33RD USENIX SECURITY SYMPOSIUM

STEVENS INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

AISecLab

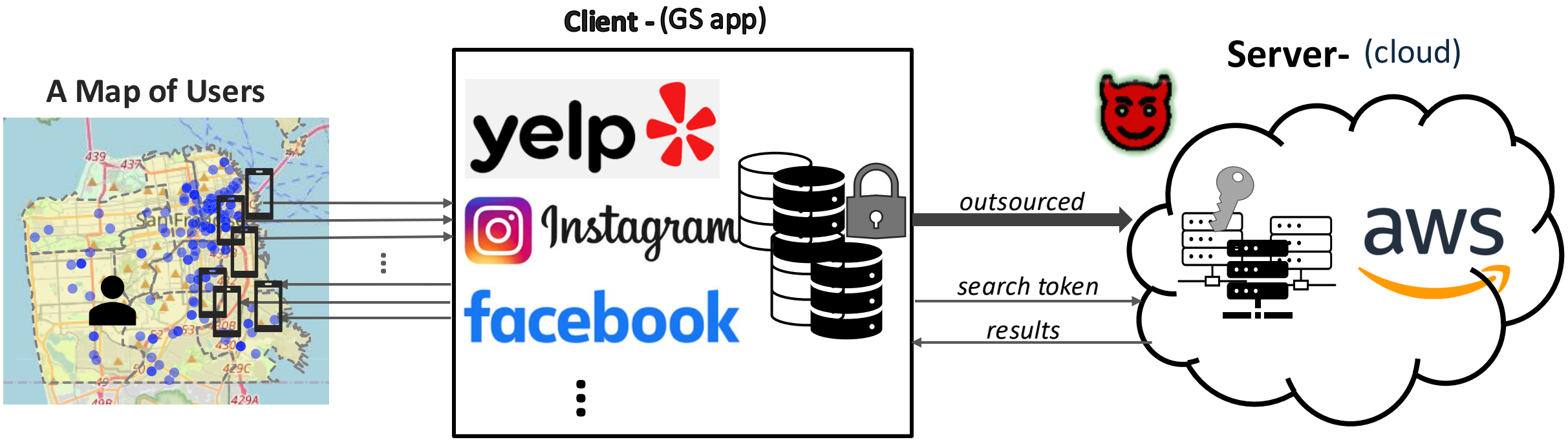
GridSE: Towards Practical Secure Geographic Search via Prefix Symmetric Searchable Encryption

