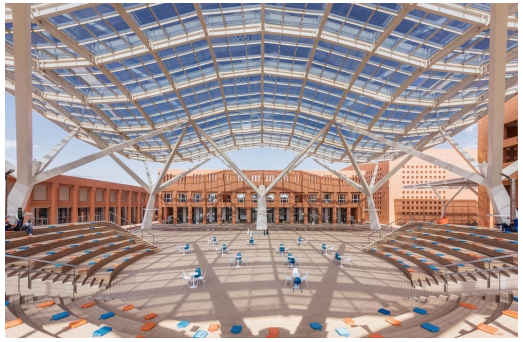
# Cryptanalysis Strikes Back

A Realistic assessment of leakage attacks on Encrypted Search

**Abdelkarim Kati**†‡

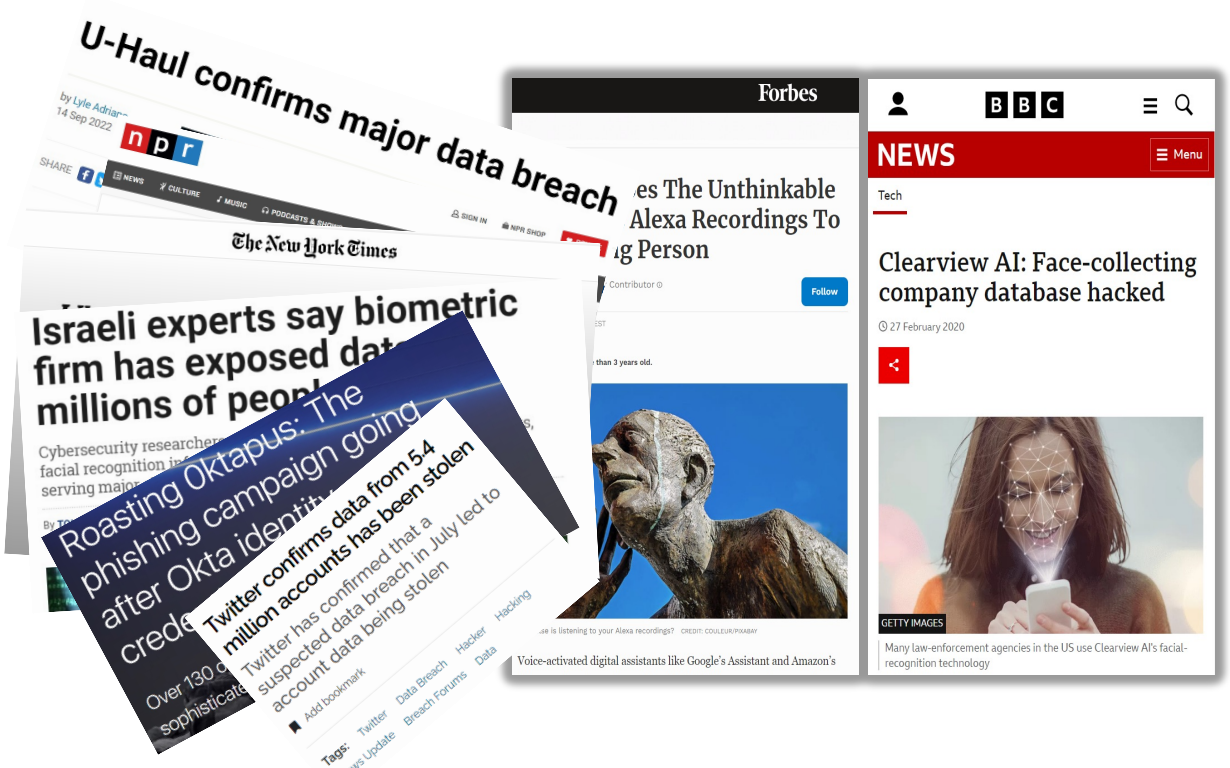
†School of Computer Science,
Mohammed VI Polytechnic University.
‡ Encrypted Systems Lab, Brown University.

January 24, 2023 at Aarhus University.



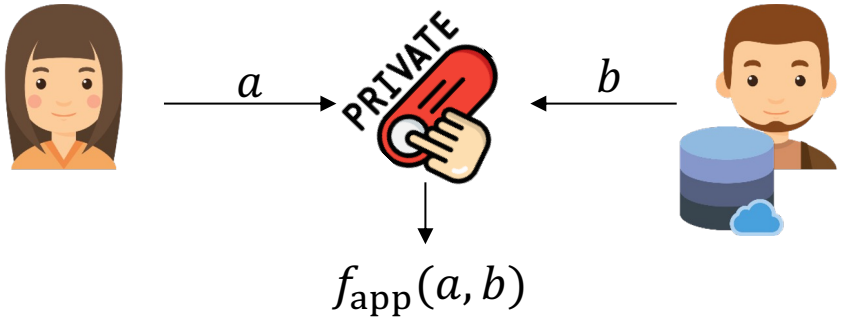
# Motivation

## Real-World Applications



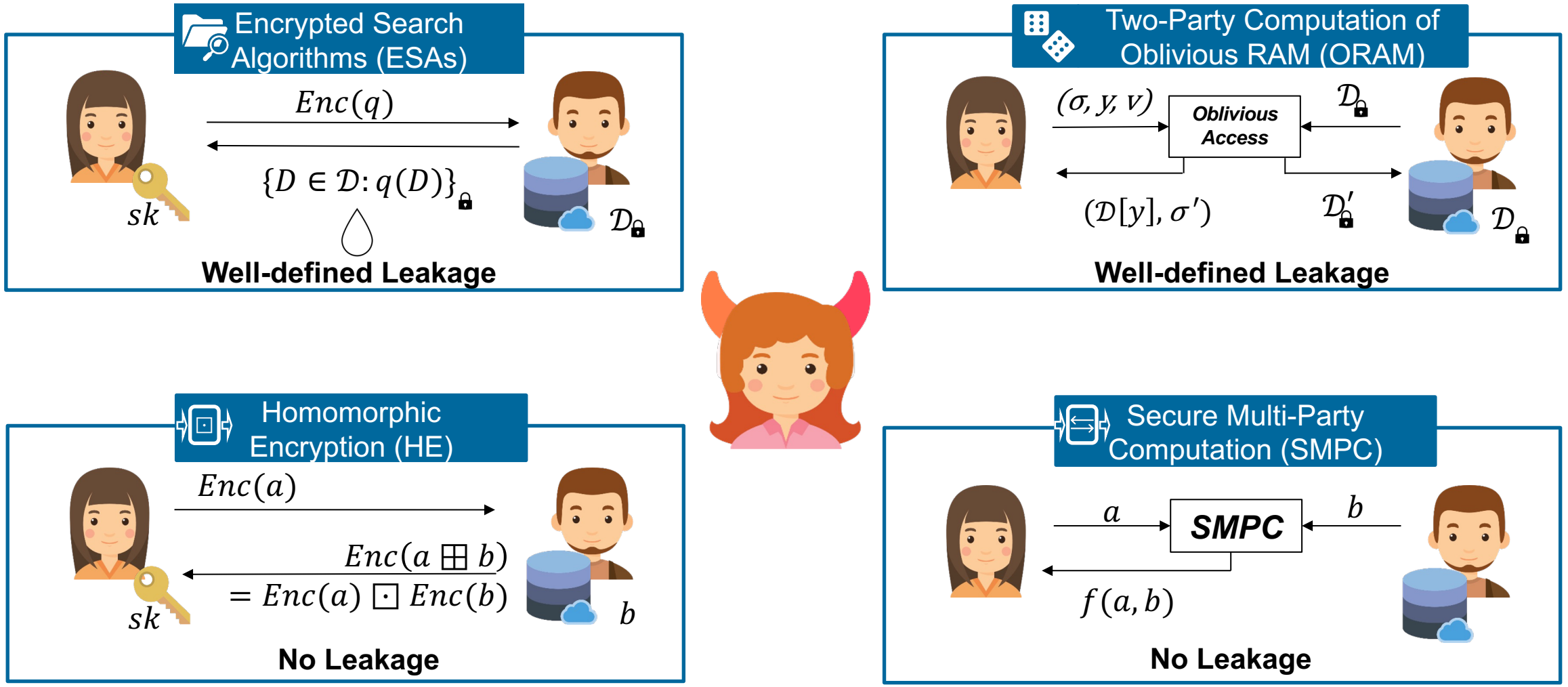

**Leakage = erosion of privacy w.r.t data protection**

## Cryptographic Mechanisms




**Privacy-Enhancing Technologies (PETs)**

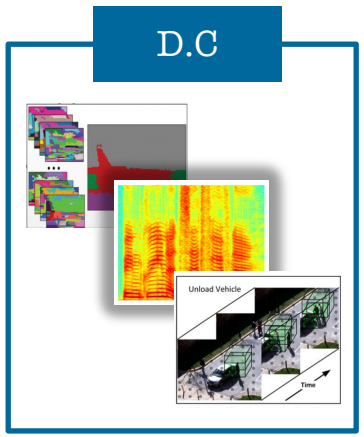
# A Realistic assessment of leakage attacks on Encrypted Search



A Realistic assessment of leakage attacks on **Encrypted Search**

# Encrypted Search Algorithms (ESAs)

**Trusted**
**Client**

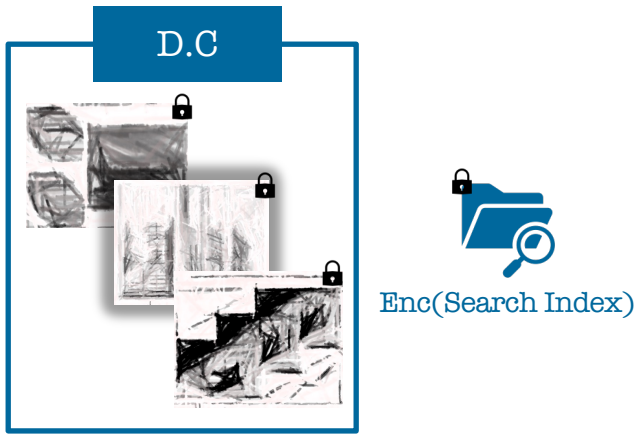



**Untrusted**
**Server**

# Encrypted Search Algorithms (ESAs)

Trusted
Client

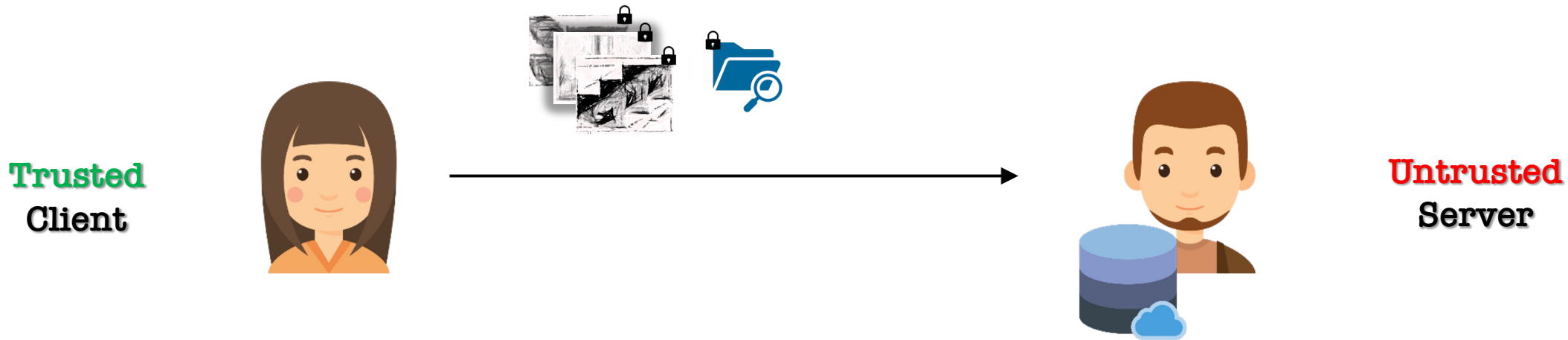
Untrusted
Server



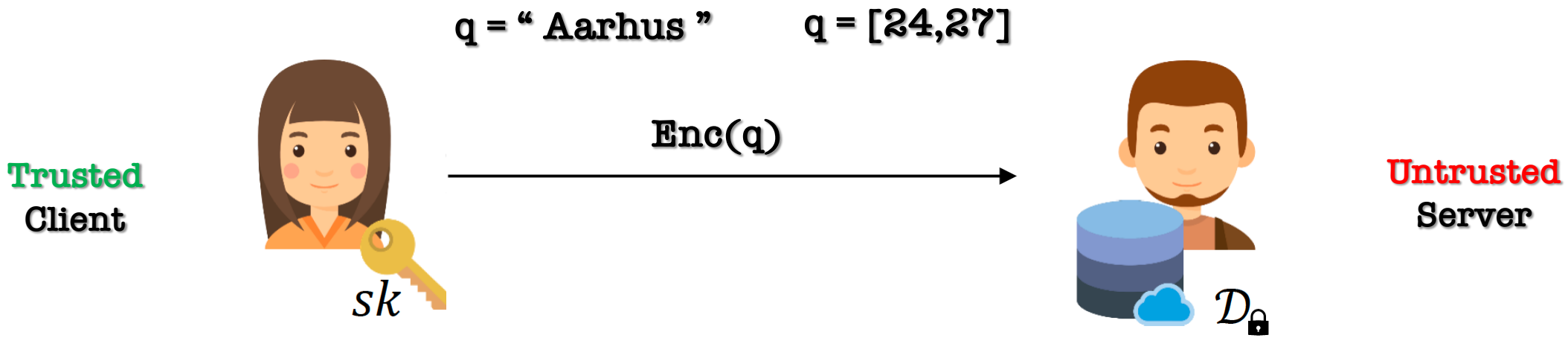



Enc(Search Index)

