SEAL[1]:Mitigating Attacks on Encrypted Databases via Adjustable Leakage

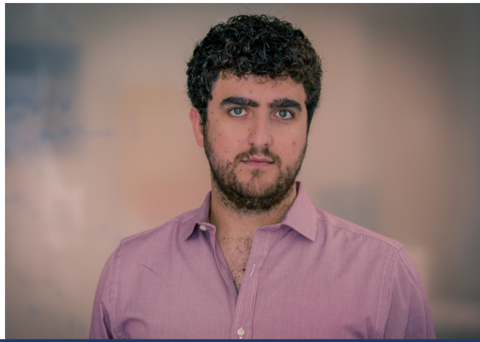
[1] **S**earchable **E**ncryption with **A**djustable **L**eakage

Ioannis Demertzis
University of Maryland
yannis@umd.edu

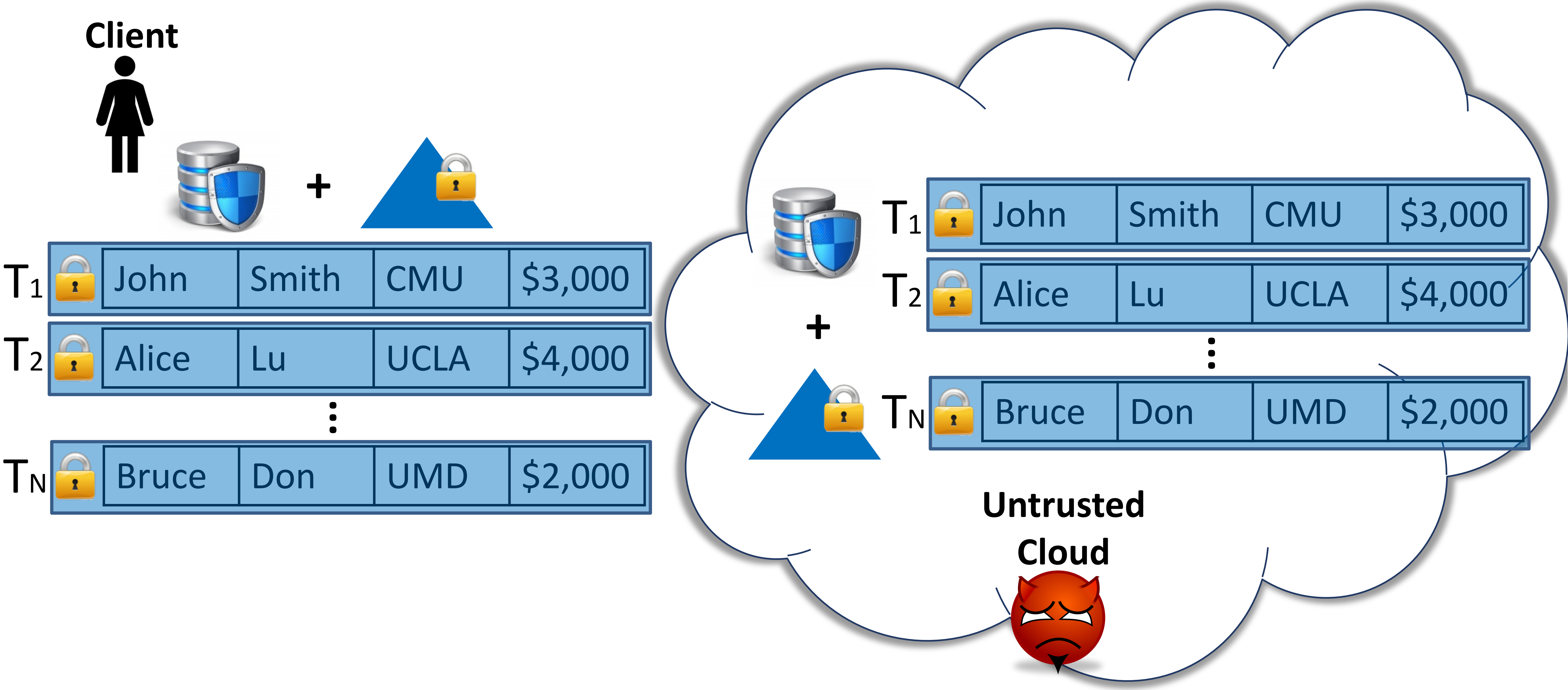
Dimitrios Papadopoulos
HKUST
dipapado@cse.ust.hk

