Trade-offs in Private Search

Vasilis Pappas  Mariana Raykova  Binh Vo  Steven Bellovin  Tal Malkin

Columbia University

Private Search Components

- Who provides the database and who submits queries? – data outsourcing (same party) vs. data sharing (different parties)
- Are there additional participants in the protocol? – various utility: computation, storage (cloud computing), semi-honest behavior
- Do queries need to be authorized? – how is the search capability provided to users and at what data granularity
- What is the search functionality? – simple matching, compound queries (AND, OR, predicates)
- Are we interested in returning content, or simple result handles? – e.g. document search, database queries
- Do we want to provide anonymity for the querier? – multiple users

Trade-offs of Privacy for Efficiency

Querier learns matching data and any other party learns nothing. Strongest privacy and least efficiency. Privacy relaxations for efficiency improvement:

- Allow leakage of search pattern of queries for a single user or within a group of users.
- Allow leakage of the pattern of matched results for the queries of a single user or within a group of users.
- Allow a bounded number of false positives (compromising database privacy) or false negatives (compromising result integrity).
- Allow leakage of structural information about the database – e.g. similarity between documents

Tunable Search System

Participants, security requirements, trust models, available resources → privacy-efficiency trade-offs → privacy relaxations

Provide tunable controls that allow the users to set the trade-offs between privacy guarantees and efficiency performance at different levels. Characteristics of the underlying protocol:

- Consists of independent modules
- Each module can be instantiated with multiple implementations.
- Different instantiations of the modules give different privacy guarantees and costs.

SADS Tunable Search System

SADS (Secure Anonymous Database Search)

Components: Enc. Bloom filters, Deterministic priv. key enc. PH-DSAEP+, Re-routable encryption

Document Retrieval

- Preprocessing – IS obtains each document encrypted under different key, does not know keys, Server knows the keys but not the correspondence between keys and documents.
- Retrieval – Client obtains document key indices from IS and obtains the corresponding decryption keys from Server

Tunable Controls

- Configurable architecture – IS and QR present or absent, different leakage to data owner
- Search with or without document retrieval – immediate or accumulated requests for decryption key from Server
- Same or different hash functions across different BFs – structural leak to IS, dependent false positive rates across BFs, parallel search across BFs through bitslicing storage
- Exponentiation caching in preprocessing – depends on the type of the database when whether it has repeating entries across documents
- Parallelization of preprocessing and search – utilize resources

Performance Evaluation

Average Query Time

Timings for different variations

Comparison to SQL (no privacy) for database record retrieval

MITLL http://www.ll.mit.edu/CST/

Duration of the preprocessing phase