Ruoyang Guo, Jiarui Li, Shucheng Yu

August 16, 2024

Geographic Search (GS) on sensitive data

- GS apps need to collect personal information
  - *users' real-time locations*

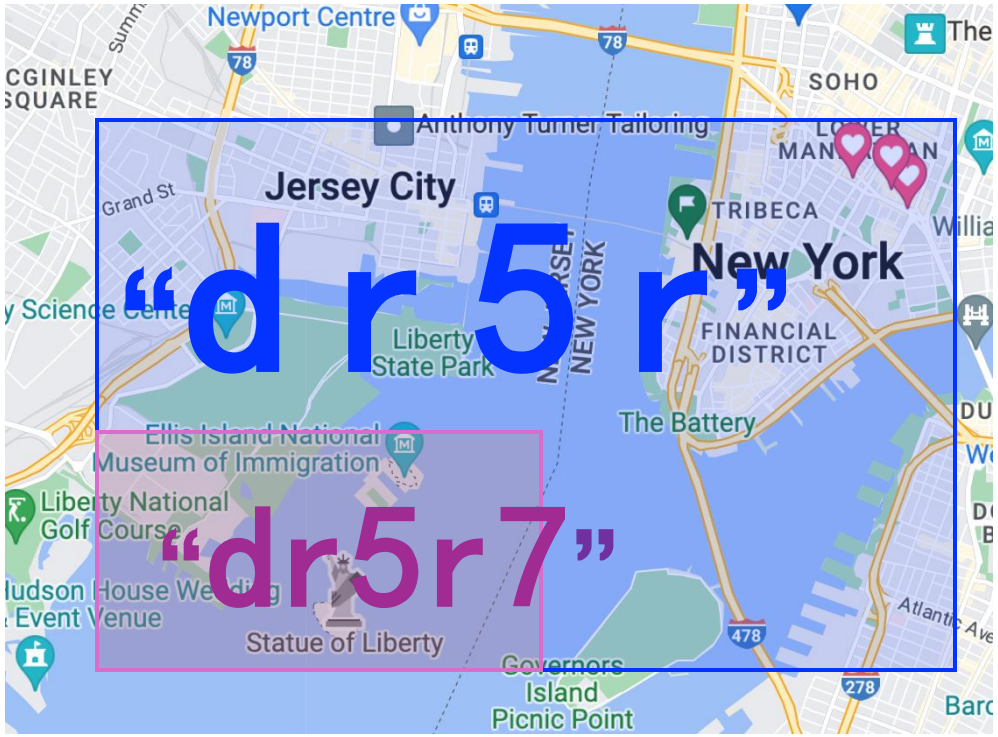


Compatibility with Discrete Global Grid Systems

- Existing GS Apps have been deeply rooted in DGGS
  - *Geohash, Google S2, Uber H3, etc.*

A tourist: *“ Best place to visit in New Jersey? ”*

Search Result: *“ Statue of Liberty”*



A Secure Geographic Search (SGS) app needs a Secure Prefix Search!

SGS Design Objectives

- Near **latency-free** instant response
- **Compatibility** with DGGS
  - Searchable Encryption for prefix (pSSE)
- **Dynamic Search**
  - GS database **updates**: add or delete entries
  - **Forward Privacy**: newly added data cannot be linked to previous queries
  - **Backward Privacy**: a query cannot be linked to previously deleted data