Charalampos Papamanthou
University of Maryland
cpap@umd.edu

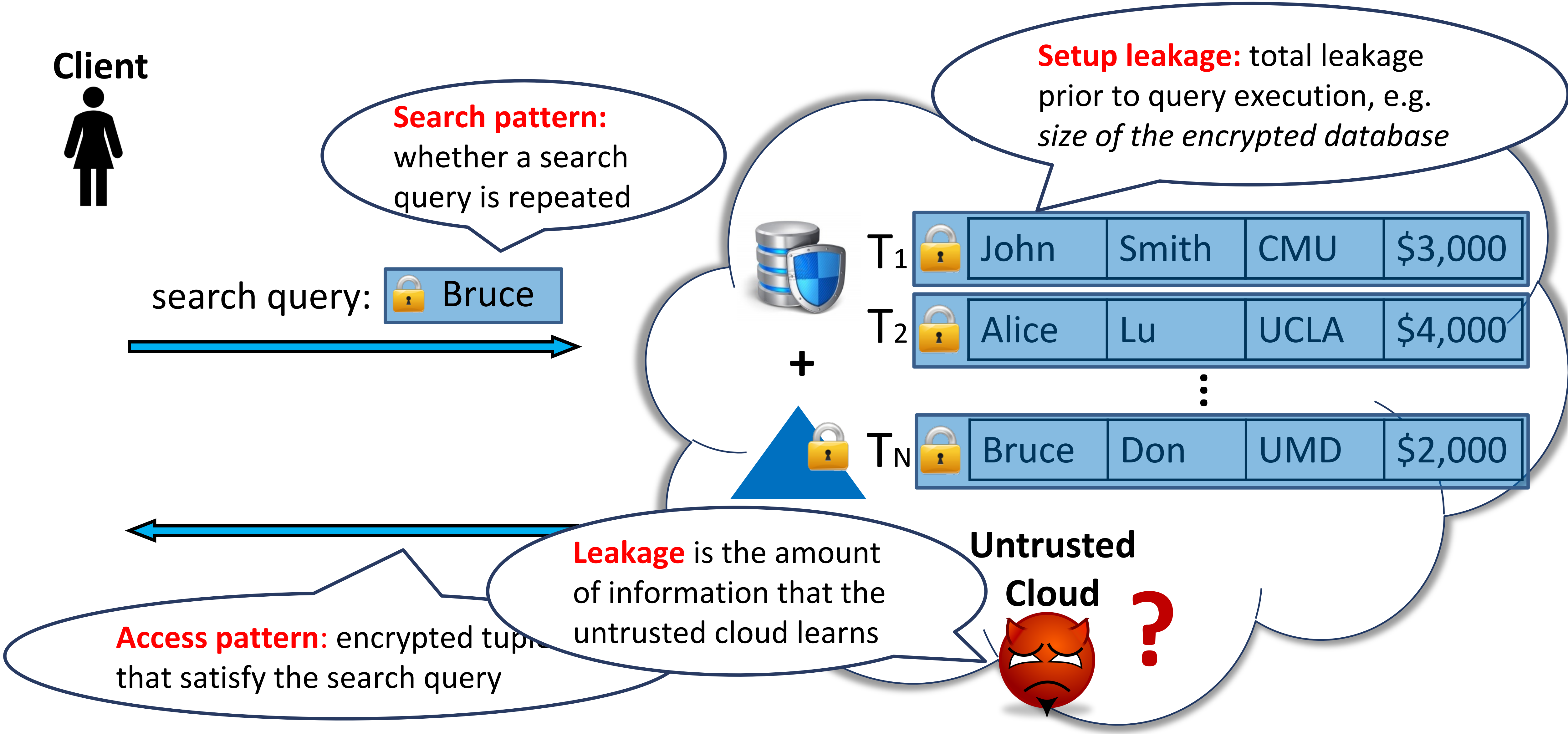
Saurabh Shintre
NortonLifeLock Research Group
saurabh.shintre@nortonlifelock.com



