# Reconstructing Encrypted Data Using Range Query Leakage

Marie-Sarah Lacharité, Brice Minaud, Kenny Paterson

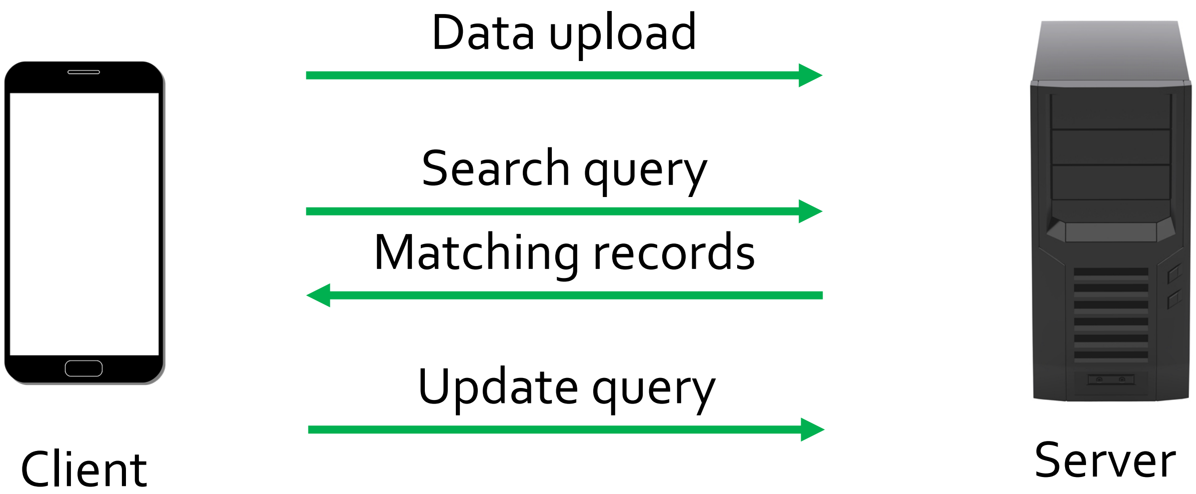
*ePrint 2017/701, to appear S&P 2018.*

Information Security Group

ROYAL HOLLOWAY UNIVERSITY OF LONDON

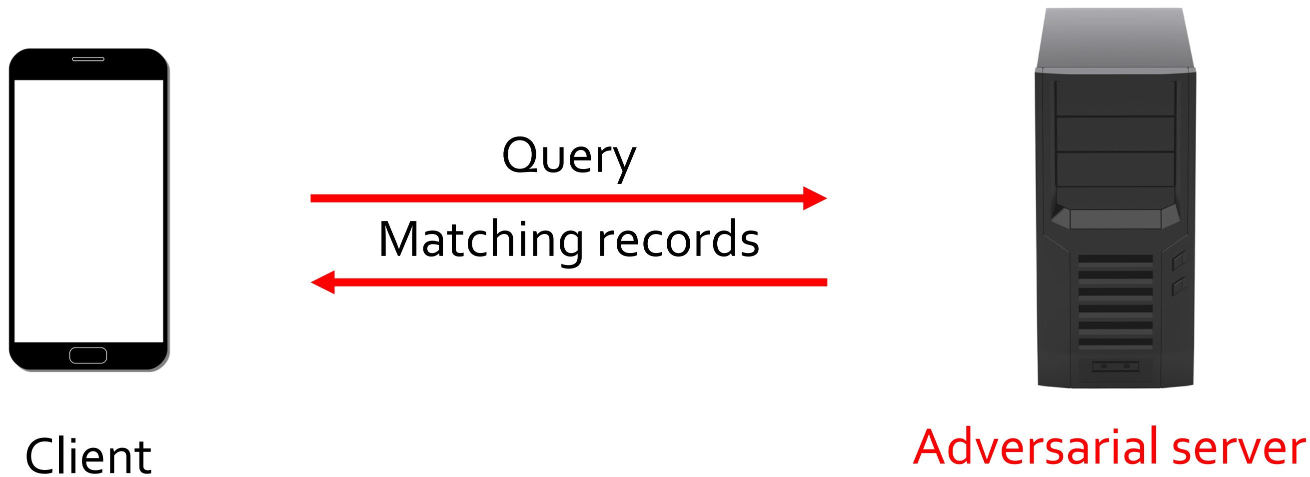
Workshop IoT+Cloud, Bochum, 7 Nov 2017.