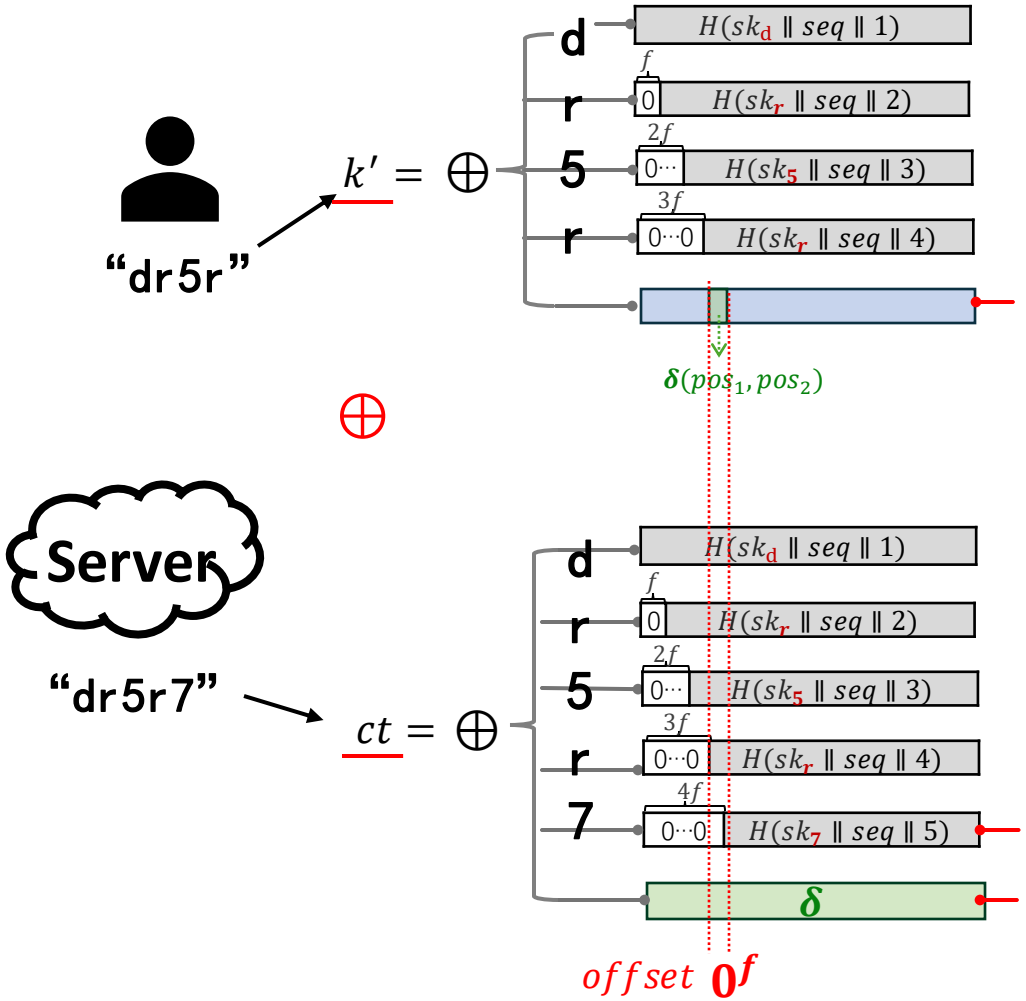
State of The Art

- **Dynamic SSE** only supports the whole keyword search
- **Secure Substring Search**
  - Tradeoff between update, efficiency, security and false-positive rate

Directly constructing SGS with existing primitives is difficult.

Our Design

1) SP^2E : To evaluate whether a keyword contains a given prefix



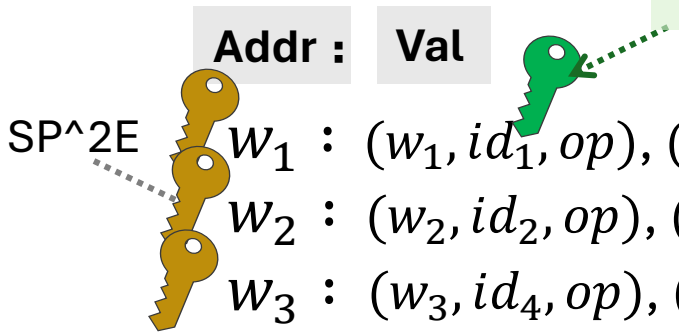
Our Design

2) Forward/Backward Privacy : To prevent updates from leaking information

- A triplet of (w, id, op)
  - w: keyword
  - id: files/entries/records in the database
  - op: deletion or addition
- A dictionary map, key-value pair

2 counters for encryption:

- **UpdtCnt** – *stores the update times occurred on files under index-key/addr w*
- ***Seq*** -- *the dictionary sequence of Addr*



Addr :	Val
w_1	$(w_1, id_1, op), (w_1, id_3, op), (w_1, id_{10}, op), (w_1, id_{11}, op)$ ...
w_2	$(w_2, id_2, op), (w_2, id_7, op), (w_2, id_8, op), (w_2, id_{50}, op)$ ...
w_3	$(w_3, id_4, op), (w_3, id_5, op), (w_3, id_6, op), (w_3, id_{32}, op)$ ...

Our Design

3) A generic framework transforming a *dynamic SSE* into a *dynamic prefix SSE*

- Keep the basic setting and processing flow of GridSE
  - *secret keys storage/arrangement*
- Follow the dictionary structure
  - key on prefix search
  - value on other generic dynamic SSE methods