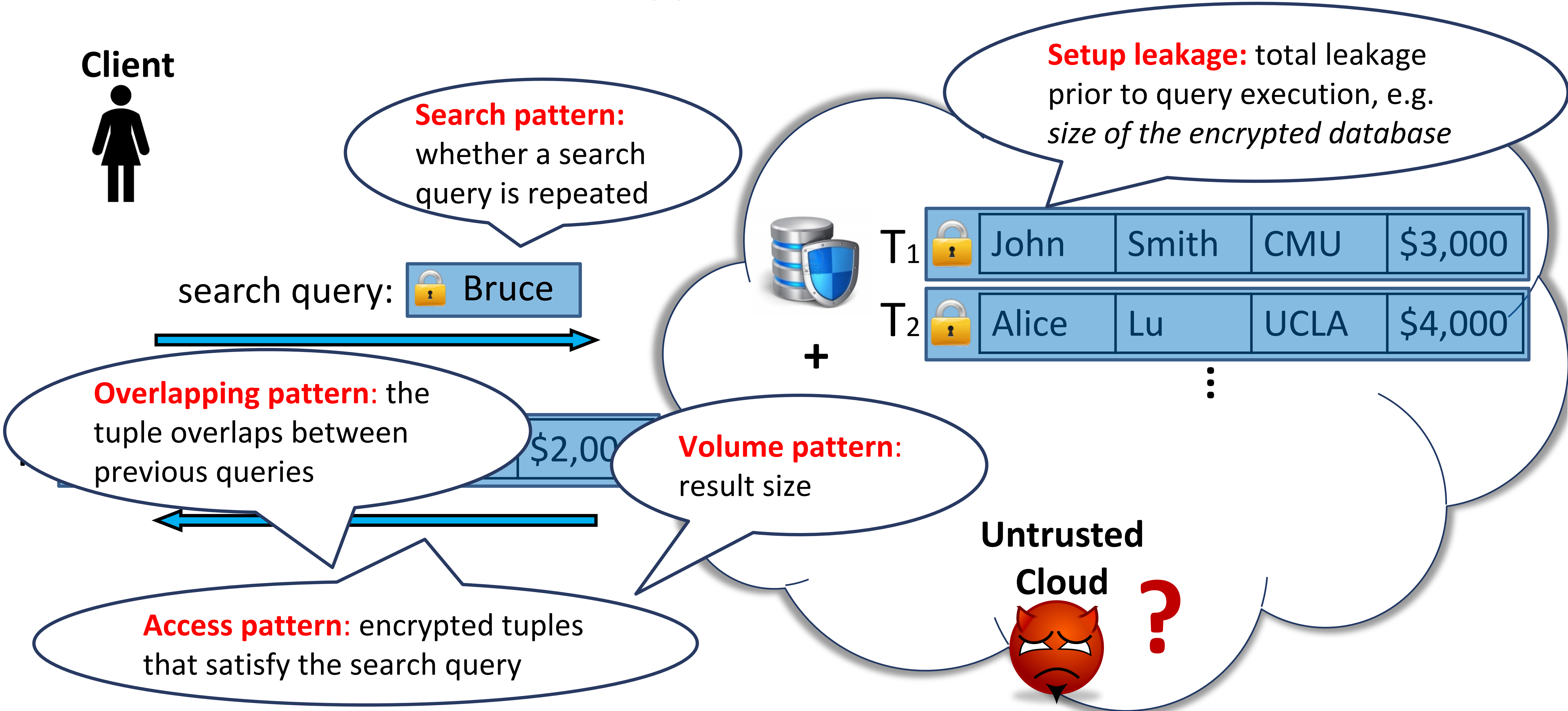
What is Searchable Encryption (SE)?



What is Searchable Encryption (SE)?



What is Searchable Encryption (SE)?



<u>Security (informal):</u> The adversary does not learn anything beyond the above leakages!

Attacks on SE

Search/ Overlapping Pattern + Volume Pattern

Keyword/Email Search

Islam et al. NDSS 2012
Cash et al. CCS 2015

Assume that the adversary knows a fraction N^{γ} ($\gamma \in [0,1]$) of the plaintext input

Search/ Overlapping

Keyword/Email Search

Zhang et al. USENIX 2016

Range Search

Dautrich et al. EDBT'13
Islam et al. CODAPSY'14
Kellaris et al. CCS 2016
Lacharite et al. S&P 2018
Grubbs et al. S&P 2019

kNN queries

Kornaropoulos et al. S&P 2019

Volume Pattern

Range Search

Kellaris et al. CCS 2016
Lacharité et al. S&P 2018
Grubbs et al. CCS 2018
Kornaropoulos et al. S&P 2020

Attacks on SE

Search/ Overlapping Pattern

Search/ Overlapping

Volume Pattern

Limitations of prior attacks:

i) Do not attack state-of-the-art schemes (e.g., range attacks)
ii) Assume that the attacker knows a great percentage of the input distribution
iii) Assume that the query distribution is known to the attacker
iv) Assume that the input database has a specific structure

Kornaropoulos et al. S&P 2019

Attacks on SE

?? Defenses ??

Search/ Overlapping Pattern
+
Volume Pattern

Keyword/Email Search

Islam et al. NDSS 2012
Cash et al. CCS 2015

Assume that the adversary knows a fraction N^{γ} ($\gamma \in [0,1]$) of the plaintext input

Search/ Overlapping

Keyword/Email Search

Zhang et al. USENIX 2016

Range Search

Dautrich et al. EDBT’13
Islam et al. CODAPSY’14
Kellaris et al. CCS 2016
Lacharite et al. S&P 2018
Grubbs et al. S&P 2019