# Outsourcing Data to the Cloud




- For **encrypted database management systems**:
    - Data = collection of records in a database (e.g. health records).
    - Query examples =
      - Find records with a given value (e.g. patients aged 57).
      - Find records within a given range (e.g. patients aged 55 to 65).
      - ...

# Security of Data Outsourcing Solutions




- **Adversaries:**
  - **Snapshot** adversary = breaks into server, gets snapshot of memory.
  - **Persistent** adversary = corrupts the server for a period of time. Sees all communication transcripts. Can be server itself.
- **Security goal = privacy:**

  Adversary learns as little as possible about the client's data and queries.

# State of the Art

- **No perfect solution.**

  Every solution is a trade-off between **functionality** and **security**.

- **Huge amount of literature.**

  [AKSX04], [BCLO09], [PKV+14] , [BLR+15], [NKW15], [K15], [CLWW16], [KKNO16] , [RACY16], [LW16] ...

- **A few "complete" solutions**:

  Mylar (for web apps)

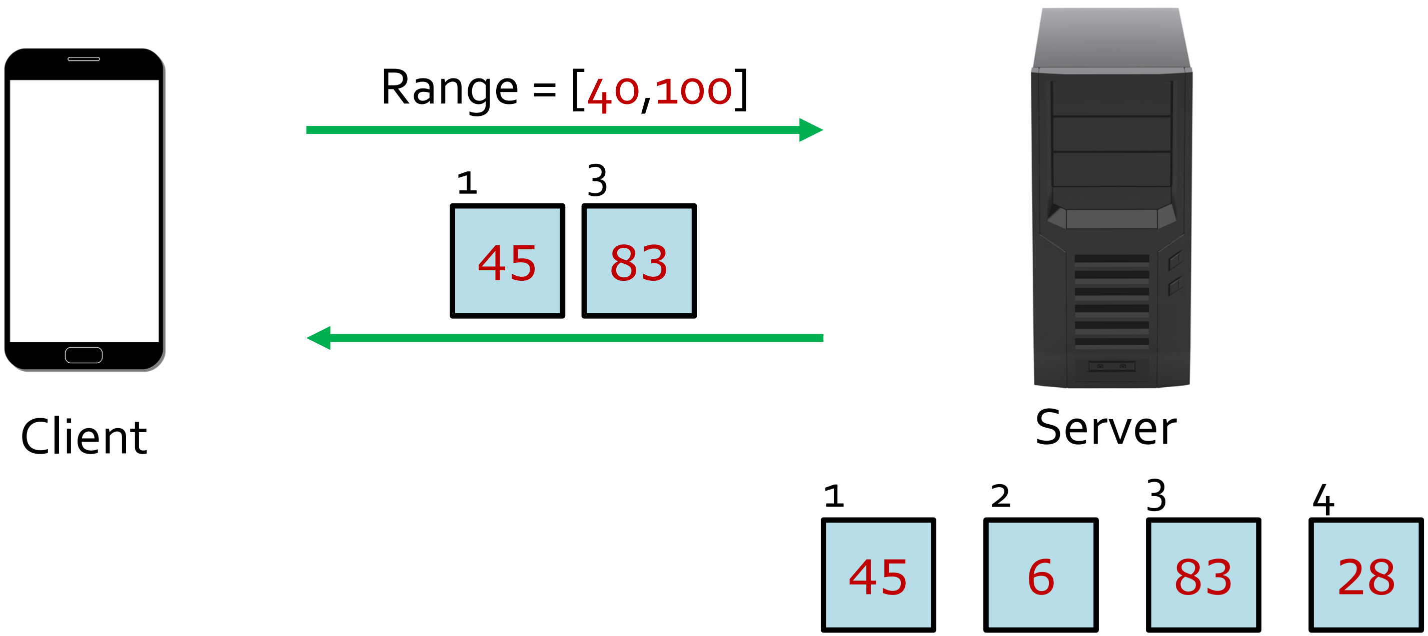
  CryptDB (handles most of SQL)