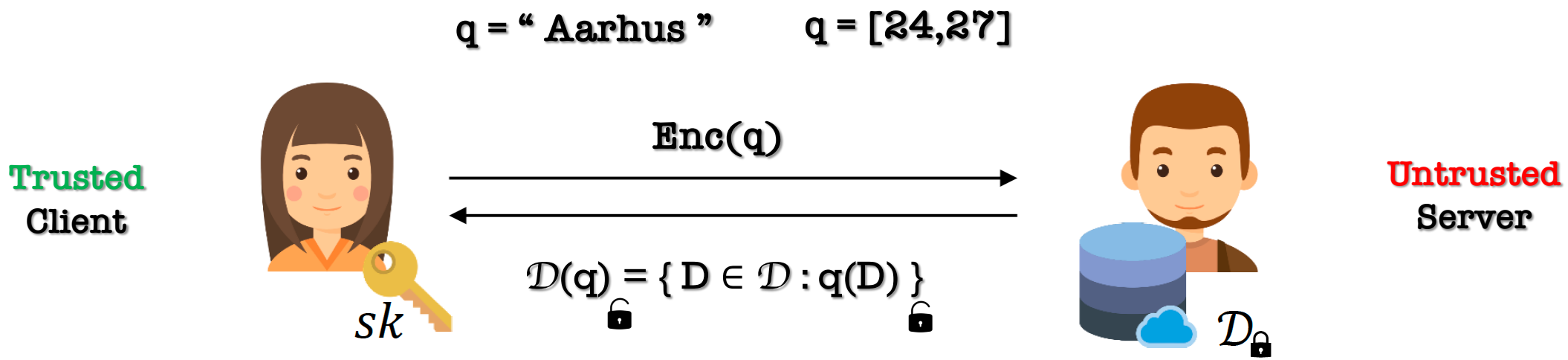
# Encrypted Search Algorithms (ESAs)



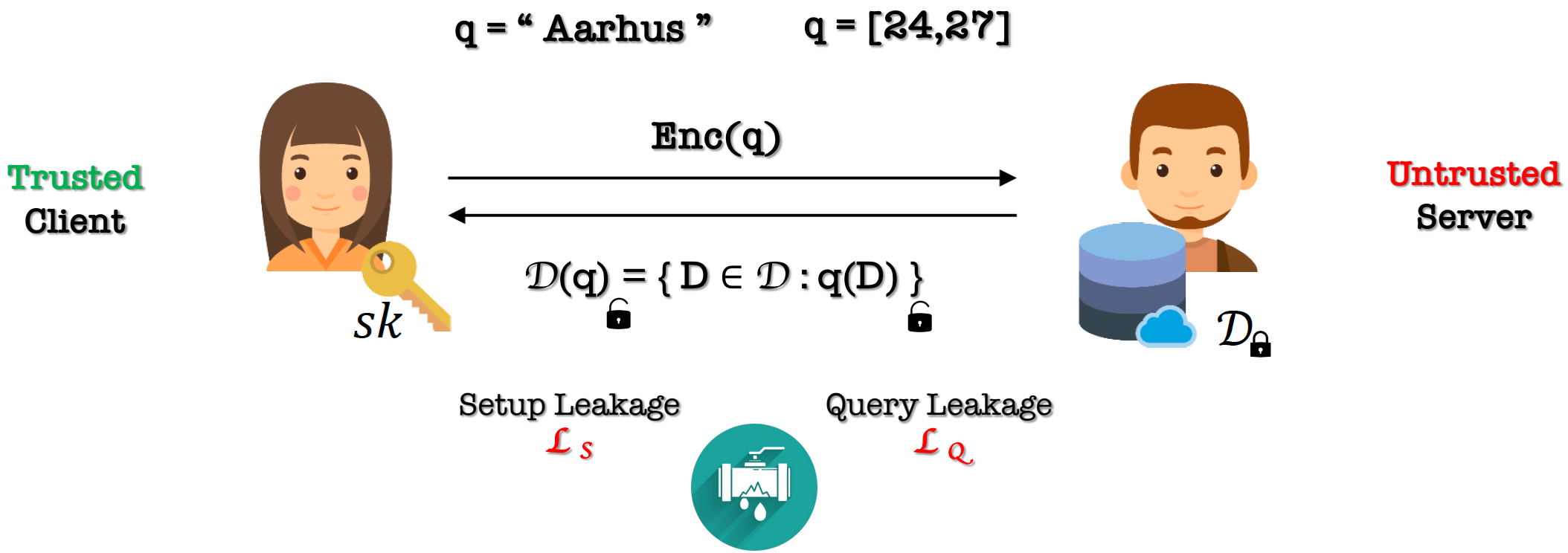
# Encrypted Search Algorithms (ESAs)



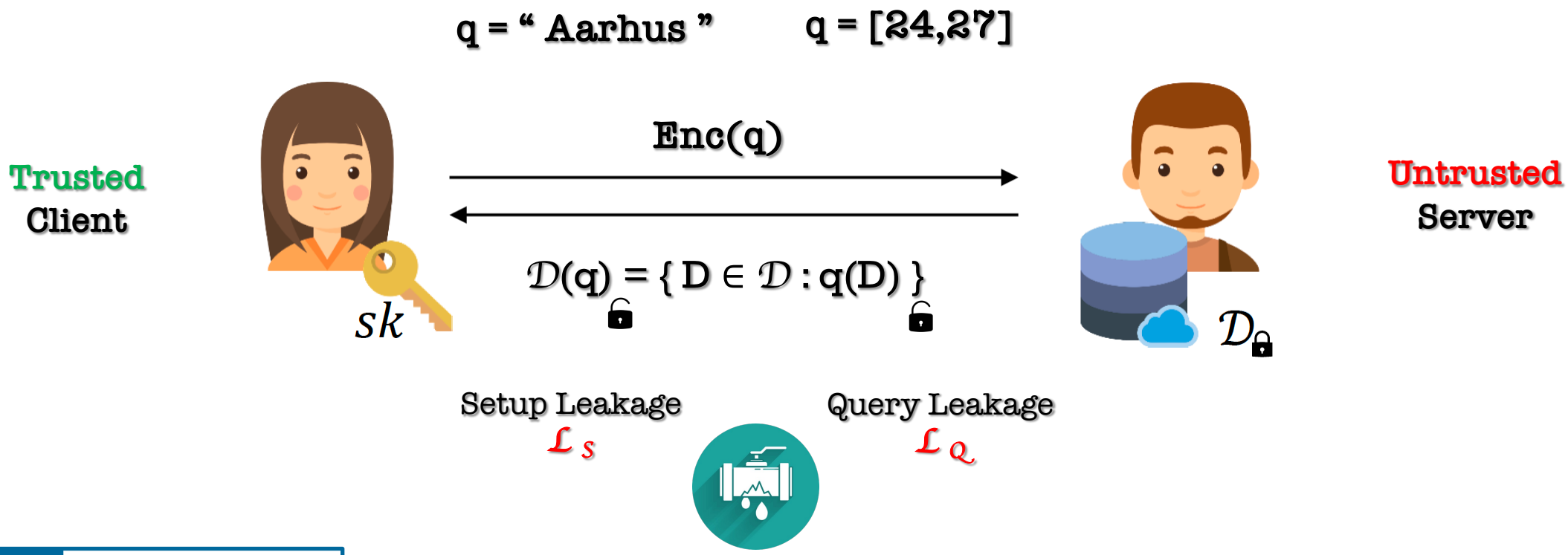
## Encrypted Search Algorithms (ESAs)



# Encrypted Search Algorithms (ESAs)



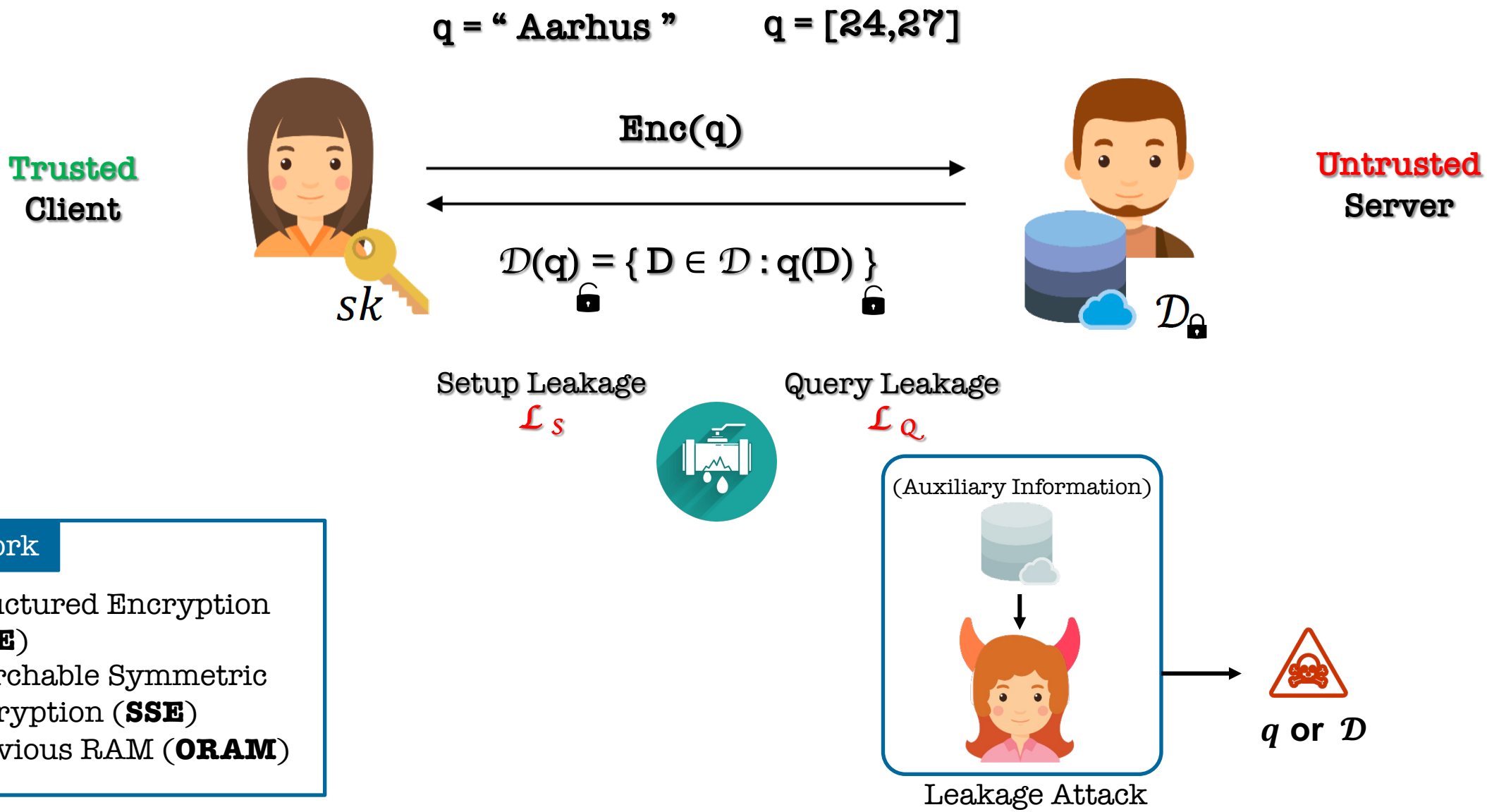
# Encrypted Search Algorithms (ESAs)




**Our work**

- Structured Encryption (**STE**)
- Searchable Symmetric Encryption (**SSE**)
- Oblivious RAM (**ORAM**)

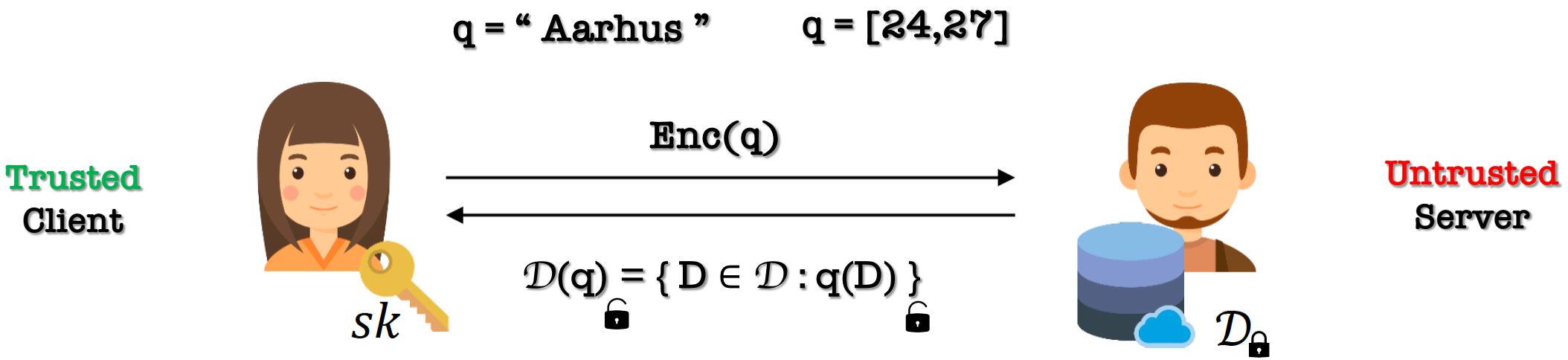
# Encrypted Search Algorithms (ESAs)




**Our work**

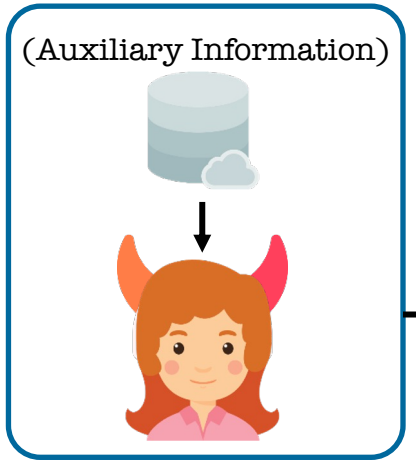
- Structured Encryption (**STE**)
- Searchable Symmetric Encryption (**SSE**)
- Oblivious RAM (**ORAM**)

