Efficient Searchable Encryption Through Compression

Ioannis Demertzis
University of Maryland
yannis@umd.edu

Rajdeep Talapatra
University of Maryland
rajdeep@umd.edu

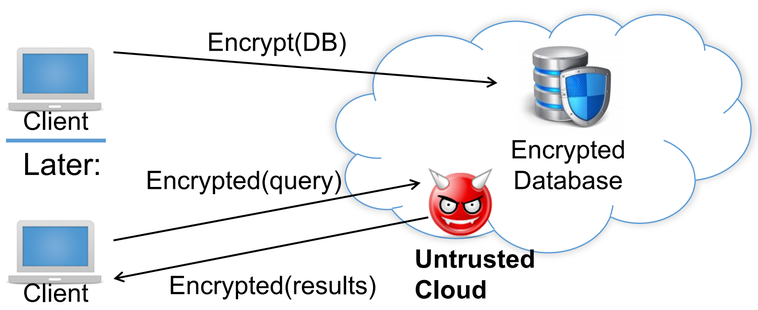
Charalampos Papamanthou
University of Maryland
cpap@umd.edu

Motivation

Cloud Computing:
- Offers: **near infinite scalability**, **easy & ubiquitous** access …
- Serious **security** and **privacy concerns** when outsourcing and querying on private company or personal data

General Data Protection Regulation (GDPR) Era:
- GDPR is more flexible when data are encrypted
- Standard Encryption **limits** **usability**, **efficiency**, **functionality**

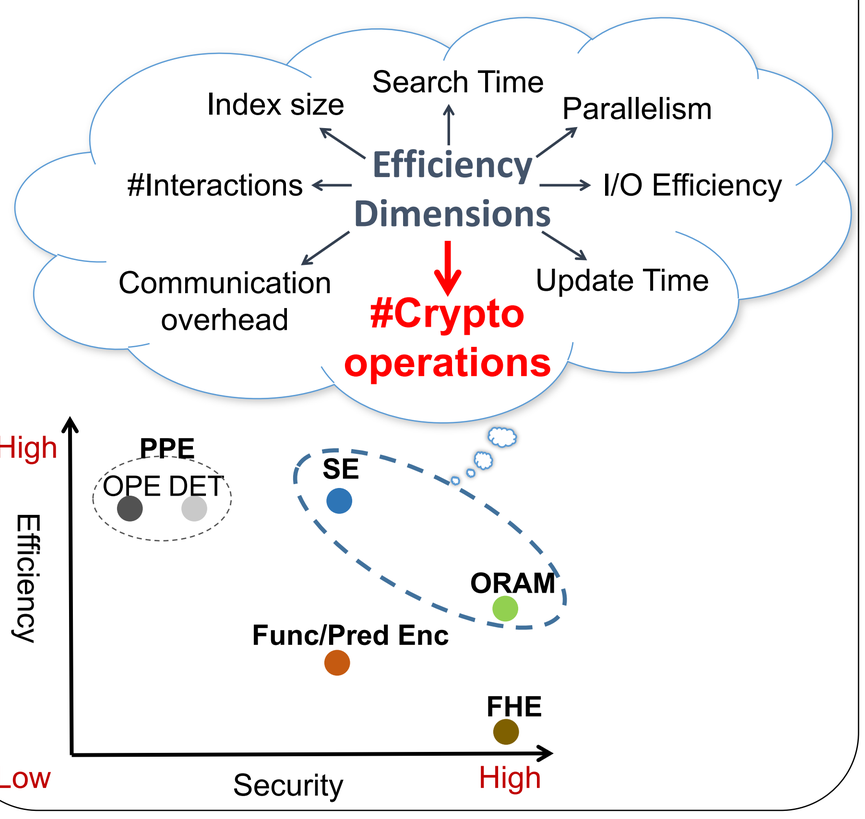
Solution: Searchable Encryption (**SE**)/Oblivious SE (**OSE**)