  ⚠ *Controversial!*

  ➔ Cipherbase (Microsoft), Encrypted BigQuery (Google), ...

- **Very active area of research.**

# Setting for this Talk: Schemes Supporting Range Queries




- All known schemes leak set of matching records = **Access Pattern**.

  OPE, ORE schemes, POPE, [HK16], Blind seer, [Lu12], [FJKNRS15],…

- Some schemes also leak # records below queried range endpoints = **rank**.

  FH-OPE, Lewi-Wu, Arx, Cipherbase, EncKV,…

# Exploiting leakage

- Most schemes prove that nothing more leaks than their leakage model allows.
- For example, leakage = access pattern, or access pattern + rank.
- *What can we really learn from this leakage?*

- **Our goal**: full reconstruction = recover the exact value for every record.

- **[KKNO16]**: $O(N^2 \log N)$ queries suffice for full reconstruction using only access pattern leakage!

  - where N is the number of possible values (e.g. 125 for age in years).

# Assumptions for our Analysis

1. Data is **dense:** all values appear in at least one record.

2. Queries are **uniformly distributed.**

Our algorithms don't actually care though – the assumption is for computing data upper bounds.

# Our Main Results

- **Full reconstruction** with $O(N \cdot \log N)$ queries from access pattern
  - in fact, $N \cdot (3 + \log N)$.

- **Approximate reconstruction** with relative accuracy ε with $O(N \cdot (\log 1/\varepsilon))$ queries.

- **Approximate reconstruction** using an *auxiliary distribution* and rank leakage.
  - more efficient in practice, evaluation via simulation.