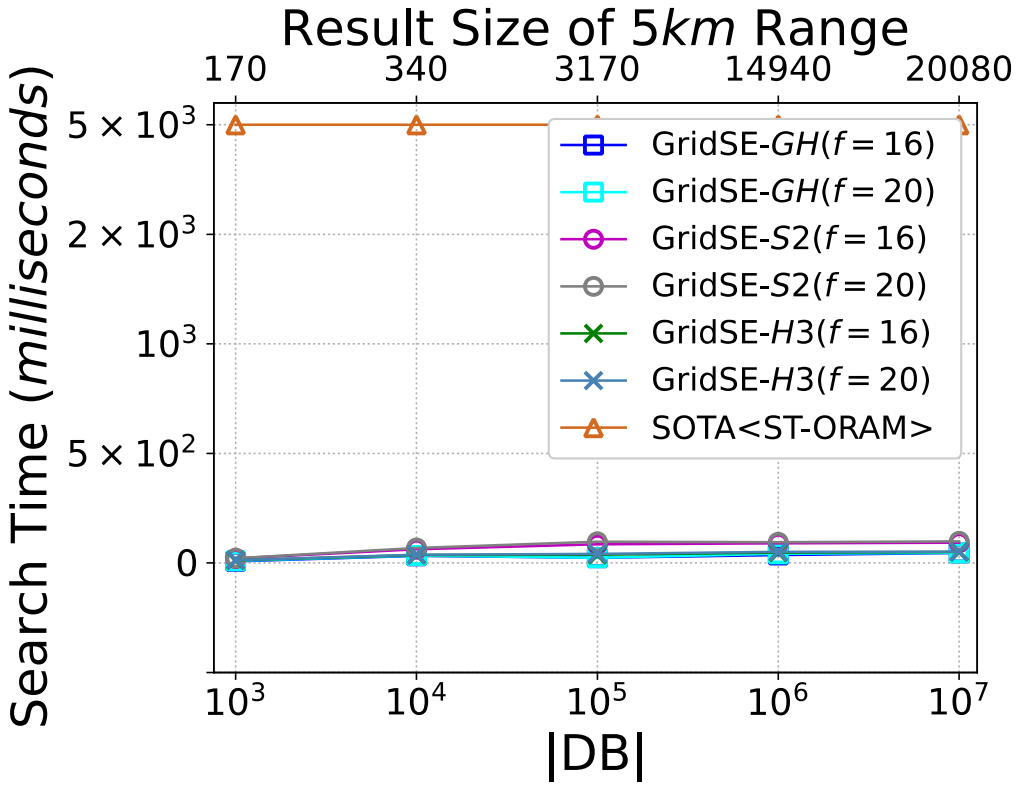
Evaluation

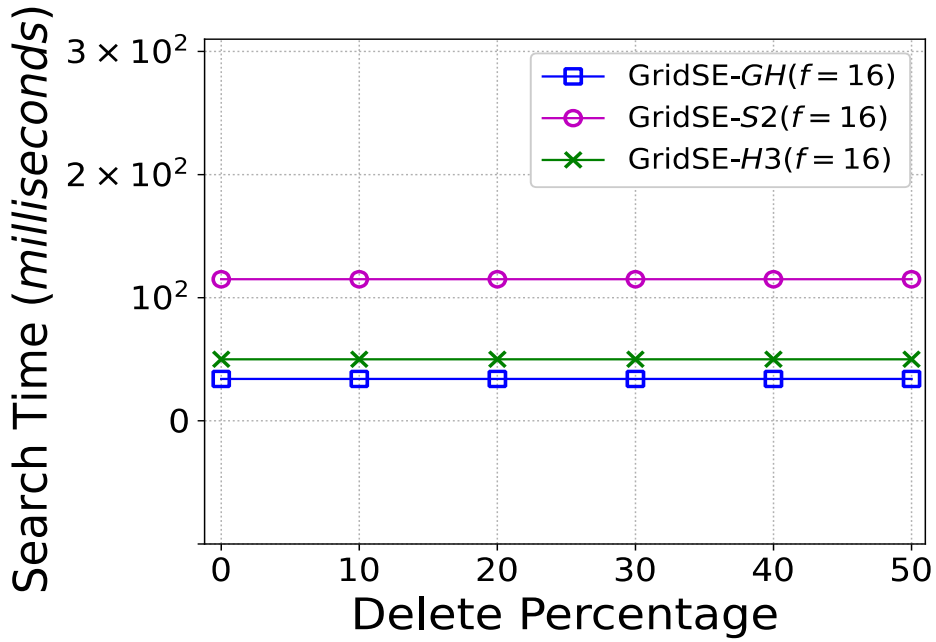
- **Dataset**
  - Gowalla location check-in dataset [1], *6,442,890 records from 196,591 users*
  - Obtain 63,369 distinct users within California
  - **Test** *various* database $|DB| = 10\text{^}5 - 10\text{^}7$