kNN queries

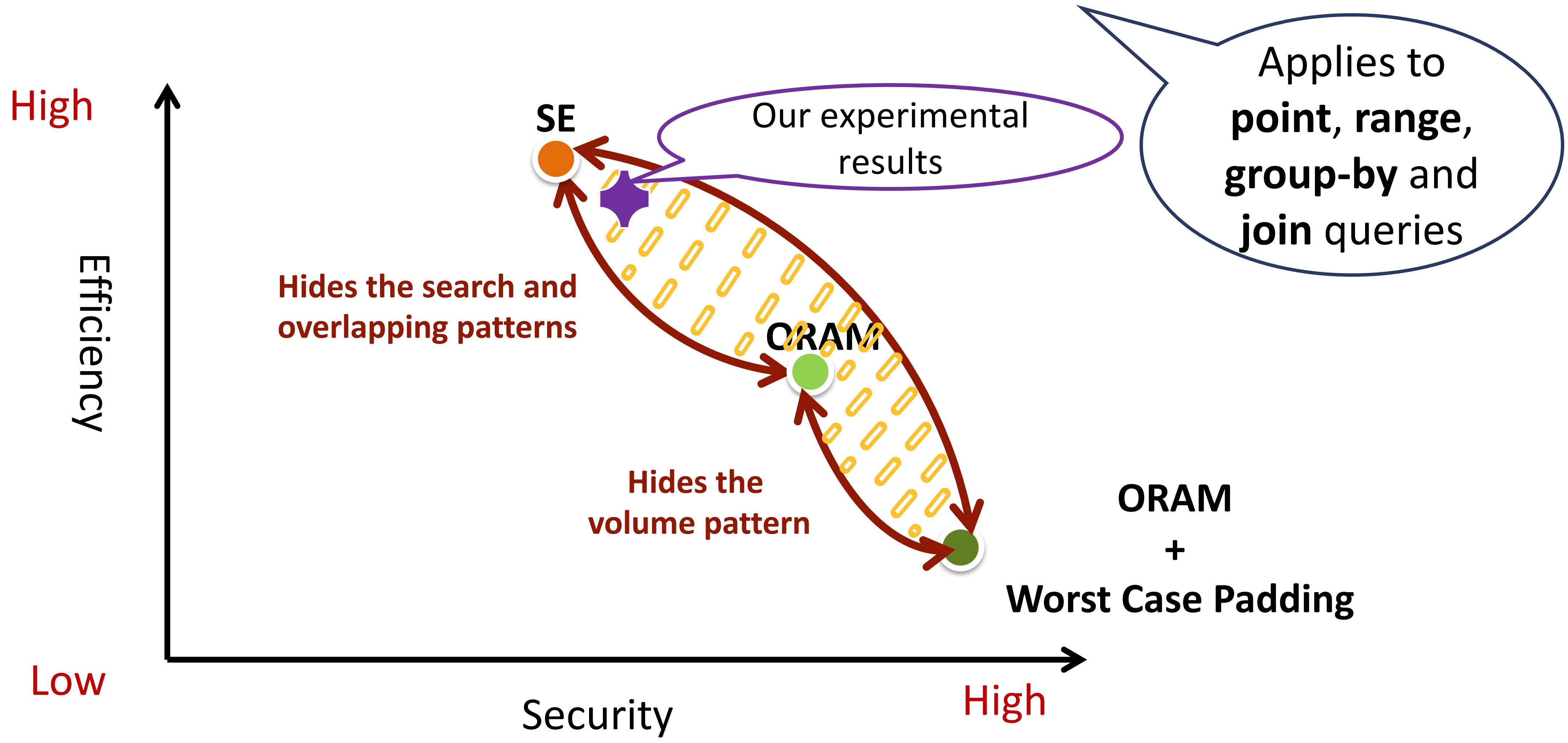
Kornaropoulos et al. S&P 2019

Volume Pattern

Range Search

Kellaris et al. CCS 2016
Lacharité et al. S&P 2018
Grubbs et al. CCS 2018
Kornaropoulos et al. S&P 2020

SEAL: Searchable Encryption with Adjustable Leakage



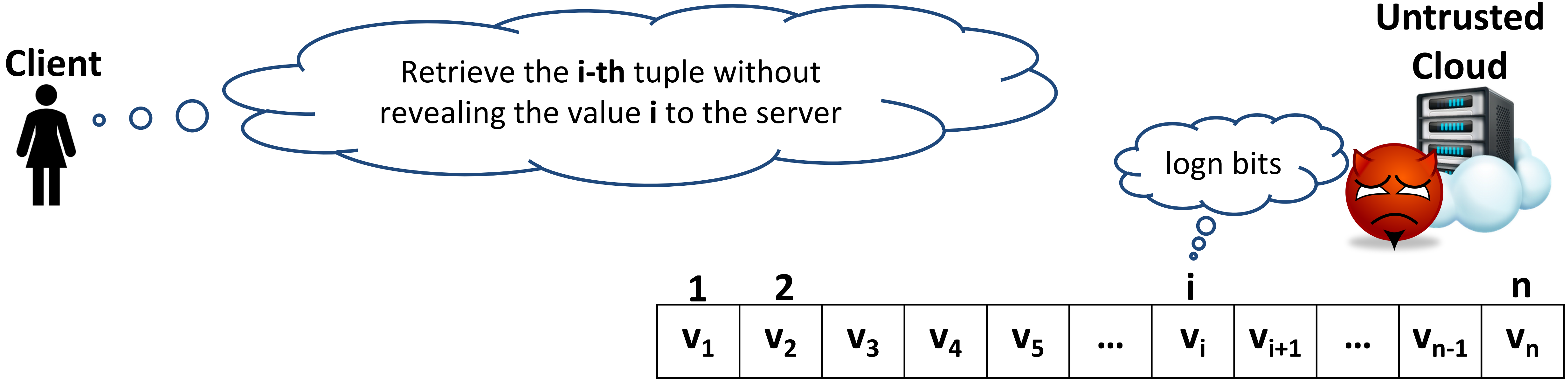
Contribution

- **SEAL: S**earchable **E**ncryption with **A**djustable **L**eakages
  - ADJable-ORAM-α (hides search and overlapping leakages)
  - ADJable-Padding-x (hides volume leakage)
- Attacks for point, range, join and group-by queries
  - First attack sketch for state-of-the-art range schemes
- New constructions for point, range, join, group-by queries
  - Using SEAL as black-box
- New customized Range Scheme, robust against attacks
- Experimental adjustment of search/overlapping/volume leakages