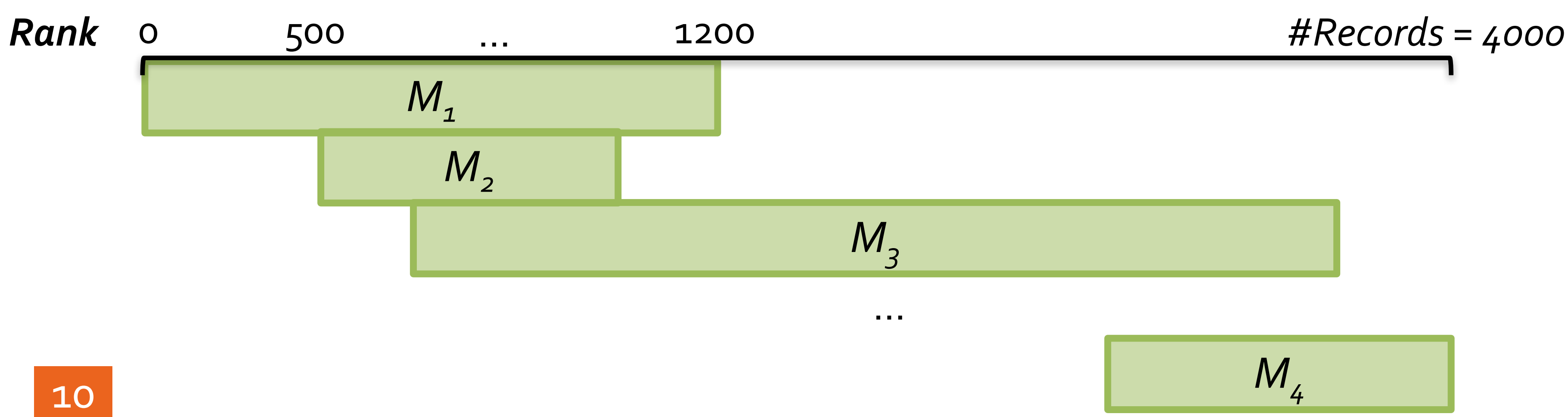
# Attack 1: Full Reconstruction

# Full Reconstruction with Rank Leakage

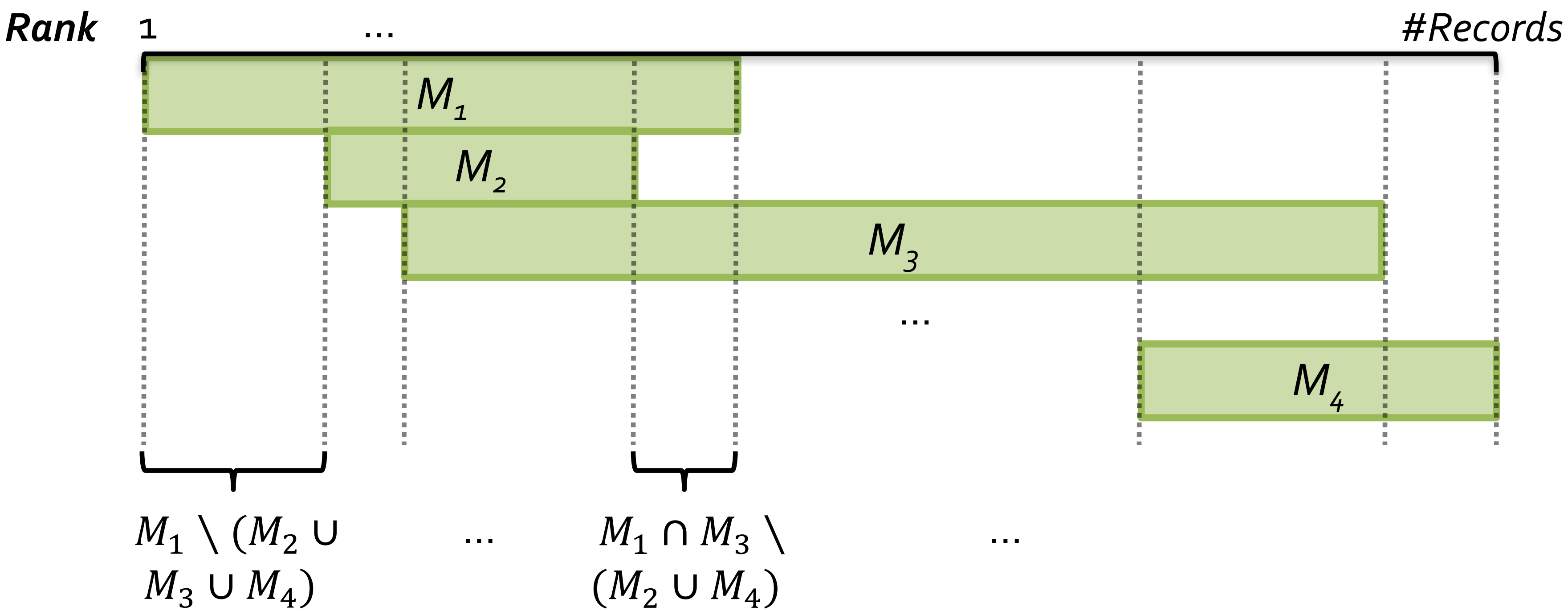
- Adversary is observing query leakage…

(Reordered for convenience)

| Hidden | Leaked | | |
|---|---|---|---|
| Query [x,y] | a = rank(x-1) | b = rank(y) | Matching IDs |
| [1,18] | 0 | 1200 | $M_1$ |
| [2,10] | 500 | 800 | $M_2$ |
| [7,98] | 600 | 3000 | $M_3$ |
| [55,125] | 2000 | 4000 | $M_4$ |



# Full Reconstruction with Rank Leakage




- Partition records into smallest possible sets using access pattern leakage.
- If this partitions records into *N* sets, **win**! Just match minimal sets with values.