# Encrypted Search Algorithms (ESAs)




Setup Leakage $\mathcal{L}_S$

Query Leakage $\mathcal{L}_Q$




$q$ or $\mathcal{D}$

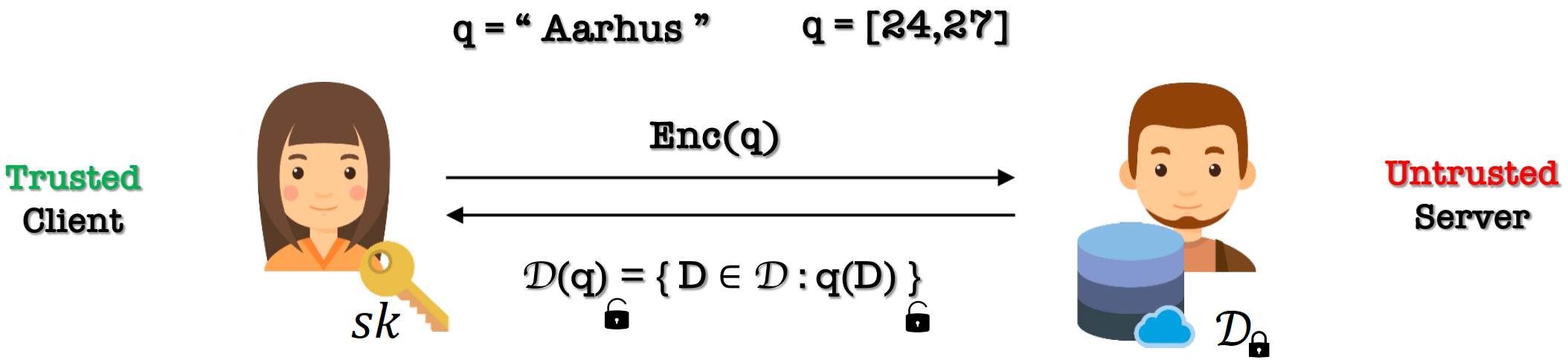
**Adversary**

Persistent & Passive

**Our work**

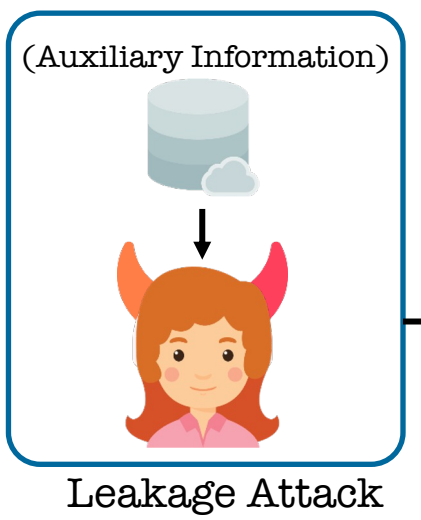
- Structured Encryption (**STE**)
- Searchable Symmetric Encryption (**SSE**)
- Oblivious RAM (**ORAM**)

# Encrypted Search Algorithms (ESAs)




Setup Leakage $\mathcal{L}_S$

Query Leakage $\mathcal{L}_Q$





**Adversary**

Persistent & Passive

**Our work**

- Structured Encryption (**STE**)
- Searchable Symmetric Encryption (**SSE**)
- Oblivious RAM (**ORAM**)

A Realistic assessment of **Leakage Attacks** on Encrypted Search

# How do we model Leakage ?

- The "Baseline" leakage profile for response-revealing EMMs
  - ✓ $(\mathcal{L}_S, \mathcal{L}_Q, \mathcal{L}_U)$ = (dsize, (qeq, rid), usize)
- The "Baseline" leakage profile for response-hiding EMMs
  - ✓ $(\mathcal{L}_S, \mathcal{L}_Q, \mathcal{L}_U)$ = (dsize, qeq, usize)
- Several new constructions have better leakage profiles
  - ✓ AZL and FZL [Kamara-Moataz-Ohirimenko'18]
  - ✓ VHL and AVHL [Kamara-Moataz'19]

| Leakage | Information |
|---|---|
| Response Length | $\lvert D(q)\rvert$ |
| Query Equality | $q_i = q_j$ |
| Co-Occurrence | $\lvert D(q_i) \cap D(q_j)\rvert$ |
| Response Identity | $\{i: D_i \in q(D)\}$ |
| Response Volumes | $\{\lvert D_i\rvert_b : D_i \in q(D)\}$ |

**(Simplified)**

## Leakage Attacks Types

**Keyword** (point) queries
[IKK12,CGPR15,BKM20,RPH21]

| Keyword | Document IDs |
|---|---|
| 'Aarhus' | 2,5,11,13,20,31 |
| 'systems' | 3,5,10,11,13,25 |
| 'lab' | 5,11,21,27 |

$$q = w$$
$$\mathcal{D}(q) = \{D \in \mathcal{D} : q \in D)\}$$
**Recover q**

$q = 'Aarhus'$

**Known-data**: Adversary knows subset of $\mathcal{D}$

**Range** queries

[KKNO16,LMP18,GLMP18,
GLMP19,GJW19,KPT20,KPT21]

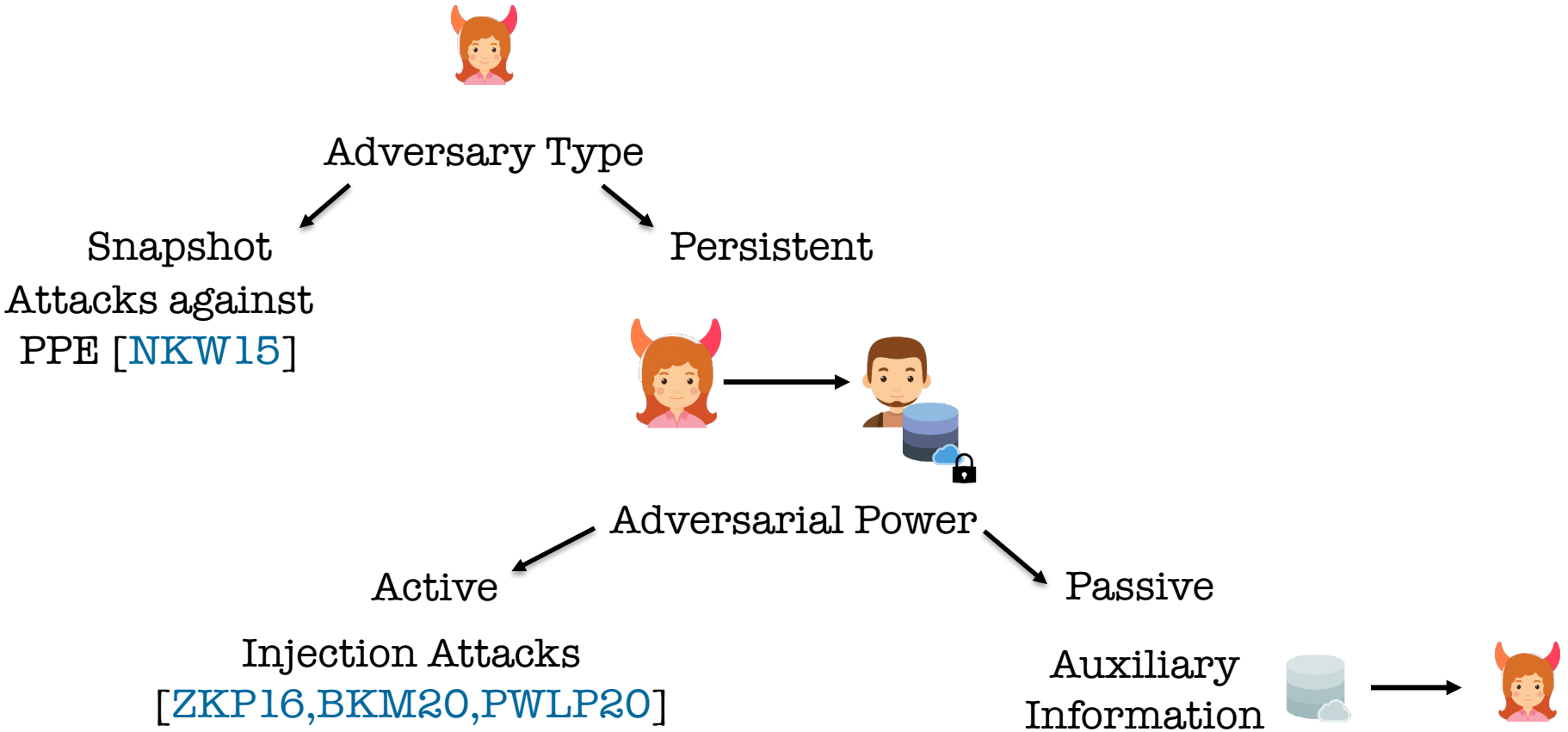
| ID | Age |
|---|---|
| 1 | 65 |
| 2 | 7 |
| 3 | 27 |

$$q = (a, b)$$
$$\mathcal{D}(q) = \{r \in \mathcal{D} : a \leq r \leq b\}$$
**Recover $\mathcal{D}$**

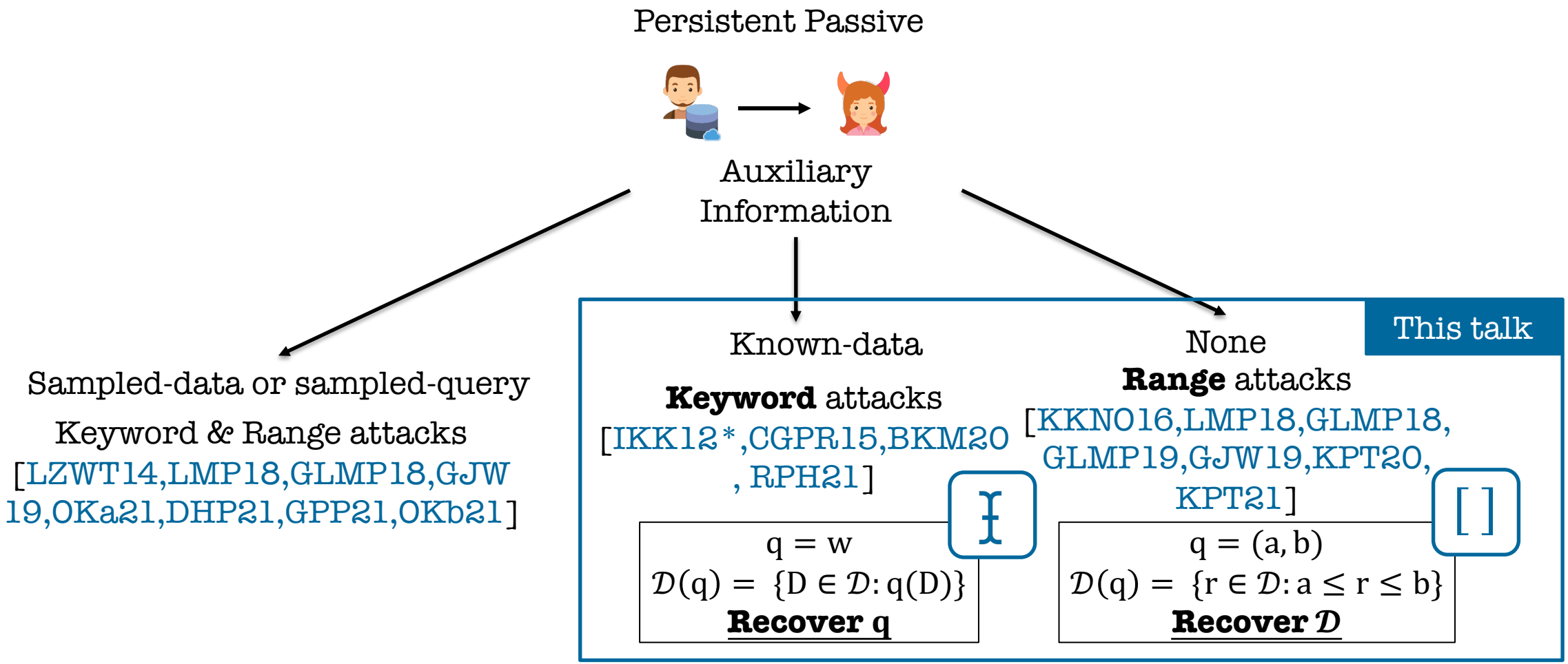
$q = (18,39)$

No auxiliary knowledge

# Leakage Attacks against ESAs

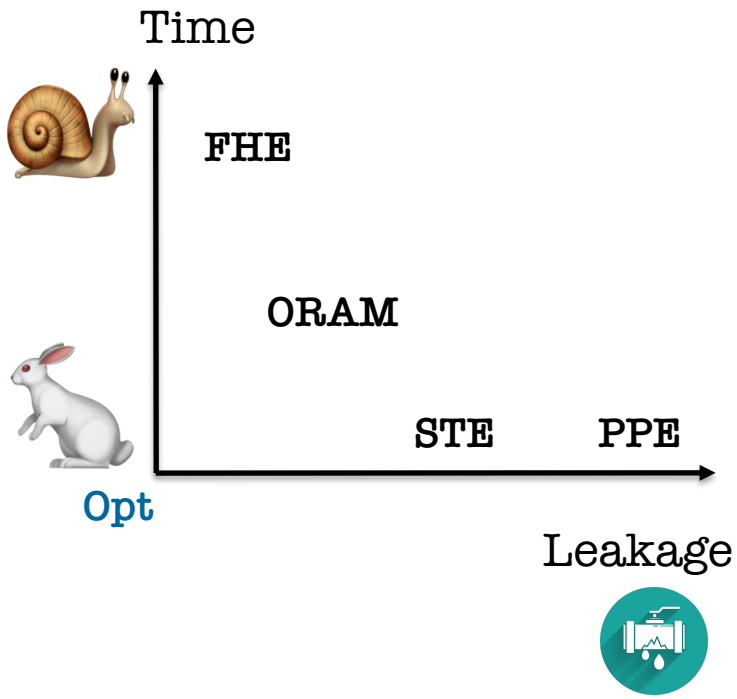


# Leakage Attacks against ESAs



# ESAs Techniques Overview

| Technique | Leakage | Query Time | |
|---|---|---|---|
| Fully Homomorphic Encryption (FHE) | • None | Linear | Considered secure but inefficient |
| Oblivious RAM (ORAM) | • Response Length + Volume | Sublinear | Our work |
| Structured Encryption (STE) | • Query Equality<br>• Response Identities + Volumes | Optimal | Considered efficient and ??? |
| Property-Preserving Encryption (PPE) | • Ciphertext Equality<br>• Ciphertext Order | Optimal | Considered efficient but insecure [NKW15] |