Focus of this talk

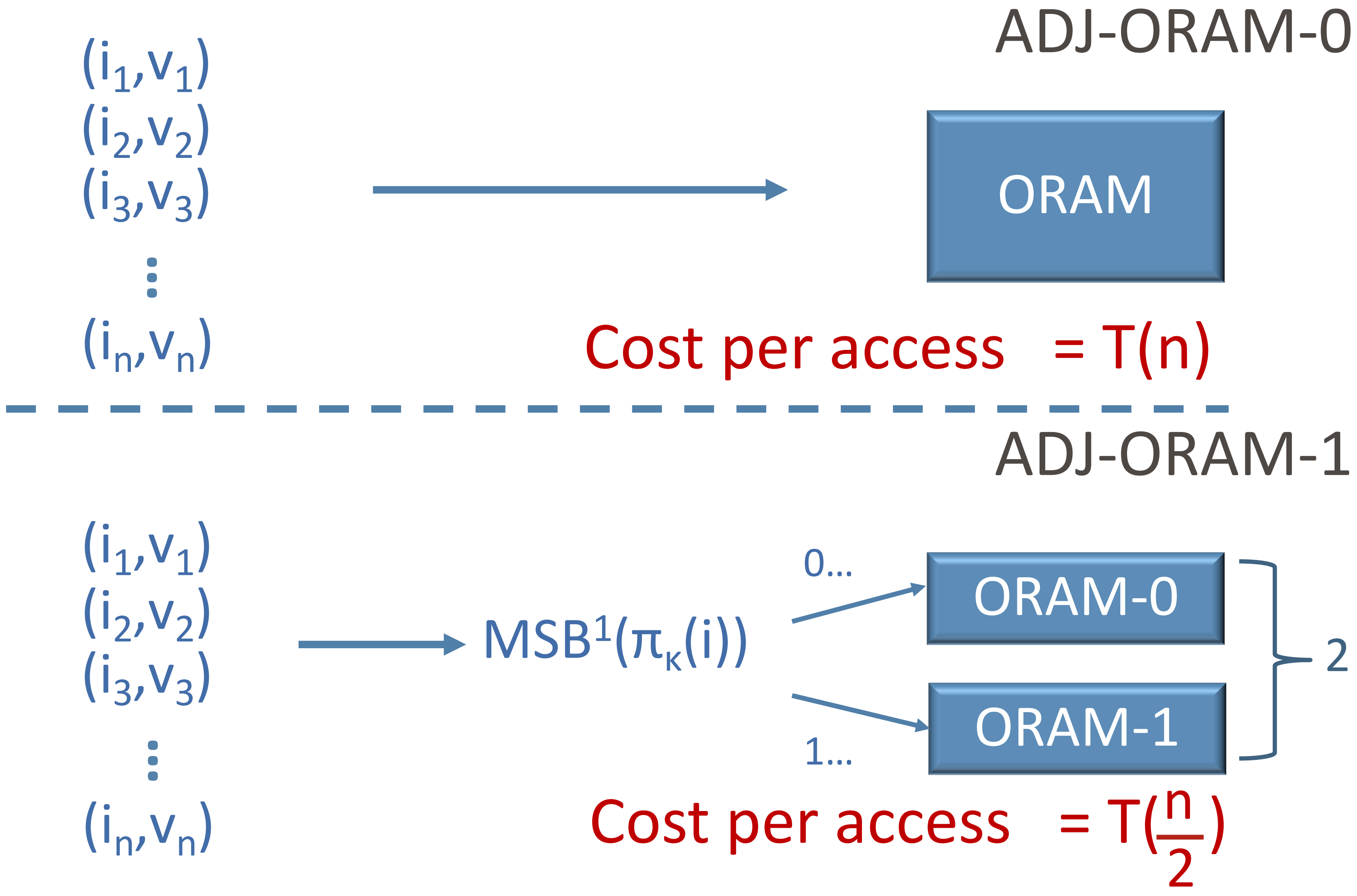
- **SEAL: Searchable Encryption with Adjustable Leakages**
  - **ADJable-ORAM-α (hides search and overlapping leakages)**
  - **ADJable-Padding-x (hides volume leakage)**
- Attacks for point, range, join and group-by queries
  - First attack sketch for state-of-the-art range schemes
- New constructions for point, range, join, group-by queries
  - Using SEAL as black-box
- New customized Range Scheme, robust against attacks
- **Experimental adjustment of search/overlapping/volume leakages**