    query $|Q| = 10 - 10\text{^}5$, spanning from $100m^2 - 100\ km^2$
- **DGGS**
  - Geohash, Google S2, Uber H3
- **Deletions**
  - *Random 10% of the result matching the queried prefix*

Evaluation

- **Search time**
  - with/without deletions

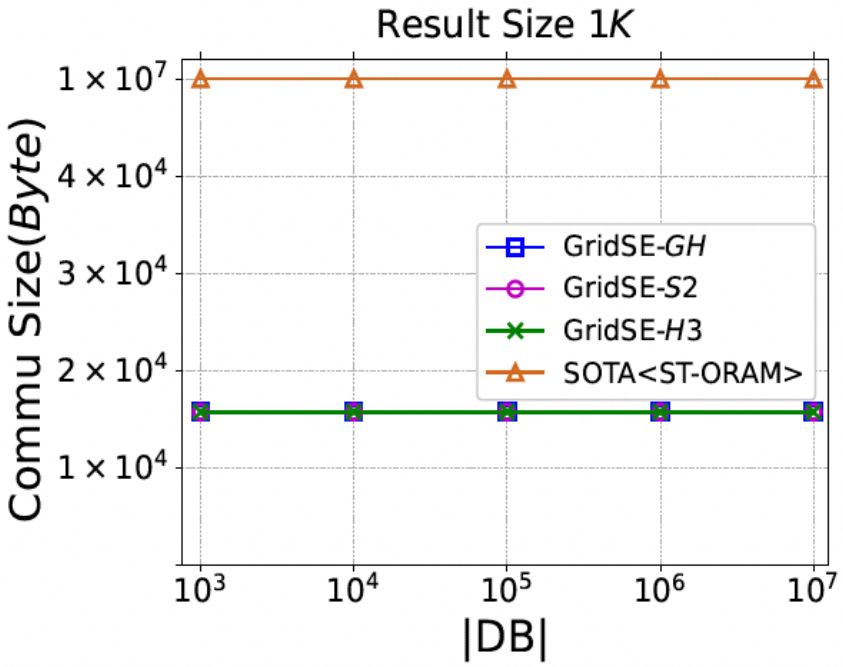


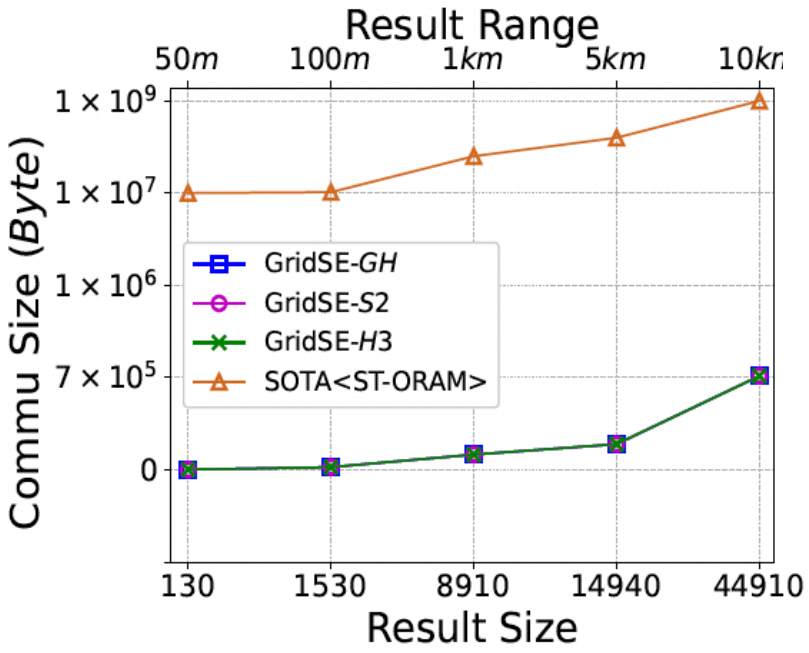


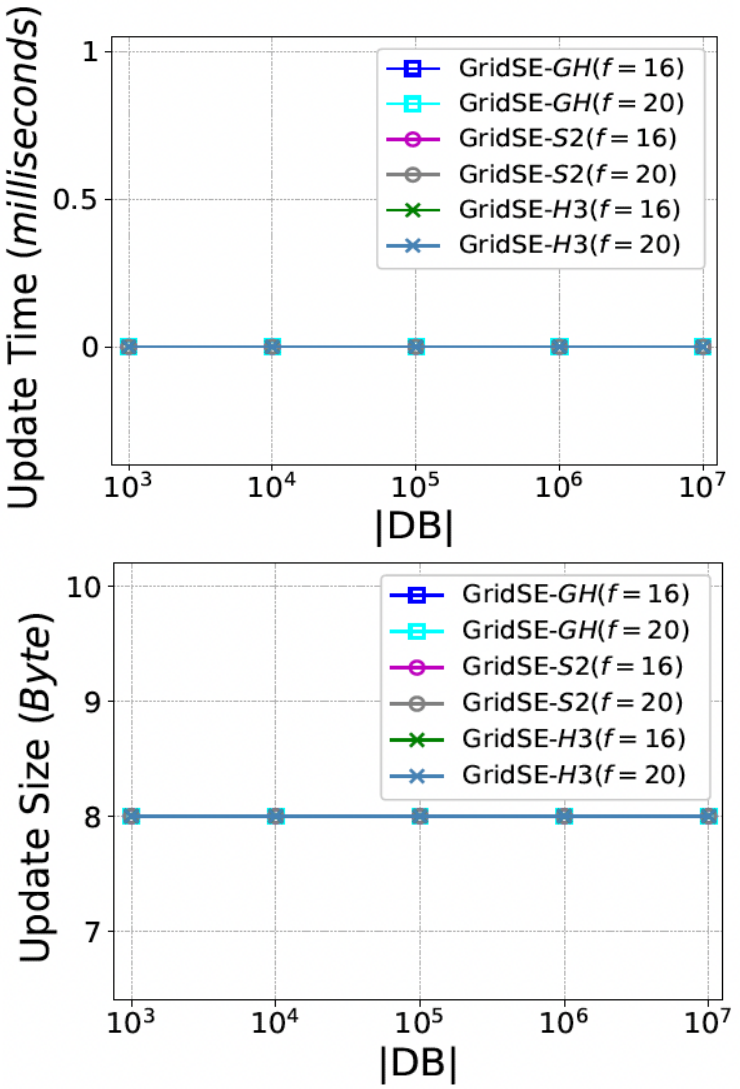
Fixed $|DB| = 10\text{^}6$, $|Q| = 10\text{^}4$

Evaluation

- **Communication**







(b)

- **Update**

Conclusion

- **GridSE** A dynamic prefix SSE scheme
  - Fast SGS with updates
  - Backward and forward privacy
- **SP^2E** A new crypto primitive
- **Performance**
  - 150X -5000X speedup in search time, 99% saving in communication cost than SOTA
  - 1.4X more computation cost and 0.9X more communication overhead than plaintext search

Q & A

- **Contact**

Ruoyang Guo

gruoyang@stevens.edu