# Uncertainty Of Security

## Constructions

"Benign leakage"

"Common leakage"

"Standard leakage"

"Accepted leakage"

"[Attacks] assume extremely strong adversarial models"

"Leakages [...] are not exploitable via leakage-abuse attacks in practice"

## Attacks & Countermeasures

"Severe threat"

"Devastating results"

"[ESAs] are extremely vulnerable to [attacks]"

"[ESA] schemes should no longer be used without countermeasures"

"Our assumptions on background information are weak"

"With some prior knowledge [...] an honest-but-curious server can recover the underlying keywords"

# Uncertainty Of Security

## Constructions

## Attacks & Countermeasures

> Benign leakage

> Standard leakage

> [Attacks] assume extremely stron… models

> Leakages […] are not exploitable … abuse attacks in practi…

> …hreat

> Devastating results

> […ESAs] are extremely …vulnerable to […attacks]

> [ESA] schemes should no longer be used without countermeasures

> …ur assumptions on background information are weak

> … some prior knowledge […] an honest-but-…server can recover the underlying keywords

A **Realistic Assessment** of Leakage Attacks on Encrypted Search

# Previous Evaluations

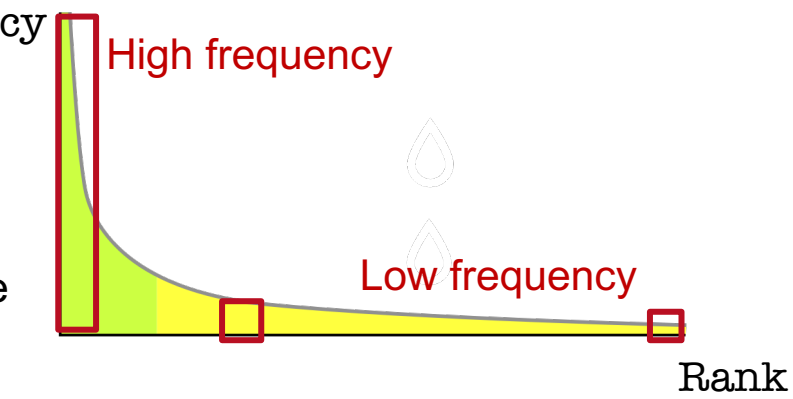
Usual evaluations for **Keyword attacks:**

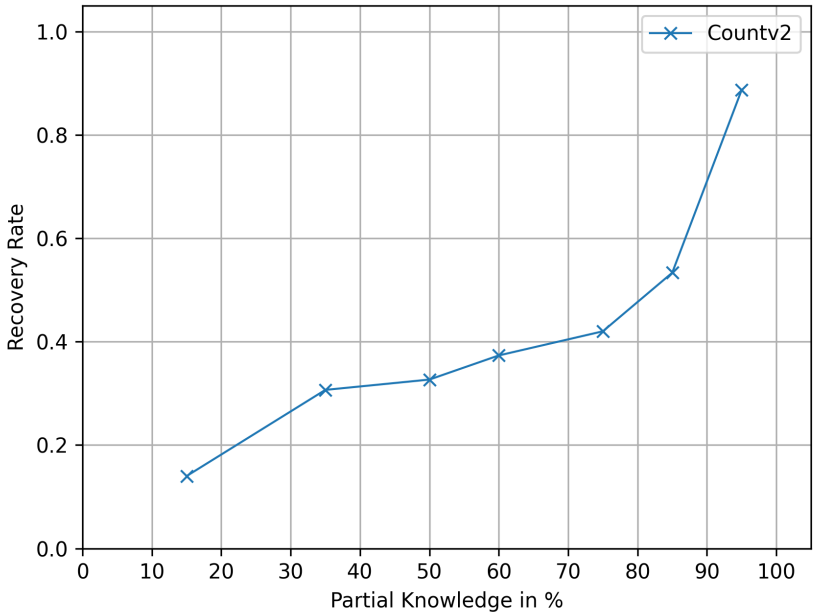
1. Enron (& Apache) email data collection

2. Restrict data to 500-3000 keywords

3. Draw 150 queries **from data collection**
→ ??? From which part of the distribution ?

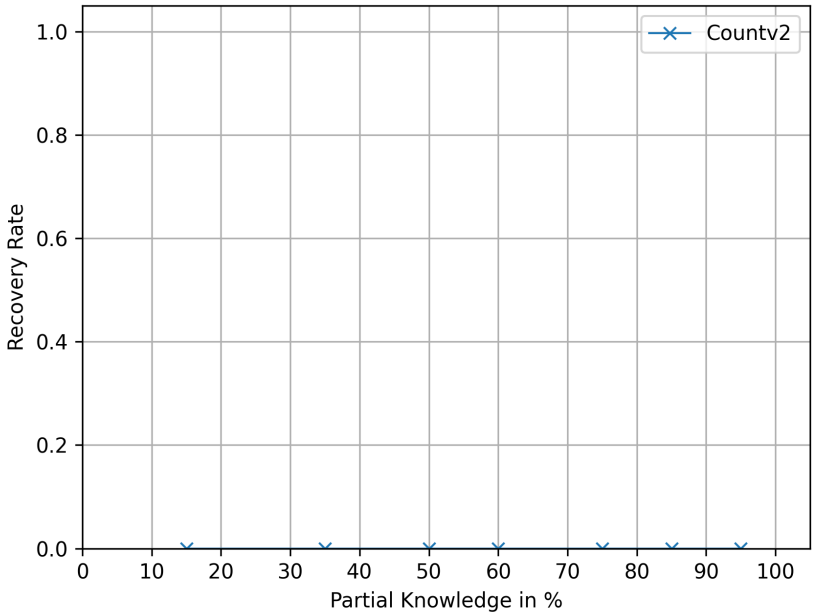
4. Evaluate on **partial knowledge**




High frequency

**or**




Low frequency

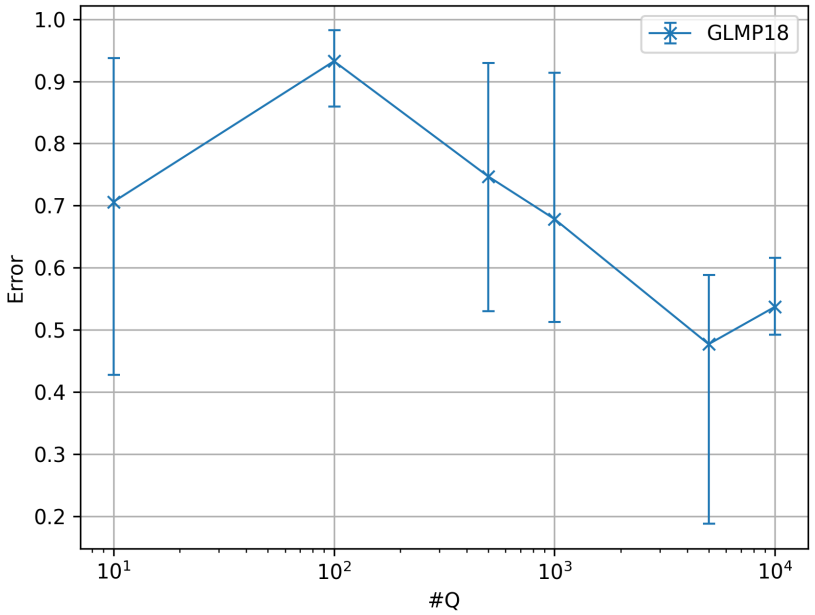
# Previous Evaluations

Usual evaluations for **Range attacks:**

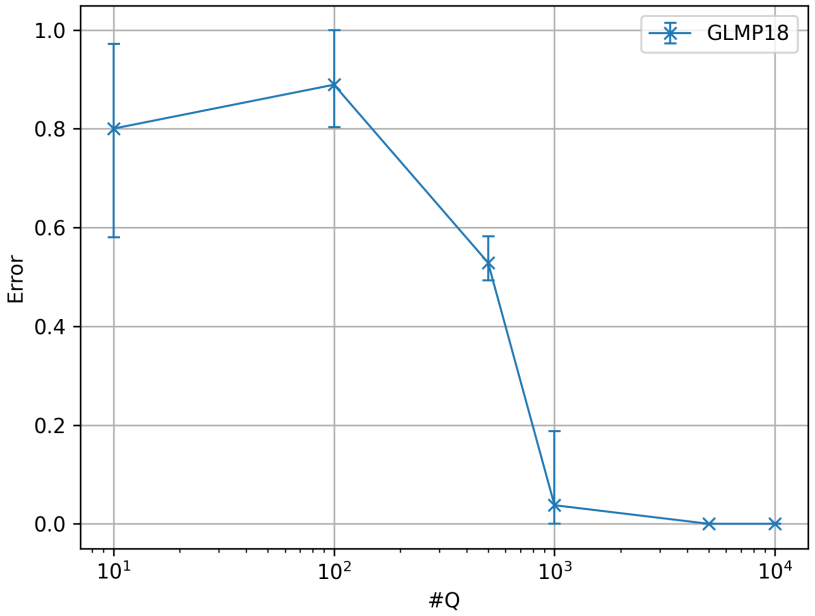
1. Subset of HCUP Data collection

2. Pick Artificial query distribution (Uniform/Zipf/...)

3. Evaluate for different amounts of queries




**or**



# Limitations & Contributions

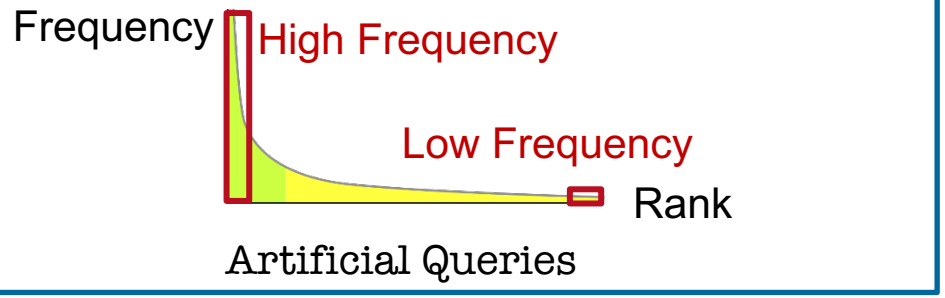
## Limitations

Systematization Lacking

Single Use Case

Few Comparisons

Closed-Source Code




Artificial Queries

## Our Contributions

Survey of ESA Cryptanalysis

New Attacks New Data

Systematic Re-Evaluation

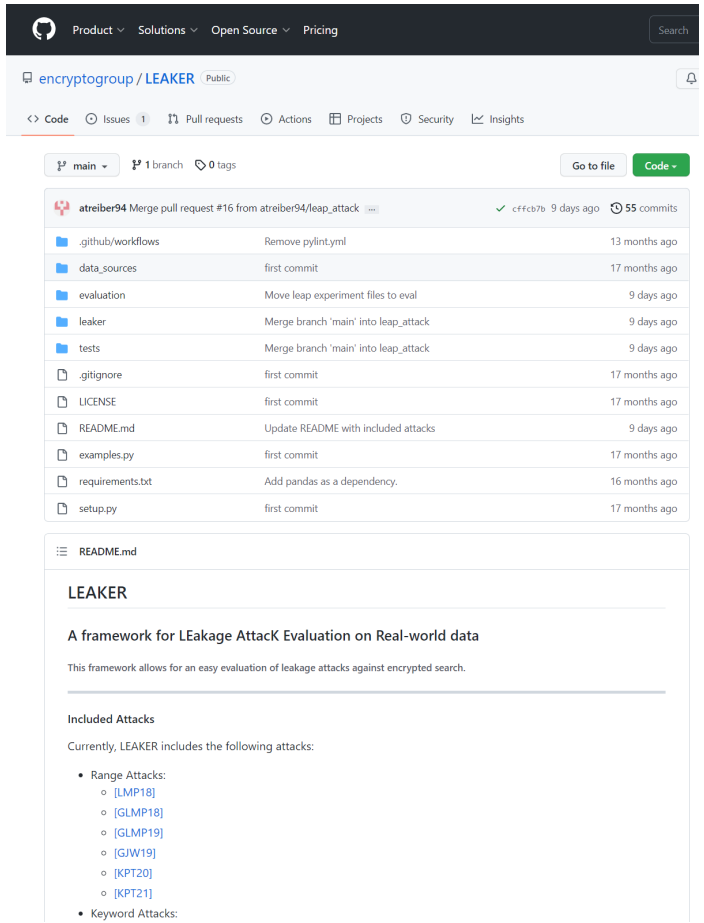
Open-Source Framework

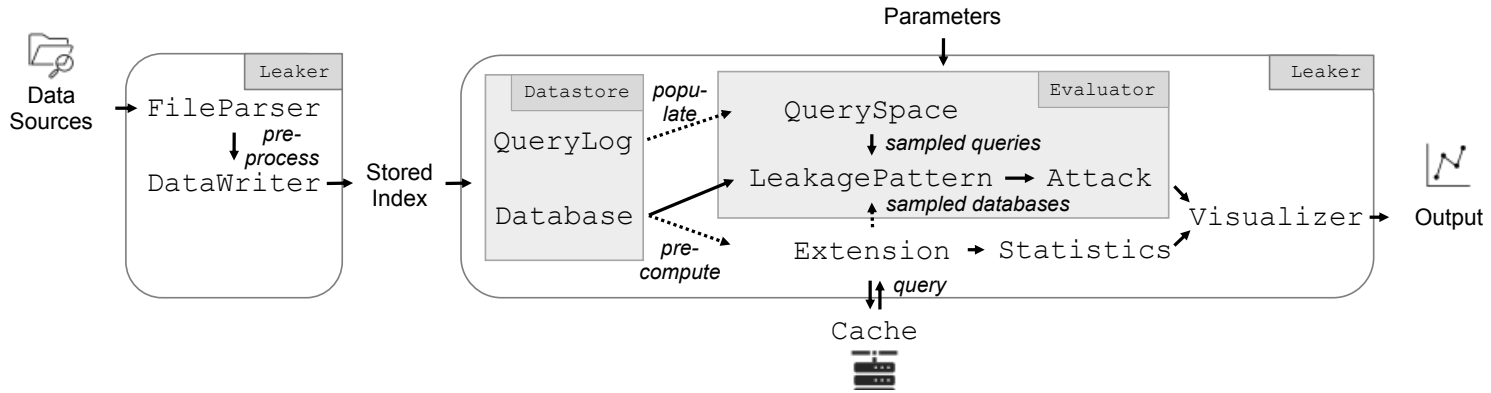
```
User,Query
216,'Aarhus'
216,'University'
106,'Visit'
216,'Cryptanalysis'
```