Adjustable-ORAM-α (ADJ-ORAM-α)

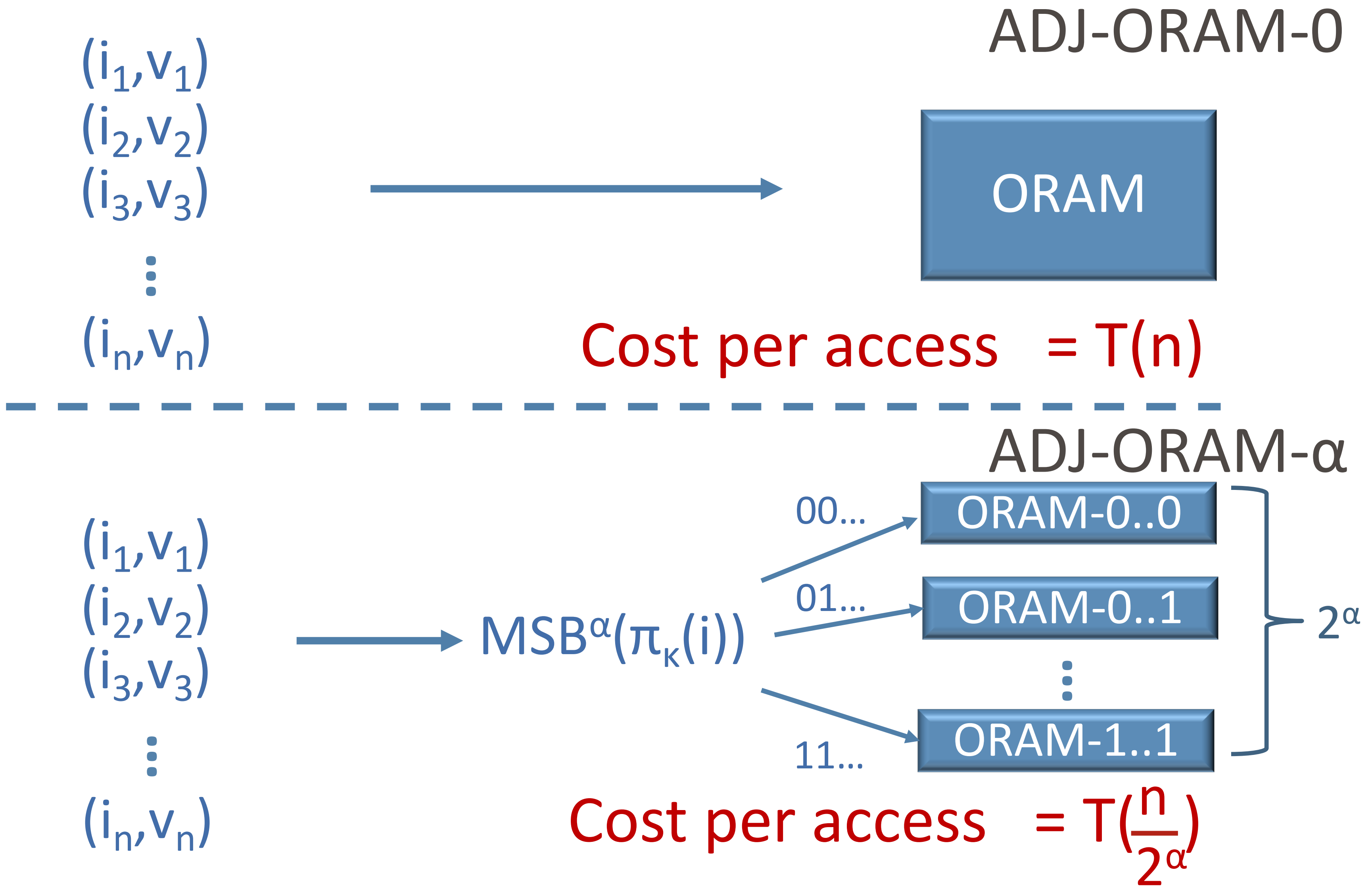


ADJ-ORAM-α: Leak α bits of the accessed memory locations!

ADJ-ORAM-α


ADJ-ORAM-0
(i_1,v_1)
(i_2,v_2)
(i_3,v_3)
$\vdots$
(i_n,v_n)
ORAM
Cost per access $= T(n)$
ADJ-ORAM-1
(i_1,v_1)
(i_2,v_2)
(i_3,v_3)
$\vdots$
(i_n,v_n)
$MSB^1(\pi_\kappa(i))$
0…
1…
ORAM-0
ORAM-1
2
Cost per access $= T(\frac{n}{2})$

ADJ-ORAM-α


ADJ-ORAM-0
(i_1,v_1)
(i_2,v_2)
(i_3,v_3)
⋮
(i_n,v_n)
ORAM
Cost per access = T(n)
ADJ-ORAM-α
(i_1,v_1)
(i_2,v_2)
(i_3,v_3)
⋮
(i_n,v_n)
$MSB^{\alpha}(\pi_{\kappa}(i))$
00…
01…
11…
ORAM-0..0
ORAM-0..1
⋮
ORAM-1..1
2^{α}
Cost per access = $T(\frac{n}{2^{\alpha}})$