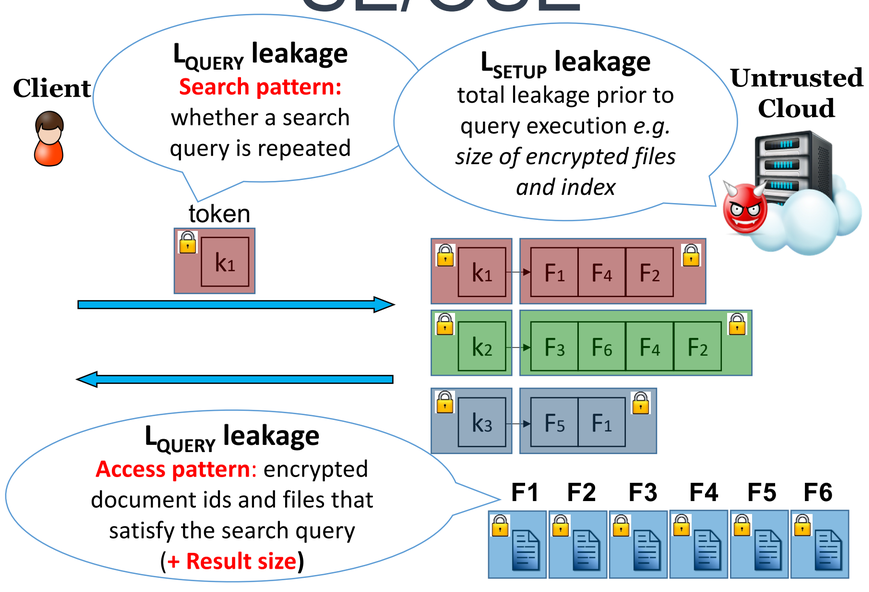
Our contribution:

- New efficiency dimension for SE: **#cryptographic operations** to retrieve query results
- Use any set of lossless **compression** algorithms and any **SE/OSE** schemes as black-boxes to improve their search efficiency without affecting their security
- Keyword search time savings: up to **188x** (Enron dataset)
- Database search: up to **62x** (location description attribute), **170-203x** (binary attributes).
- Our OSE approach reduces the index search time to access **1M** tuples from **21 hours** to **20 minutes**

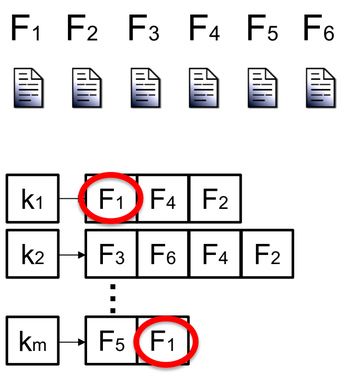
Related Work



SE/OSE

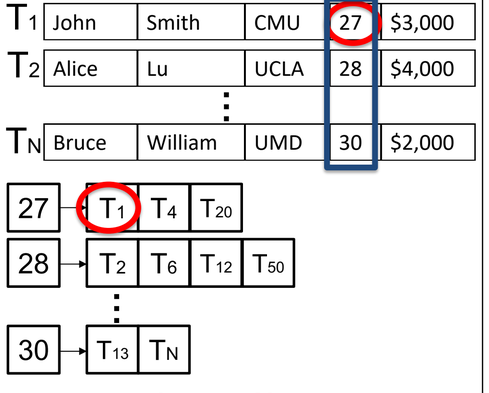


Keyword Search



L_{SETUP}: N,m
SE-> L_{QUERY}: Access + Search
OSE-> L_{QUERY} : Result size

Database Search

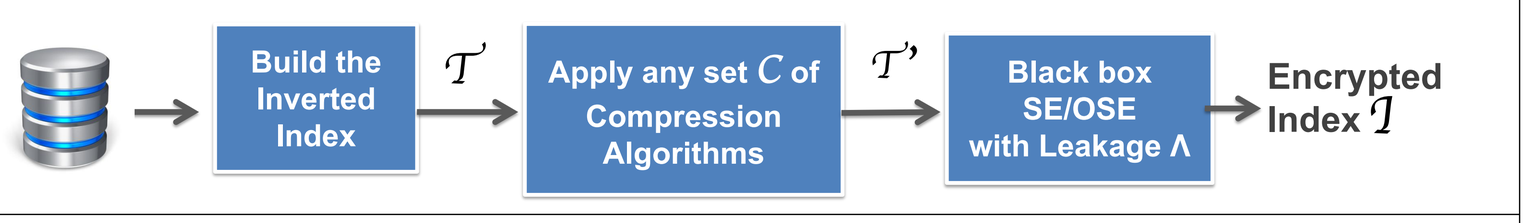


L_{SETUP}: N
SE-> L_{QUERY}: Result size + Search
OSE-> L_{QUERY} : Result size

Our Approach

General Idea of microSE/microOSE

microSE/microOSE with leakage Λ', where Λ'<=Λ



Combing Encryption with Compression

Compress(Encrypt(x))
e.g. Compress(E21B74E9E4AC)
- The compression does not affect security
- Compression becomes useless