First Real-World Query Logs

# LEAKER Framework

https://encrypto.de/code/LEAKER




- Re-Implementation of major attacks in open-source Framework

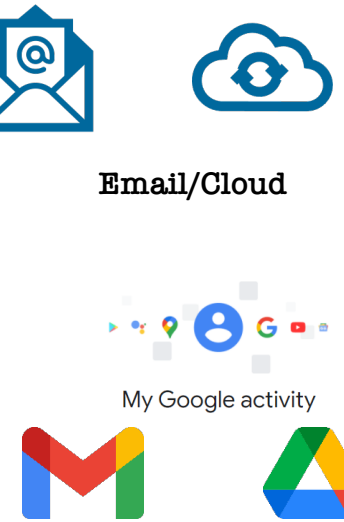
  On Release: [ IKK12, CGPR15, LMP18,GLMP18, GLMP19, GJW19, BKM20,KPT20,KPT21,RPH21 ]
  Since then: [KPT19,FMA+20,NHP+21,Sie22]
  In development: [OK21,DHP21,OK22,???]

- Modular design & supports interoperability
- Easy to implement new attacks & Countermeasures
- Easy to pre-process & use new data types.

# Data Sources

## **Keyword** Queries

### Email/Cloud




7 Query Logs &
7 Data Collections
7 Users
16-100 Queries
200-47k Documents
19k-895k Keywords

### Web


1 Query Log &
1 Data Collection
656k Users
2.9M Queries
151k Documents
268k Keywords

### Genetic


1 Query Log &
1 Data Collection
1.3k Users
54k Queries
115k Documents
690k Keywords

## **Range** Queries

### Scientific


3 Query Logs &
1 Data Collection
3 Users
215-8k Queries
5M Records
Domain N = 10k
Density 96%

### Medical


3 Data
Collections
2k-8k Records
Domain N =
73 – 10k
Density 3.3%-
81%

### Salaries


1 Data
Collection
536 Records
Domain N =
395
Density 2.3%

### Sales


1 Data
Collection
143 Records
Domain N =
6.3k
Density 2.3%

### Insurance


1 Data
Collection
886 Records
Domain N =
25k
Density 1.2%

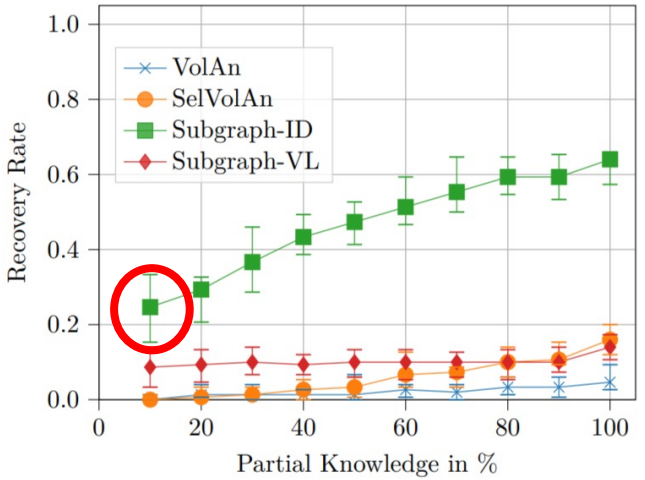
# Evaluation Summary

None of the attacks worked against low-[frequency] keywords

[BKM20]

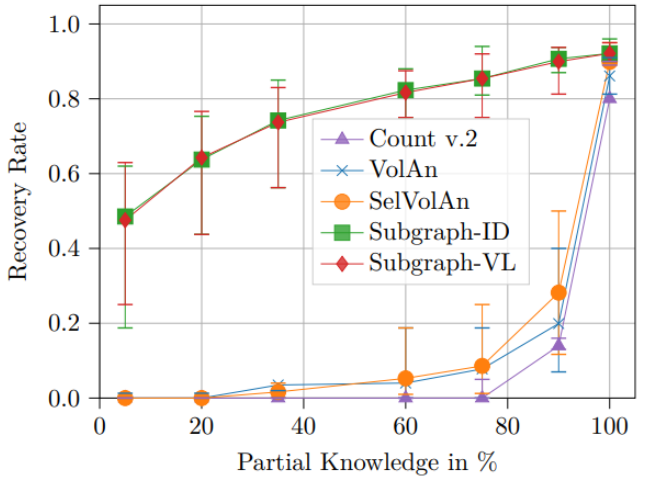
Users are more likely **to search for a specific email**

[RPH21]




**(Lowest) Mean Frequency: 1.54!**

(On the TAIR Dataset)




**Mean Frequency:** 326!

(On the Gmail Datasets)


[BKM20] L. Blackstone, S. Kamara, T. Moataz. Revisiting leakage abuse attacks. NDSS'20

[RPH21] R.G. Roessink, A. Peter, F. Hahn. Experimental review of the IKK query recovery attack: Assumptions, recovery rate and improvements. ACNS'21

## Evaluation Summary (Keyword Search)

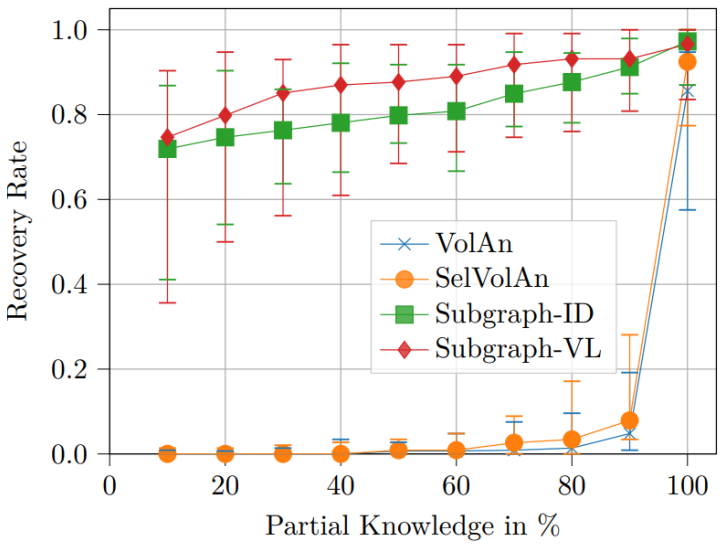
(subjective)

| Attacks | Leakage | Success Cases | Risk |
|---|---|---|---|
| • VolAn [BKM20]<br>• SelVolAn [BKM20] | • Response length<br>• Response volume | • High adversarial knowledge | Low |
| • [IKK12]<br>• Count V.2 [CGPR15]<br>• DetIKK [RPH21] | • Co-occurrence | • High adversarial knowledge | Low |
| • SubgraphID [BKM20]<br>• SubgraphVL [BKM20] | • Response identities<br>• Response volumes | • Low adversarial knowledge | High |

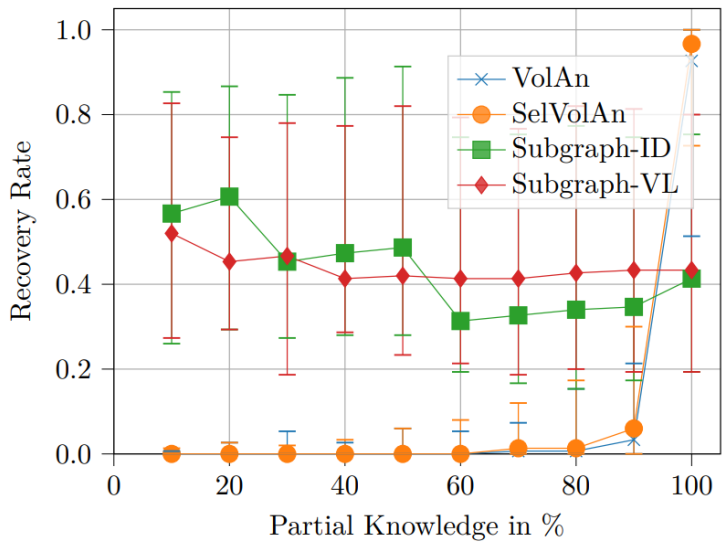
**=> Suppression of identifier and volume leakage of responses necessary!**

# Evaluation Summary (Keyword Search)

**AOL**
**single user & low frequency**

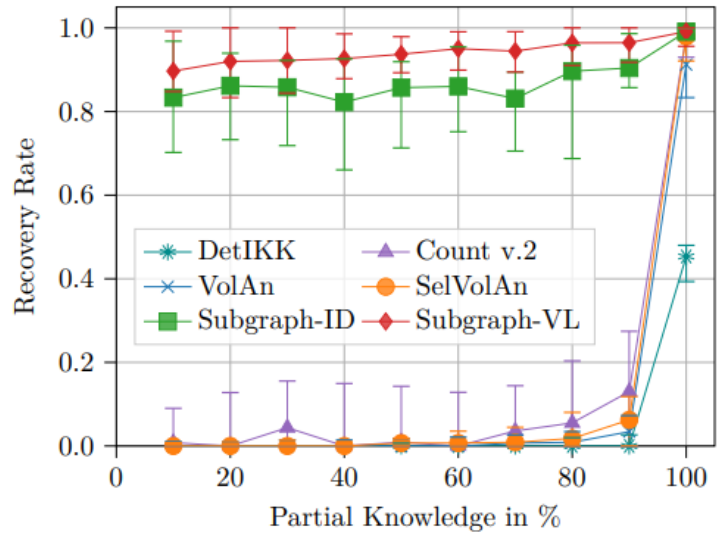



including queries outside of partial knowledge




with repeating queries

**AOL**
**single user & high frequency**



## Evaluation Summary (Range Search)

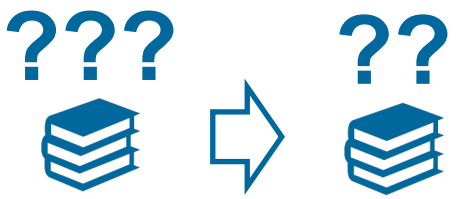
(subjective)

| Attacks | Leakage | Success Cases | Risk |
|---|---|---|---|
| • [GLMP18]<br>• [GJW19] | • Response length | • None | Very low |
| • APA [KPT21] | • Response length<br>• Query equality | • Evenly distributed data | Medium |
| • [LMP18] | • Response identities | • Dense | Medium |
| • GenKNNO [GLMP19]<br>• ApprValue [GLMP19]<br>• ARR [KPT20] + ApprOrder [GLMP19] | • Response identities | • Large widths<br>• Skewed values | Medium |
| • ARR [KPT20] | • Response identities<br>• Order | • Most cases | High |

**=> Research on order reconstruction + Leakage suppression for range case!**

# Nuanced highlights given LEAKER

- [BKM20] attacks on identifier or volume leakage work much better than previously anticipated
- [IKK12,CGPR15] keyword attacks perform much worse than previously anticipated
- Range attacks rarely work on our data and success highly depends on query/data distributions
- [OK22] attacks recovery rate given a specific leakage profile highly depends on prior assumption over query/data
- ESA cryptanalysis is very nuanced



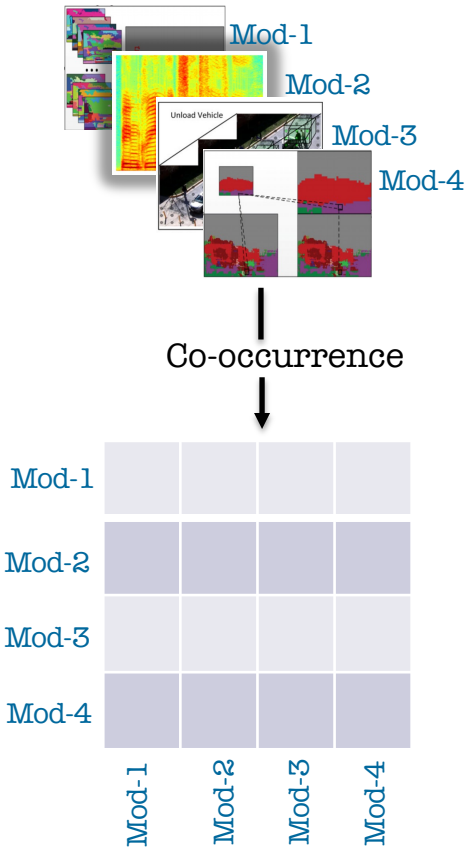

[BKM20] L. Blackstone, S. Kamara, T. Moataz. Revisiting leakage abuse attacks. NDSS'20
[IKK12] M. S. Islam, M. Kuzu, M. Kantarcioglu. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. NDSS'12
[CGPR15] D. Cash, P. Grubbs, J. Perry, T. Ristenpart. Leakage-abuse attacks against searchable encryption. CCS'15
[OK22] S. Oya and F. Kerschbaum. IHOP: Improved Statistical Query Recovery against Searchable Symmetric Encryption through Quadratic Optimization. USENIX'22

# Statistical query recovery attacks

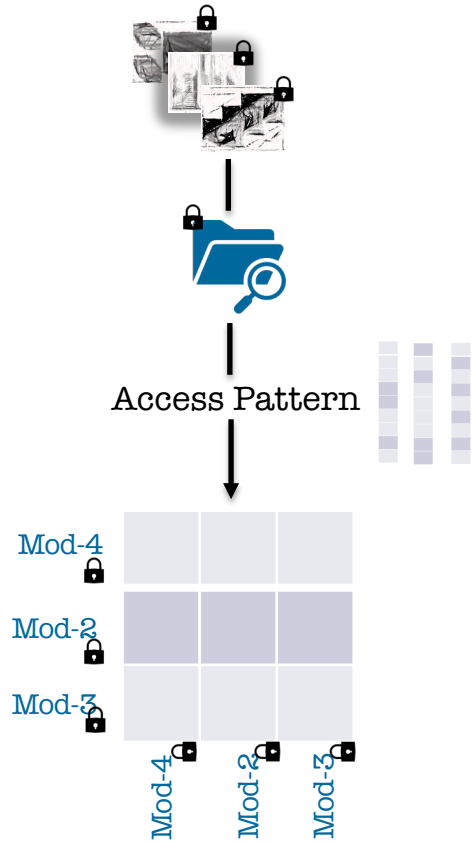
Statistical-based query recovery attacks achieve [lower] accuracy and are [not] considered a serious threat.