Adjustable-Padding-x

- **Observation 1:** In a dataset of size N a query result can have up to N different sizes

- **Observation 2:** We can perform worst-case padding to eliminate the volume pattern leakage (1 unique size)

- Adjustable Padding: Pad all the query results to the closest power of **x**.
  - The server can observe up to $\log_x N + 1$ different sizes
  - Volume Pattern leakage $\log\log_x N + 1$ bits

- At the end, we pad the dataset to have x*N entries to avoid leaking extra information

SEAL(α,x)

- Uses ADJ-ORAM-α, ADJ-Padding-x and an oblivious dictionary as black-boxes

- Parameter α is defined in the range [0,log N]
  - α=0 all the search/overlapping pattern bits are protected
  - α=logN all the search/overlapping pattern bits are leaked

- For larger x values less volume pattern bits are leaked
  - x=N no volume pattern bits are leaked

- SEAL(α,x) can be used as a building block for point/range/join/group-by queries providing a security/efficiency trade-off

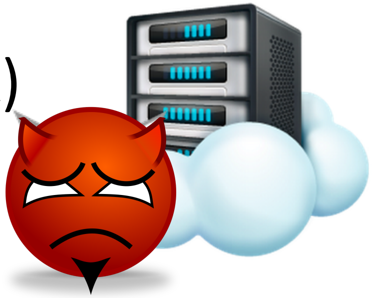
Outline

- SEAL: Searchable Encryption with Adjustable Leakages
  - ADJable-ORAM-α (hides search and overlapping pattern leakages)
  - ADJable-Padding-x (hides volume pattern leakage)
- Attacks for point, range, join and group-by queries
  - First attack-sketch for the state-of-the-art range schemes
- New constructions for point, range, join, group-by queries
  - Using SEAL as black-box
- New more efficient customized Range Scheme robust against attacks
- **Experimentally adjusting these leakages**

Threat Model and Attacks

Client

Untrusted Cloud



Attacker's Goals:

(i) Decrypt the client's encrypted queries (Query Recovery attack)

(ii) Decrypt the encrypted database (Database Recovery attack)

Attacker's Power:

→Has access to the server observing all the possible encrypted queries

→Has plaintext access to the input dataset

Query Recovery-Success Rate (QR_{SR})=
Correctly Decrypted Queries /|Q|

Database Recovery-Success Rate (DR_{SR})=
Correctly Decrypted Tuples / N

Attack Configuration

- Modified Frequency Analysis Attack proposed by Naveed et al. [CCS2016]

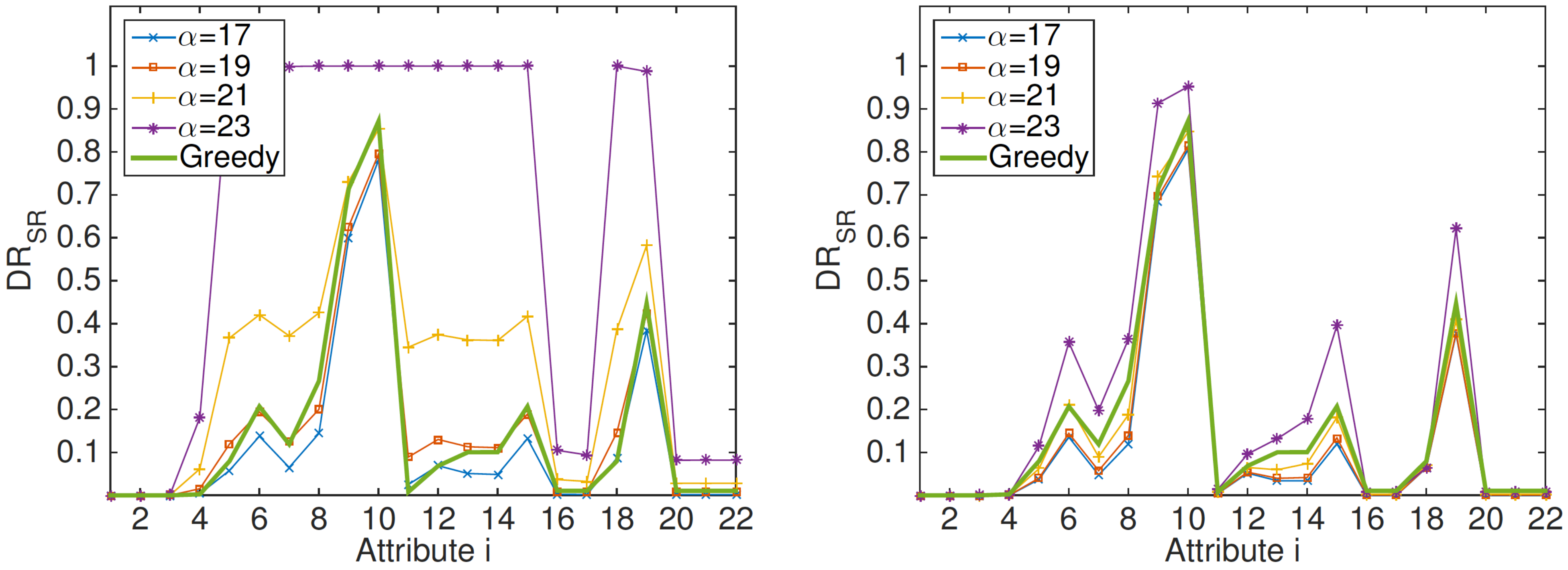
- 1 real dataset with 6,123,276 records of reported crime incidents
  - 22 attributes with different distributions:
    - ID, Case Number, Date, Block, ICR, Primary Type, Description, Location Description, Arrest, Domestic, Beat, District, Ward,
    - Community Area, FBI Code, X Coordinate, Y Coordinate, Year,
    - Updated On, Latitude, Longitude, Location.