# Full Reconstruction with Rank Leakage

- Expected number of queries **sufficient** for full reconstruction is at most:

$$N \cdot (2 + \log N) \quad \text{for } N \geq 27.$$

  Essentially a coupon collector's problem.

- Expected number of **necessary** queries is at least:

$$1/2 \cdot N \cdot \log N - O(N)$$

  for *any* algorithm.

- This algorithm is "data-optimal", i.e. it fails iff full reconstruction is impossible for *any* algorithm given the input data.

# Full Reconstruction **without** Rank Leakage

- **Very generic setting:** use only access pattern leakage.
- **Partition** (as before), then **sort**.
- Expected number of **sufficient** queries is at most:

$$N \cdot (3 + \log N) \qquad \text{for } N \geq 26$$

  - i.e. sorting step is very cheap in terms of data.

- Expected number of **necessary** queries is at least:

$$1/2 \cdot N \cdot \log N - O(N)$$

  for *any* algorithm.

- Still data-optimal!

# Attack 2: Reconstruction with Auxiliary Data

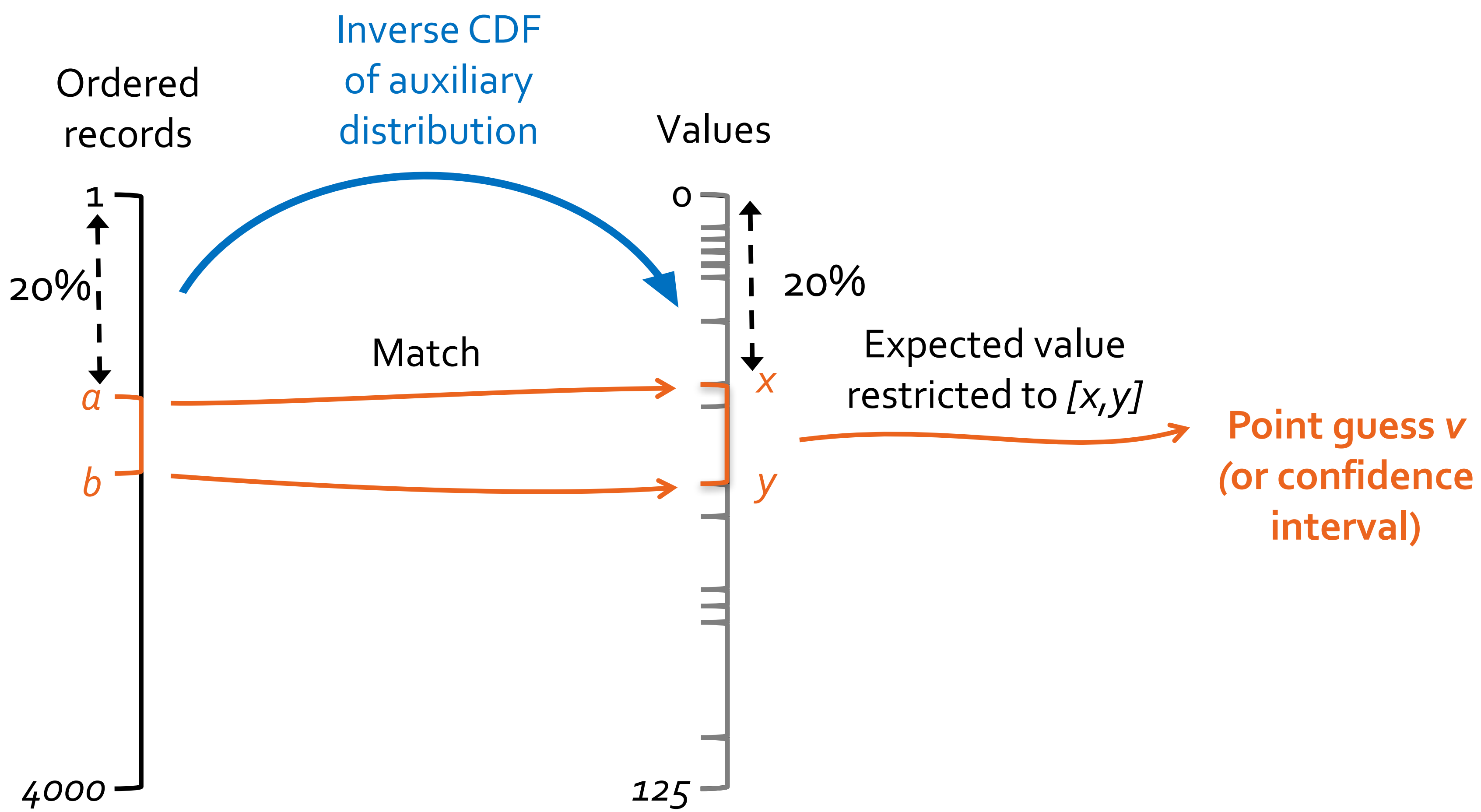
# Reconstruction with Auxiliary Data and Rank Leakage