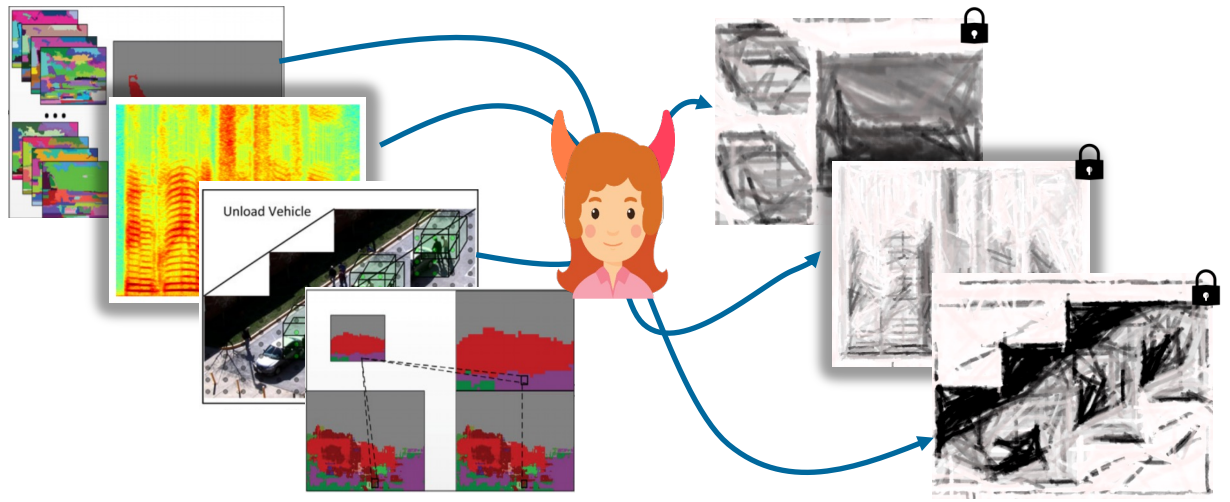
[OK22]

## Auxiliary Information

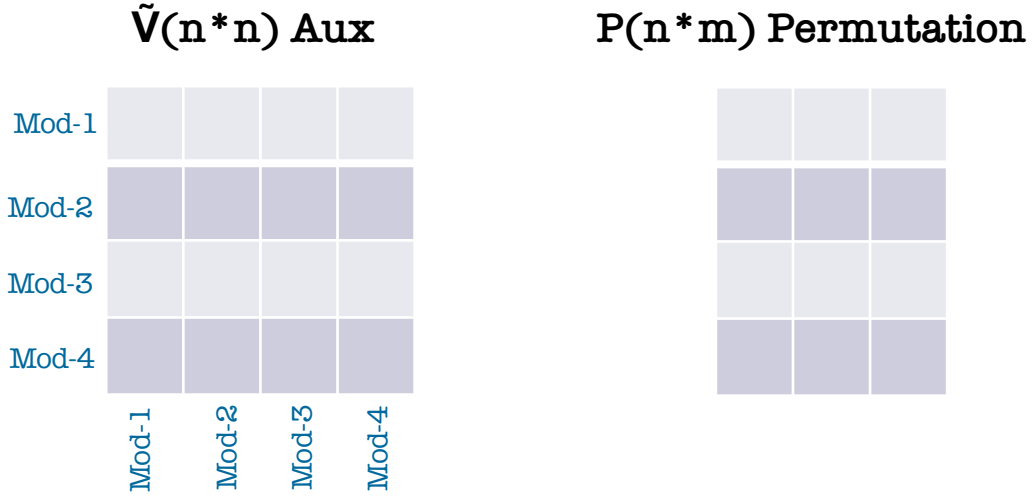



## Observations



# Statistical query recovery attacks

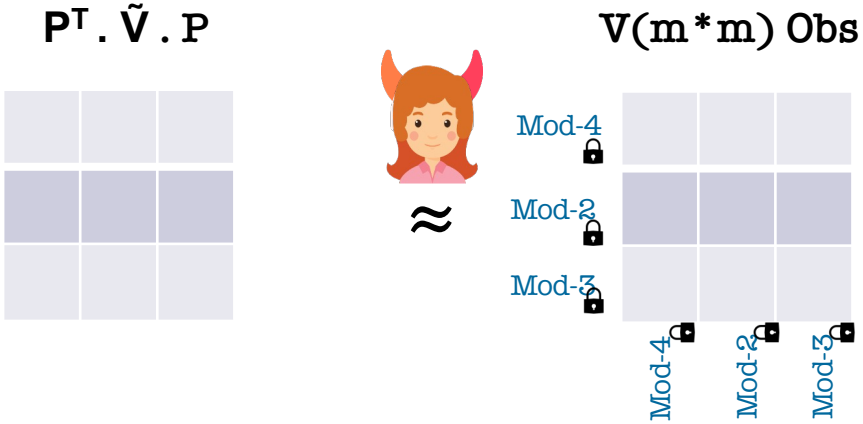



The query recovery is formulated as a quadratic assignment problem and iteratively solved via linear assignments.

[OK22]

Q.A.P

$$\mathbf{P} = \arg\min_{\mathbf{P}\in\mathcal{P}} \sum_{i,i'\in[n]} \sum_{j,j'\in[m]} c_{i,i',j,j'} \cdot \mathbf{P}_{i,j} \cdot \mathbf{P}_{i',j'}$$



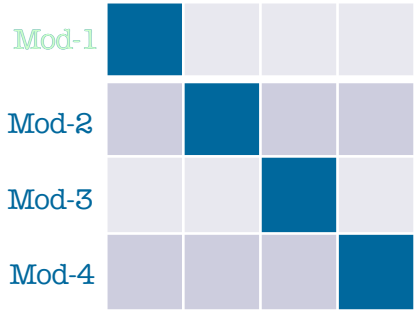

Examples:

- IKK : P = argmin $||\tilde{V} - P^T . \tilde{V} . P||_2$
  --> simulated annealing
- graphM : P = argmin $||\tilde{V} - P^T . \tilde{V} . P||^2_2 - tr(CP)$
  --> convex-concave rel.

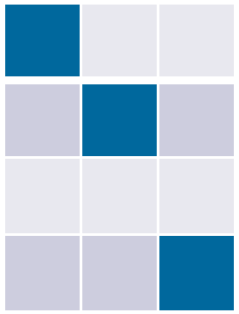

[IKK] Islam et .al. Access pattern disclosure on searchable encryption: ramifications, attacks and mitigation. NDSS12.
[graphM] Pouliot and wright. The shadow nemesis: inference attacks on efficiently deployable, efficiently searchable encryption. CCS16.

# Statistical query recovery attacks

**Ṽ(n*n) Aux**




**P(n*m) Permutation**



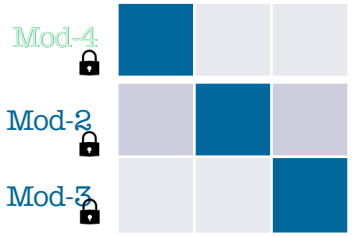
$$\mathbf{P} = \arg\min_{\mathbf{P}\in\mathcal{P}} \sum_{i\in[n]} \sum_{j\in[m]} c_{i,j,} \cdot \mathbf{P}_{i,j} .$$

L.A.P

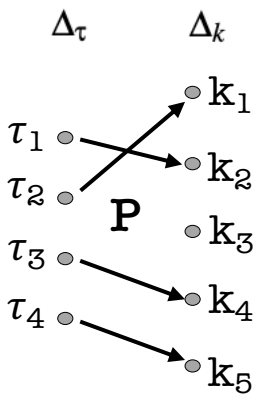
This very efficient, but a lot of information is wasted because of not using the off-diagonal terms.

| | 0,5 | 0,51 | 0,18 | 0,2 | |
|---|---|---|---|---|---|
| | 0,01 | 0,02 | 0,31 | 0,29 | 0,49 |
| | 0,01 | 0 | 0,33 | 0,31 | 0,51 |
| | 0,3 | 0,31 | 0,02 | 0 | 0,2 |

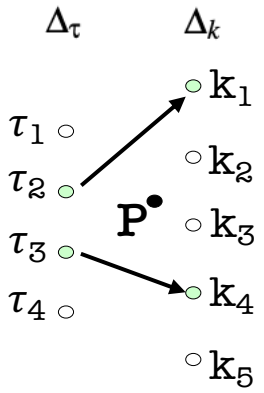
**V(m*m) Obs**



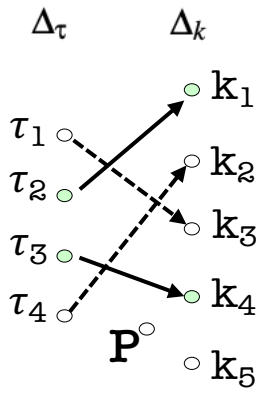

**Hungarian algorithm**



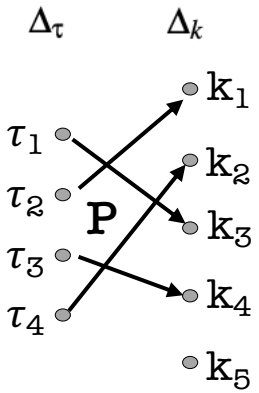

Full Assignment




Select fixed mappings




Compute mapping for free tokens L.A.P/Freeze



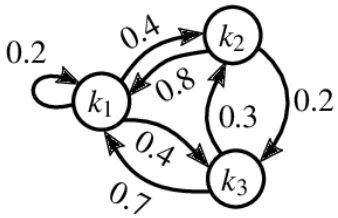

After iteration Solve L.A.P

$$\Delta_\tau^\circ = \{\tau_1, \tau_4\}\ \Delta_\tau^\bullet = \{\tau_2, \tau_3\}\ \ \Delta_k^\circ = \{k_2, k_3, k_5\}\ \Delta_k^\bullet = \{k_1, k_4\}$$

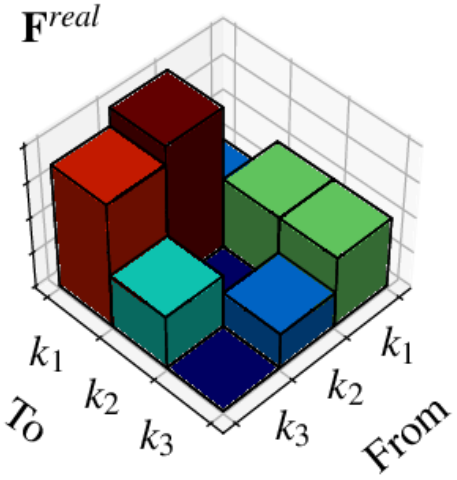
# Statistical query recovery attacks

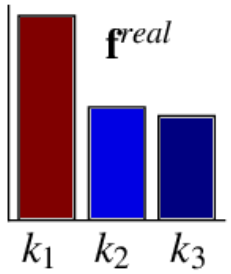
Adversary can exploit Qeq in the dependent setting where the client's queries are correlated, even when access obfuscation defenses are applied.

[OK22]

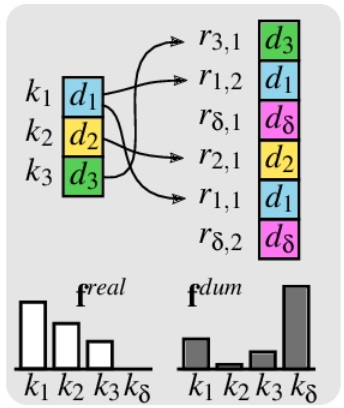



Markov Model

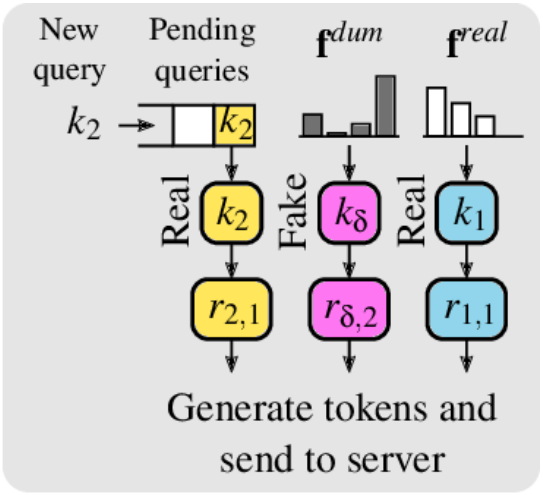


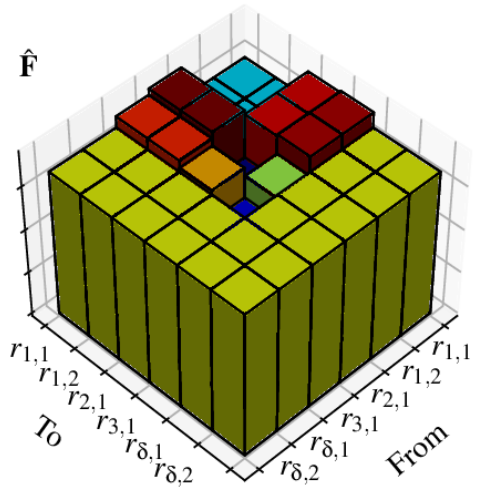

Markov matrix ($\mathbf{F}^{real}$) and its stationary distribution ($\mathbf{f}^{real}$) of the queried keywords.




