\mathcal{A}_0$;

Algorithm 4: Algorithm run by $\mathcal{A}_1$ the transition adversary from $game_0$ to $game_1$. Restrictions phases are omitted.

Transition Game 2→3

$\mathcal{D}_{DDH} \leftarrow (g_1, g_1^{\alpha}, g_1^{\beta}, g_1^{\delta})$;
$\mathcal{A} \leftarrow Setup()$;
$predict \xleftarrow{\$} [1..N]$;
$I \xleftarrow{\$} [0, .., l]$;
for *i from* 1 *to* N **do**
  $(\gamma_i, \rho_i, P_i, K_i) \leftarrow KeyGen(\kappa)$;
  $\mathcal{A} \leftarrow (i, P_i, K_i)$
end
for *a polynomial number* l *of times* **do**
  $\mathcal{A} \rightarrow$ query;
  switch *query* **do**
    case *hash of word* w *through* h
      if *this is the* I*-th call to* $\mathcal{O}$ **then**
        $\mathcal{A} \leftarrow g_1^{\beta}$
      else
        $\mathcal{A} \leftarrow g_1^{\mathcal{O}[w]}$
      end
    case *Index for word* w *and user* $u_{predict}$
      $\mathcal{A} \leftarrow e((g_1^{\alpha})^{\mathcal{O}[w]}, g_2)$;
    otherwise
      normal handling of the query;
    end
  endsw
end
$\mathcal{A} \rightarrow (u_{chall}, w_0^*, w_1^*)$;
$b \xleftarrow{\$} \{0, 1\}$;
if $chall \neq predict$ *OR* $I = 0$ *and* $\mathcal{O}$ *has been called with input* w_b^* *OR* $I \neq 0$ *and* w_b^* *does not correspond to the* I*-th call to* $\mathcal{O}$ **then**
  $b_{DDH} \xleftarrow{\$} 0, 1$;
  $\mathcal{D}_{DDH} \rightarrow b_{DDH}$;
  HALT;
end
$\mathcal{A} \leftarrow e(g_1^{\delta}, g_2)$;
$\mathcal{A} \rightarrow b^*$;
if $b^* = b$ **then**
  $\mathcal{D}_{DDH} \rightarrow 1$;
else
  $\mathcal{D}_{DDH} \rightarrow 0$;
end

Algorithm 5: Listing for the distinguishing algorithm $\mathcal{D}_{DDH}$ from $game_2$ to $game_3$.