- As before, queries have ranges chosen uniformly at random.
- Assume access pattern and rank are leaked.
- We now also assume that an **approximation to the distribution on values** is known.

  "Auxiliary distribution".

  From aggregate data, or from another reference source.
- We show experimentally that, under these assumptions, **far fewer queries** are needed.

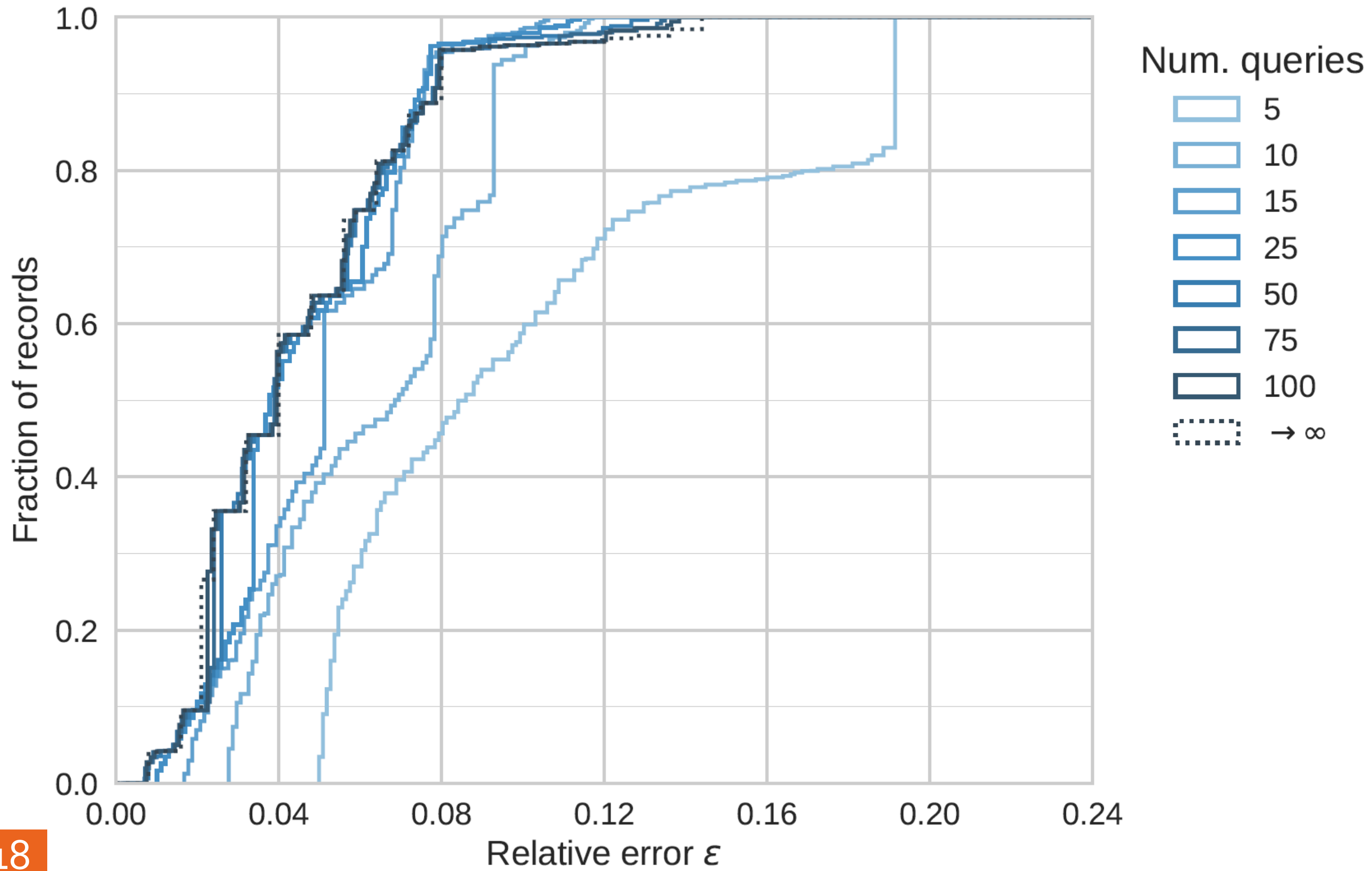
# Auxiliary Data Attack: Estimating Step