PANCAKE setup.




PANCAKE query.




Markov matrix ($\hat{F}$) of the queried replicas by following PANCAKE protocol.

# Statistical query recovery attacks

## Sorting

Step 1 :
- Initializes an empty mapping

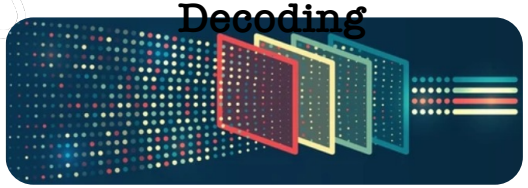
Step 2 :
- Computes the stationary distribution $\pi$,

Step 3 :
- Calculate the histogram of the sequence of queries v.
  - ≈ to the average number of visits over the M.C states)

Step 4 :
- Map the closest value in $\pi$ to $vi$, for all $i \in [t]$;
  - the average number of visits to the $i^{th}$ state is approximately equal to the $i^{th}$ component of the stationary distribution $\pi$.

Step 5:
- output the mapping and the error score
  - Error: the total distance between the avg.# visits and the selected component of the stationary distribution

## Decoding



Step 1 :
- Initializes an empty mapping

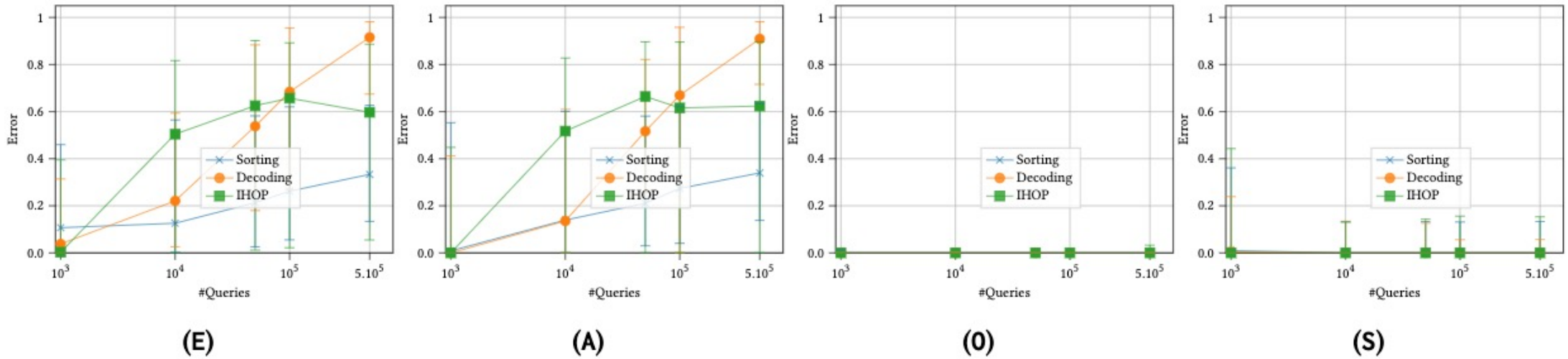
Step 2 :
- Computes the Observation matrix of HMM O=($o_{i,j}$) $i \in [m], j \in$ [#I],
  - The frequency f$j$, of each unique query $j \in$ [#I], is first calculated using query equality leakage.
  - Set $o_{i,j}$ to 1−|$f_j$−$\pi_i$| i.f.f $|f_j-\pi_i|_1 < \epsilon$ , error parameter.
  - Normalize O, s.t the sum of each row is equal to 1.

Step 3:
- Compute transition matrix $P^A$ and a uniform initial distribution $\mu$ to form HMM parameters Θ:=($P^A$,O, $\mu$).

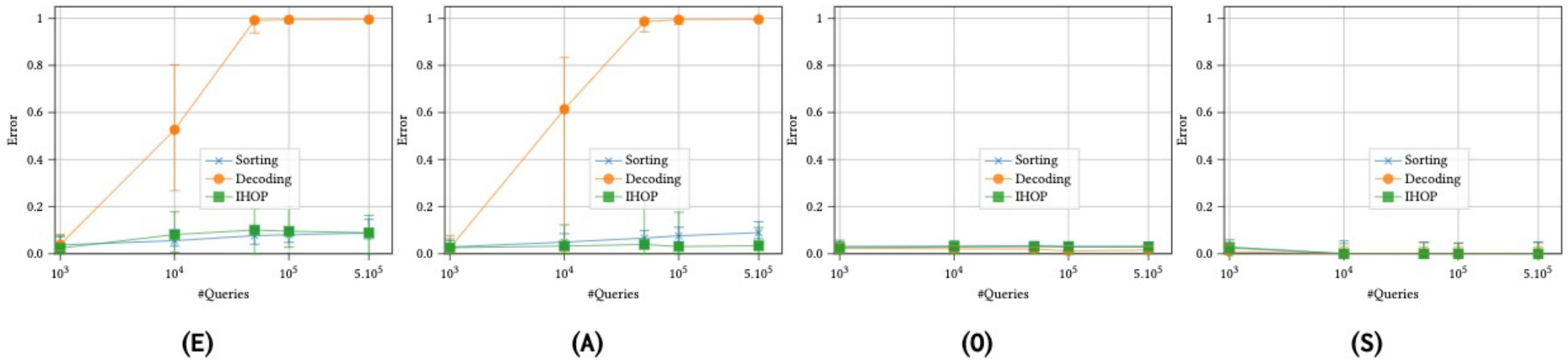
Step 4 :
- (Mapping $\alpha$ the attacked query sequence to the state identifiers of unique queries via the equality leakage, the likelihood $s$ of this mapping given the observation ) ← Viterbi .
  - Generate a sequence of observed states that matches the set of observation states of the created HMM parameters

Step 5 :
- A new map $\alpha'$translates the states $\alpha$ maps to actual keywords using the adversary's knowledge.
  - error parameter, we set $s'=1-s$ such that the result with the maximum likelihood will correspond to the lowest score.