Encrypt (Compress(x))
e.g. Encrypt (Compress(aabbbbbb))
- The input distribution leaks information
- Very good compression ratio

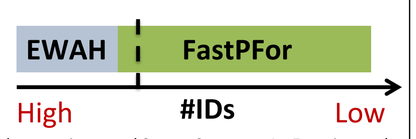
Utilize only the leakage of SE/OSE
- The compression does not affect SE/OSE security
- Good compression ratio

Construction's Highlights

Prior SE schemes: each identifier is a random string of κ bits → **Compression of a sequence of random strings is useless**

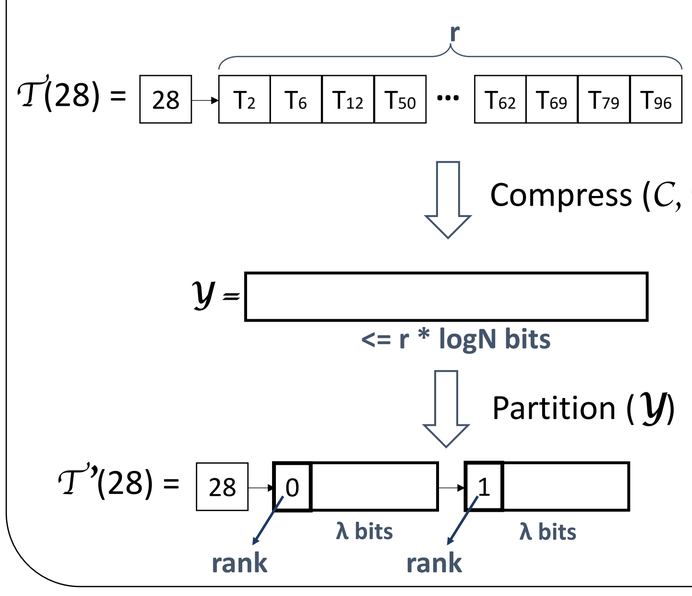
Solution: Assign an id chosen uniformly at random from the range [0, N − 1] (N = #tuples) → **Achieves good compression + The compression does not affect SE's security**

Problem: Compress the sequence of identifiers that are uniformly distributed in the range [0, N-1]
- Chosen compression algorithms based on [WLP+17]:
  - EWAH
  - FastPFor



[WLP+17]:"Jianguo Wang, Chunbin Lin, Yannis Papakonstantinou, and Steven Swanson. An Experimental Study of Bitmap Compression vs. Inverted List Compression. SIGMOD 2017 "

microSE



microSE/microOSE

microSE: Assume that $|\mathcal{T}(28)| = |\mathcal{T}(32)|$ then $|\mathcal{T}'(28)| \gtrless |\mathcal{T}'(30)|$

By leaking the **search pattern** the adversary knows that $\mathcal{T}'(28)$ and $\mathcal{T}'(30)$ are different queries

microOSE: $|\mathcal{T}'(28)|$ **has to be equal to** $|\mathcal{T}'(30)|$ since the **search pattern** is **not** leaked **!!!**