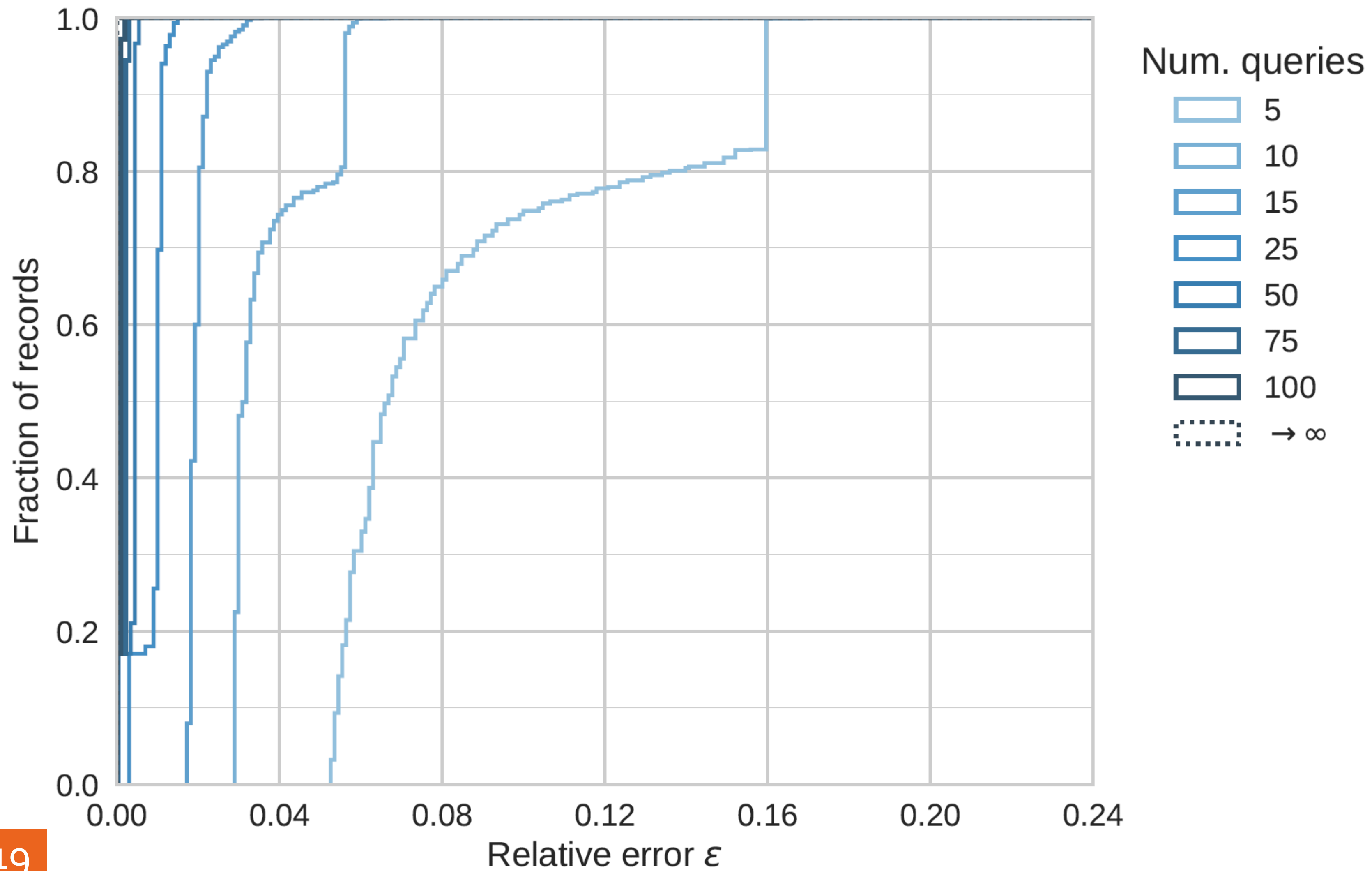
# Auxiliary Data Attack: Experimental Evaluation