# Evaluation results (R.W Q-log )



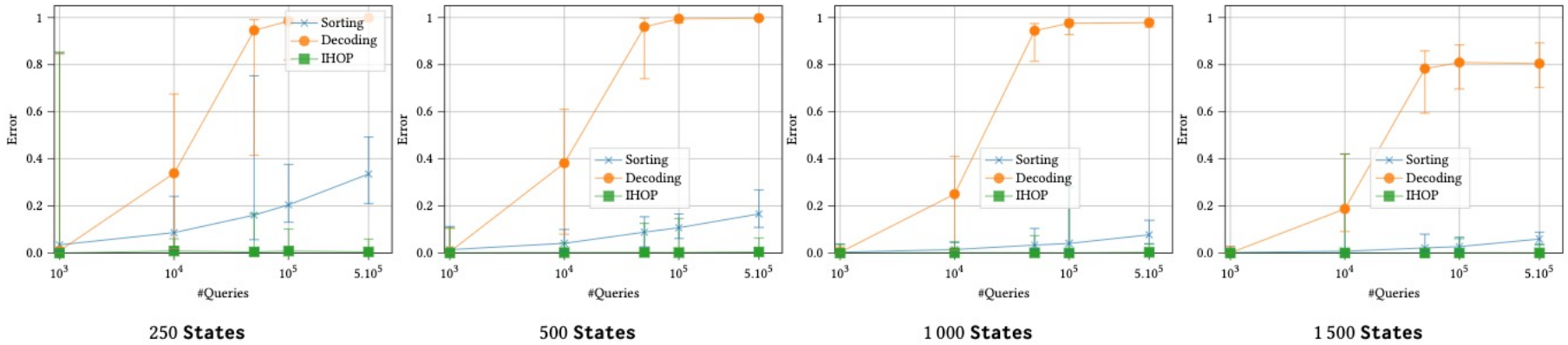

Evaluation for each of 5 users on AOL



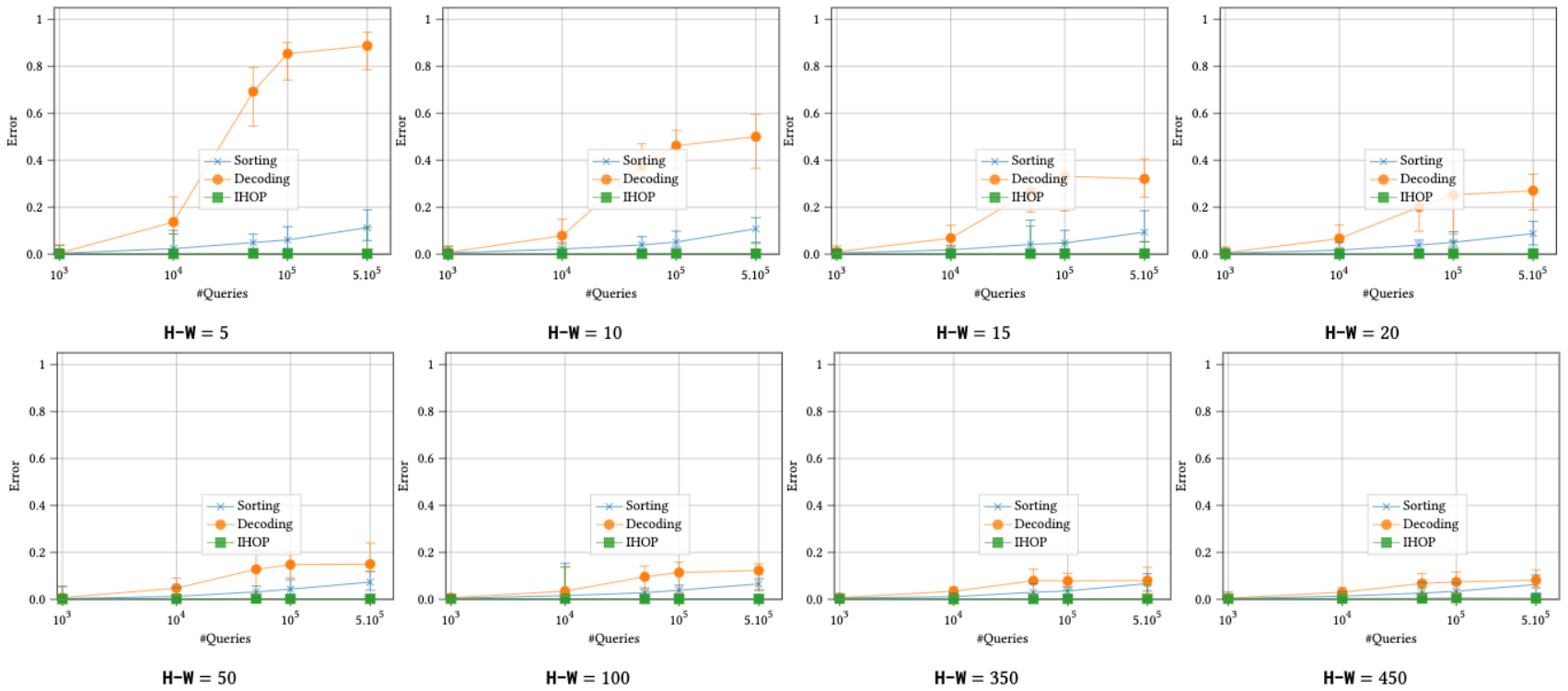

Evaluation for each of 5 users on TAIR

# Evaluation results (Art.Distributions )

Evaluation for Zipf-Zipf Artificial distribution with fixed H-W

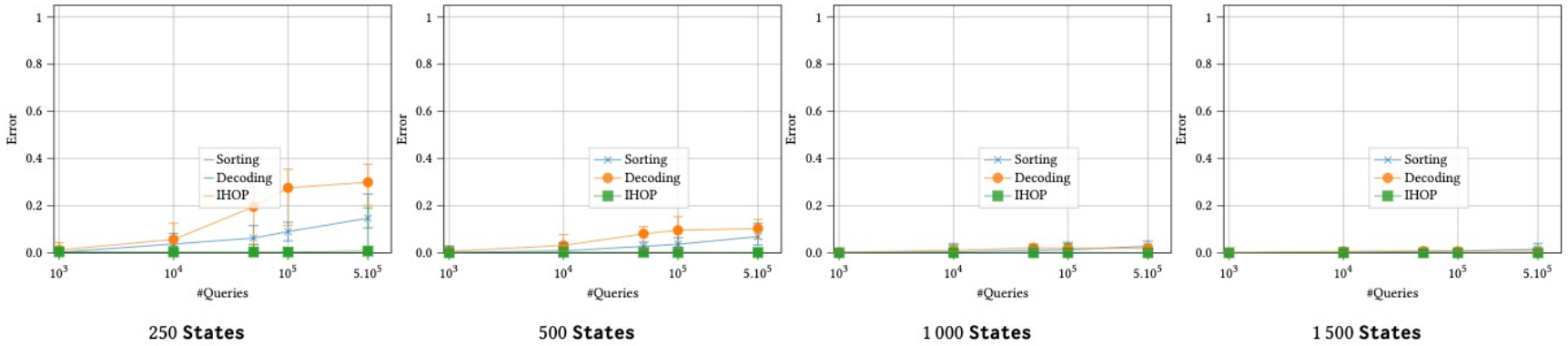



Evaluation for Zipf-Zipf Artificial distribution with variable H-W

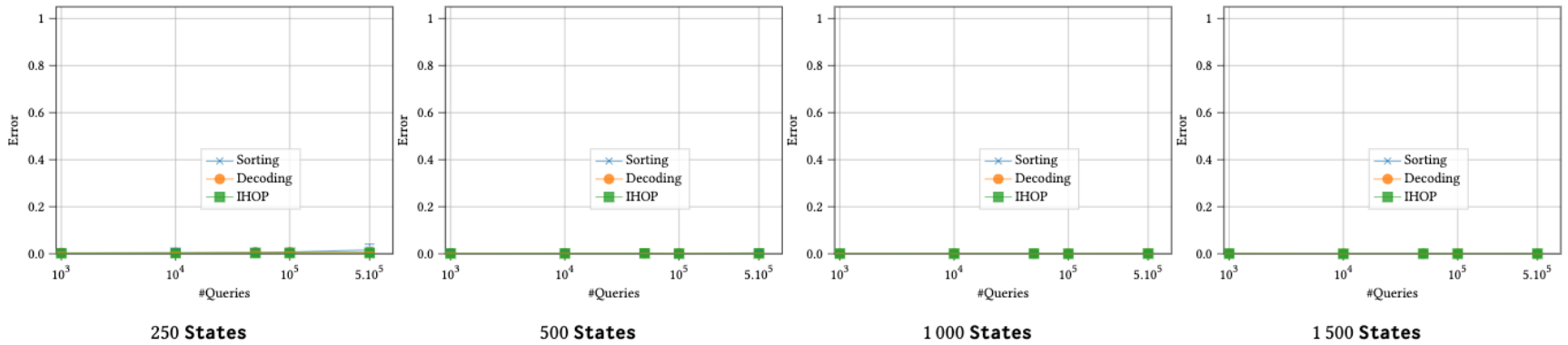


# Evaluation results (Art.Distributions )

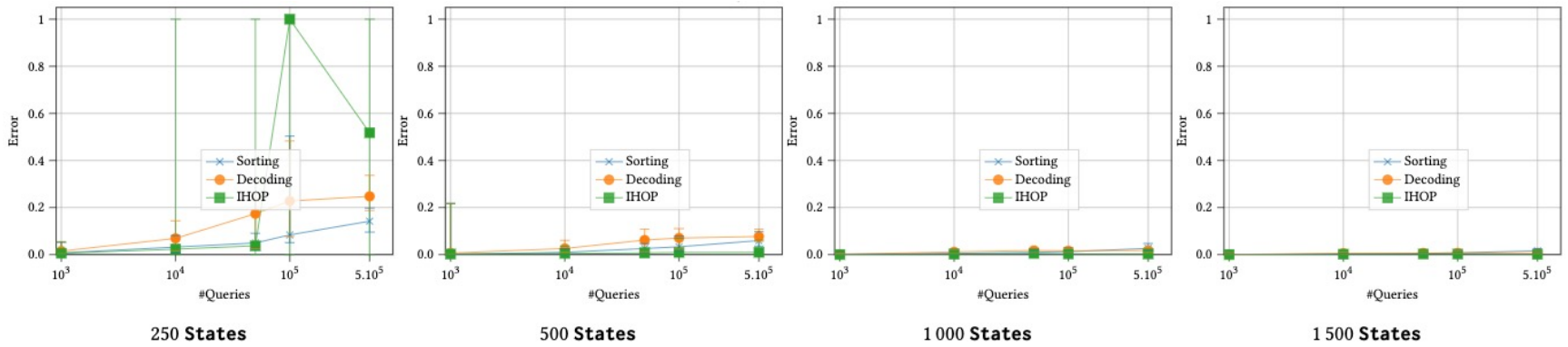
Evaluation for *Erdos* Artificial Distribution.

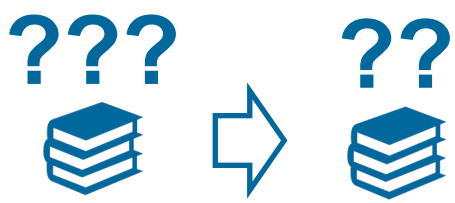



Evaluation for *Uniform* Artificial distribution.




Evaluation for *Zipf* Artificial distribution.




Thank you
for your attention

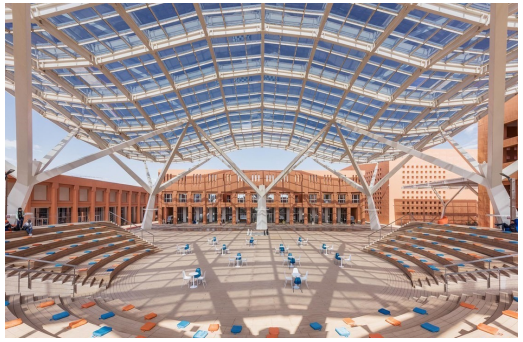
# Cryptanalysis Strikes Back

A Realistic assessment of leakage attacks on Encrypted Search

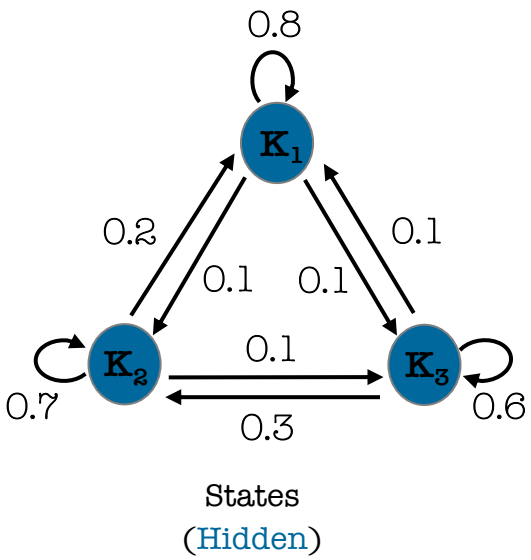
**Abdelkarim Kati**†‡
together with T. Moataz, S. Kamara and A. Treiber.

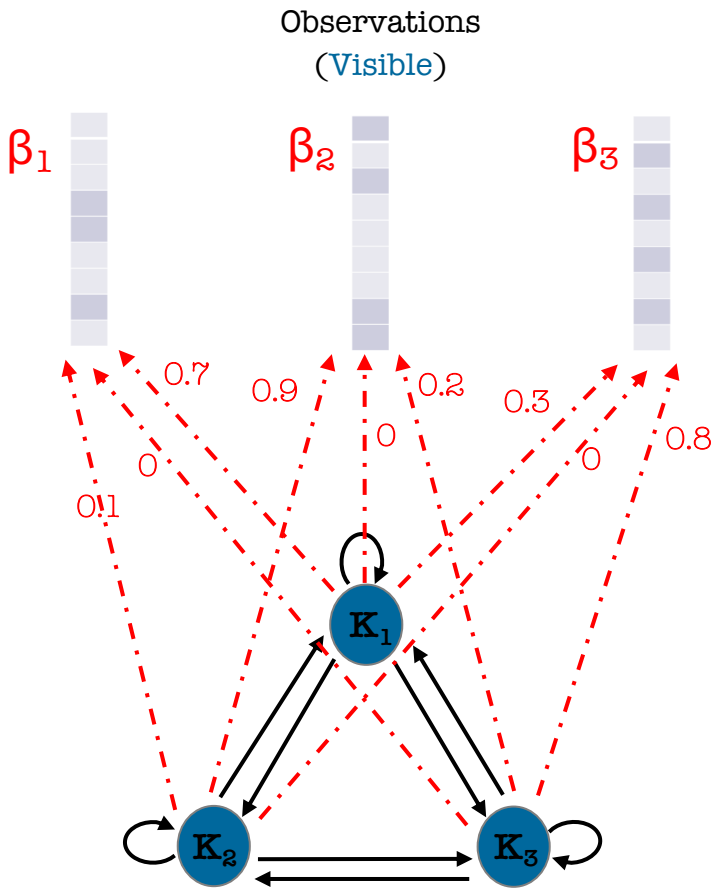
†School of Computer Science,
Mohammed VI Polytechnic University.
‡ Encrypted Systems Lab, Brown University.

January 24, 2023 at Aarhus University.



# Viterbi Algorithm (Uncovering Problem)




| | K1 | K2 | K3 |
|---|---|---|---|
| K1 | 0.8 | 0.1 | 0.1 |
| K2 | 0.2 | 0.7 | 0.1 |
| K3 | 0.1 | 0.3 | 0.6 |

State transition probabilities

| | K1 | K2 | K3 |
|---|---|---|---|
| | 0.6 | 0.2 | 0.2 |

Initial state probabilities

| | β1 | β2 | β3 |
|---|---|---|---|
| K1 | 0.7 | 0 | 0.3 |
| K2 | 0.1 | 0.9 | 0 |
| K3 | 0 | 0.2 | 0.8 |

Emission probabilities

* MAPLE: Markov Process Leakage attacks on Encrypted search (under submission)

# Viterbi Algorithm (Uncovering Problem)

**Input**

Observation Sequence

$O = ( o_1 , o_2, o_3, o_4, o_5, o_6 )$



$\beta_1$ $\beta_3$ $\beta_1$ $\beta_3$ $\beta_3$ $\beta_2$

**Viterbi**

| | $o_1=\beta_1$ | $o_2=\beta_3$ | $o_3=\beta_1$ | $o_4=\beta_3$ | $o_5=\beta_3$ | $o_6=\beta_2$ |
|---|---|---|---|---|---|---|
| $K_1$ | 0.4200 | 0.1008 | 0.0564 | 0.0135 | 0.0033 | 0 |
| $K_2$ | 0.200 | 0 | 0.0010 | 0 | 0 | 0.0006 |
| $K_3$ | 0 | 0.0336 | 0 | 0.0045 | 0.0022 | 0.0003 |

Accumulated probability matrix

| | $o_1=\beta_1$ | $o_2=\beta_3$ | $o_3=\beta_1$ | $o_4=\beta_3$ | $o_5=\beta_3$ |
|---|---|---|---|---|---|
| $K_1$ | 1 | 1 | 1 | 1 | 1 |
| $K_2$ | 1 | 1 | 1 | 1 | 3 |
| $K_3$ | 1 | 3 | 1 | 3 | 3 |

$I_6 = 2$

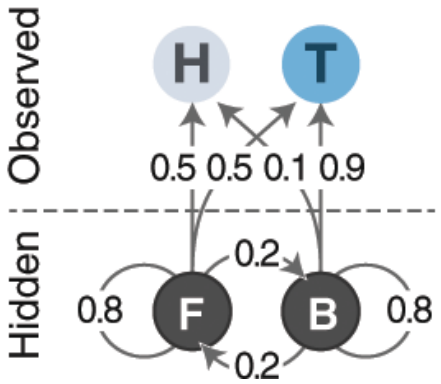
Backtracking matrix

**Output**

Observation Sequence

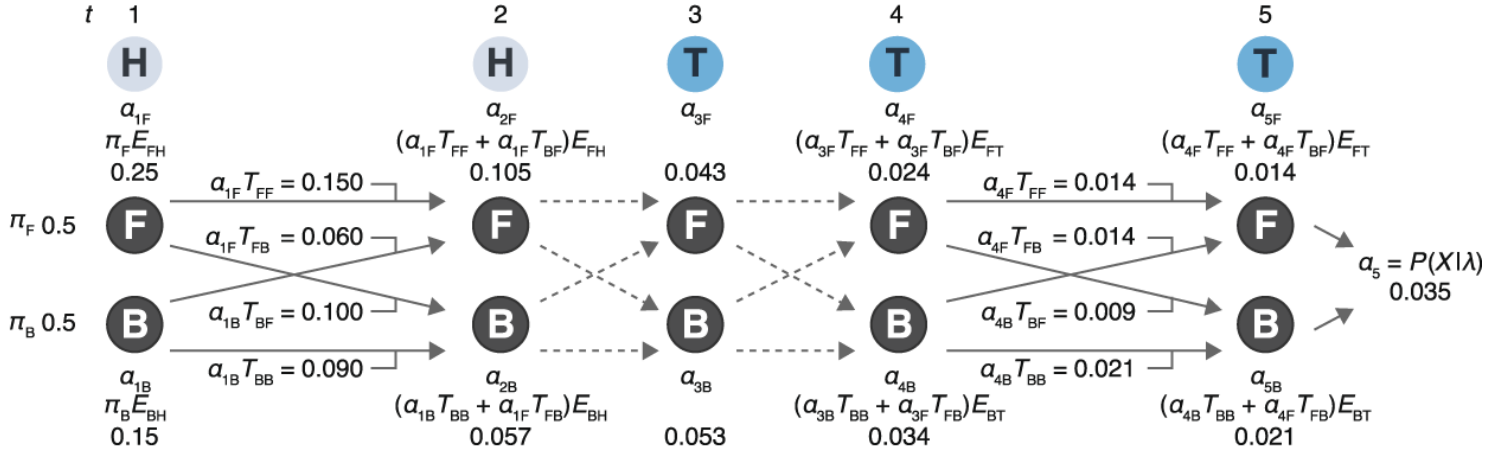
$S^* = ( K_1 , K_1, K_1, K_3, K_3, K_2 )$

# Baum-Welch Algorithm (Estimation Problem)




Hidden Markov Model
of an unstable coin

Forward probability




Ground truth

$$E = \begin{array}{c} \\ F \\ B \end{array}\begin{array}{c} \begin{array}{cc} H & T \end{array} \\ \begin{bmatrix} 0.5 & 0.5 \\ 0.1 & 0.9 \end{bmatrix} \end{array}$$

$$T = \begin{array}{c} \\ F \\ B \end{array}\begin{array}{c} \begin{array}{cc} F & B \end{array} \\ \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix} \end{array}$$

Initial estimates

$$\hat{E}_0 = \begin{array}{c} \\ F \\ B \end{array}\begin{array}{c} \begin{array}{cc} H & T \end{array} \\ \begin{bmatrix} 0.5 & 0.5 \\ 0.3 & 0.7 \end{bmatrix} \end{array}$$

$$\hat{T}_0 = \begin{array}{c} \\ F \\ B \end{array}\begin{array}{c} \begin{array}{cc} F & B \end{array} \\ \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix} \end{array}$$

HMM true parameters
And initial estimations

Backward probability