- TPC-H Benchmark
  - 8 tables (61 different attributes)
    - PART, SUPPLIER,PARTSUPP, CUSTOMER, NATION, LINEITEM, REGION,ORDERS

Database Recovery Attack for Point Queries (Crime Dataset)

Parameter **α** controls **the search/overlapping pattern** leakage (**α**=[**0**…**logN**])

Parameter **x** controls **the volume pattern** leakage (**x**=[**No padding**, 2, … **N**])

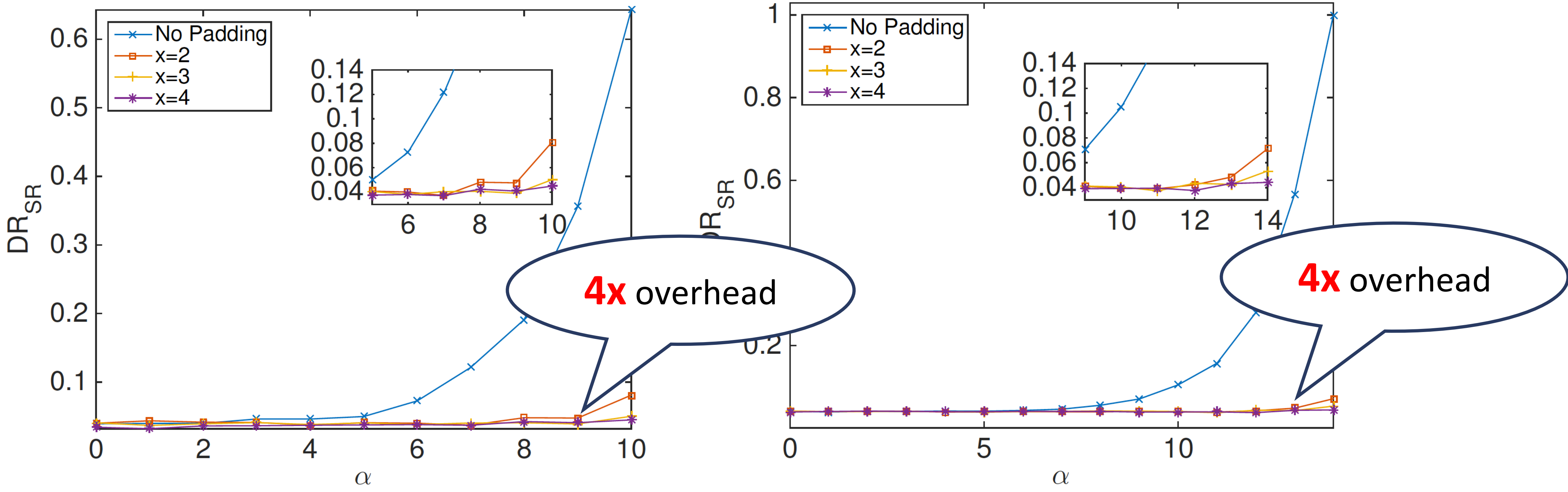


(a) $x = \perp$ (b) $x = 2$

Database Recovery Attack for Join Queries

Parameter **α** controls **the search/overlapping pattern** leakage (**α**=[**0**…**logN**])

Parameter **x** controls **the volume pattern** leakage (**x**=[**No padding**, 2, … **N**])



(a) SUPPLIER ⋈NATION (b) CUSTOMER ⋈NATION

Adjusting Parameters “α” and “x” in Practice

Finding appropriate parameter values is ***data-dependent***:

- Size of the database
- Number of distinct values
- Distribution of the searchable attribute

Approach: Before outsourcing the database, for a given attribute, use existing/our all-powerful attacks and try different values of “a” and “x”

General Guidelines:

- **Point/Join/Group-by queries:** α=logN-3 and x=4 (~**32x** overhead)
- **Range Queries**: x=8 (~**12x** overhead)

Thank you!! Questions?