- Ages, $N = 125$ (0 to 124).
- Health records from US hospitals (NIS HCUP 2009).
- **Target:** age of individual hospitals' records.
- **Auxiliary data:** aggregate of 200 hospitals' records.
- **Measure of success:** proportion of records with value guessed within $\varepsilon$.

# Auxiliary Data Attack: Results for Typical Target Hospital



# Auxiliary Data Attack: Results with Perfect Auxiliary Distribution



# Summary and Conclusions

# Summary of the attacks

- Our results : full reconstruction in ≈N log N queries with only access pattern!

  Efficient, data-optimal algorithms + matching lower bound.

| Attack | Req'd leakage | Other req'ts | Suff. # queries |
|---|---|---|---|
| KKNO16 | AP | Density | $O(N^2 \log N)$ |
| Full | AP + rank | Density | $N \cdot (\log N + 2)$ |
| | AP | Density | $N \cdot (\log N + 3)$ |
| ε-approximate | AP | Density | $5/4\, N \cdot (\log 1/\varepsilon) + O(N)$ |
| Auxiliary | AP + rank | Auxiliary dist. | Experimental |

- For N = 125, about 800 queries suffice for full reconstruction!
- If an auxiliary distribution + rank leakage is available, after only 25 queries, 55% of records can be reconstructed to within 5 years!

# Conclusions

- Many clever schemes have been designed, enabling range queries on encrypted data.

  OPE, ORE schemes, POPE, [HK16], Blind seer, [Lu12], [FJKNRS15], FH-OPE, Lewi-Wu, Arx, Cipherbase, EncKV,...

- Second-generation schemes **defeat the snapshot adversary** (with caveats).

- But as our attacks show, **no known scheme offers meaningful privacy** vs. a **persistent adversary** (including server itself).

  In realistic settings, N log(N) queries suffice; even less if auxiliary distribution + rank leakage is known.

- More research needed!