Solution 1
Compress*(C, $\mathcal{T}(28)$) pads to the worst-case

Solution 2
Store the overflowed lists in a local stash on the client side

Experiments

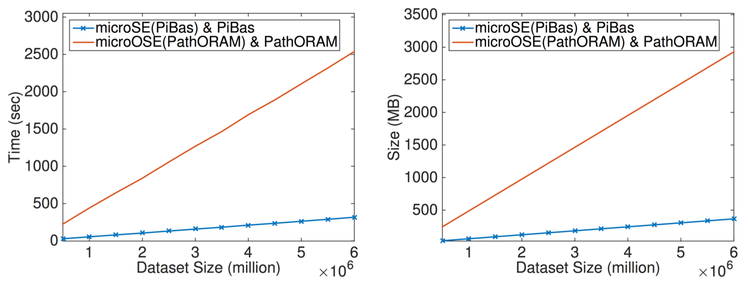
Setup
- Our SE Scheme (**microSE**) + used *PiBas* [CJS+14] as black-box SE → **microSE**(**PiBas**)
- Our OSE Scheme (**microOSE+Solution2***) +used PathORAM [SDS+12] as black-box → **microOSE(PathORAM)**
- Java implementation using JavaX.crypto and Bouncy Castle
- 64bit machine with Intel Xeon E5-2676v3 with 64GB RAM

*Local stash for solution 2 was always smaller than the stash of PathORAM

Database Search
- Chicago crime incidents dataset with 6,123,276 records
  - 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

Keyword Search
- Enron email dataset with 30,109 emails of 150 employees of the Enron corporation sent between 2000-2002

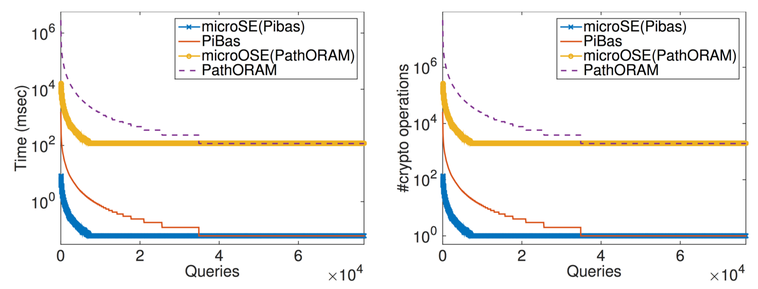
Setup Costs



(a) Construction time (b) Index size

Crime Dataset

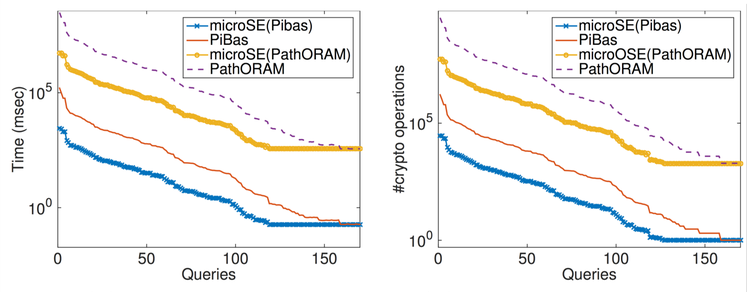
Search Time – Keyword Search



(a) Search Time (b) # PRF evaluations

- **microSE**(PiBas) & **microOSE**(PathORAM) are up to **188x** more efficient than PiBas and PathORAM

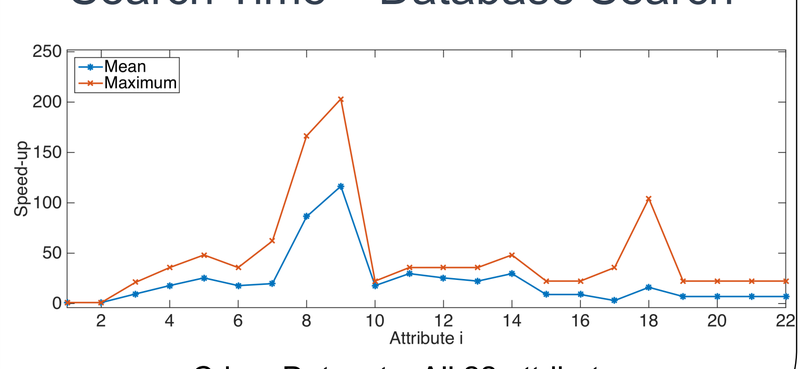
Search Time – Database Search



(a) Search Time (b) # PRF evaluations

- **microSE**(PiBas) is up to **62x** more efficient than PiBas
- **microOSE**(PathORAM) accesses 1M tuples in **20 minutes** vs **21 hours** of PathORAM

Search Time – Database Search



Crime Dataset – All 22 attributes