Applying MPC at Scale

- Suyash Rathi

Papers

- ANONIZE: A Large-Scale Anonymous Survey System
- Blind Seer: A Scalable Private DBMS

ANONIZE GOAL: Anonymity vs Authenticity

- Scenario: Medical Survey
- Authenticity: Only Legitimate users can give the survey
- Anonymity: No link between user and survey data for honest feedback

Issues

- Scenario: College Course Review
- Side channel attack can be used to get the order in which students completed the survey.
- Third parties may not be secure. Eg. Cornell incident.
- Cryptographic voting technique – current techniques:
  - Users check-out a single use “token” from server – not related to user.
  - User completes the survey using the token.
  - TIME attack – clear link.

Literature: PRF (Wiki)

- In cryptography, a pseudorandom function family, abbr. PRF, is a collection of efficiently-computable functions which emulate a random oracle in the following way: no efficient algorithm can distinguish (with significant advantage) between a function chosen randomly from the PRF family and a random oracle (a function whose outputs are fixed completely at random).
- The guarantee of a PRF is that all its outputs appear random, regardless of how the corresponding inputs were chosen, as long as the function was drawn at random from the PRF family

Literature: NIZK (Wiki)

- Non-interactive zero-knowledge (NIZK) proof systems are fundamental cryptographic primitives used in many constructions of various cryptographic protocols.

Definition of Ad-hoc survey system

- An *ad-hoc survey scheme* Γ is an 8-tuple of PPT algorithms and interactive PPTs (GenRA,GenSA,Reg^{RA},Reg^{U}, GenSurvey, authorized, Submit, Check) where
- GenRA(1^n) outputs a key-pair pk_{RA}, sk_{RA}.
- GenSA(1^n) outputs a key-pair pk_{SA}, sk_{SA}.
- Reg^{RA}(sk_{RA}, 1^n, pk_{RA}, id_i) is an interactive PPT that outputs either success or fail.
- Reg^{U}(1^n, pkRA, *id*) is an interactive PPT that outputs a bitstring $cred_{id}$ or fail.
- GenSurvey(1^n, sid, *L*, sk_{SA}) outputs a bitstring pk_{sid}. Here sid is a unique arbitrary identifier and *L* is a description of the set of users eligible to participate in the survey.
- Authorized(pk_{SA}, sid, pk_{sid}, *id*) outputs either YES or NO.
- Submit(1^n, sid, pk_{sid},*m*, $cred_{id}$) outputs Sub = (tok,*m*, tokauth).
- Check(pk_{RA}, pk_{SA}, sid, pk_{sid}, Sub) outputs either accept or fail.

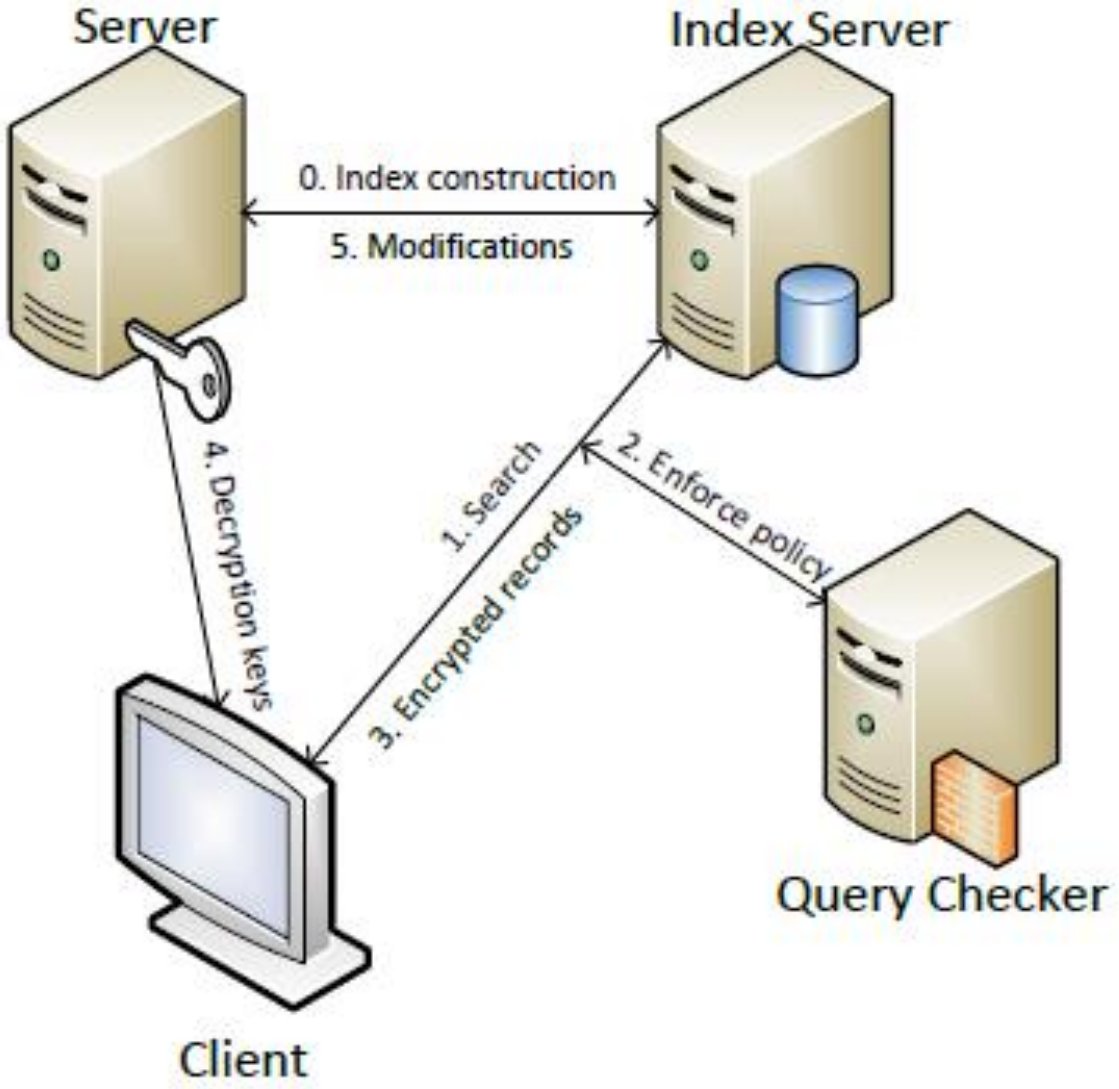
Summary

- Securely implements a million-person anonymous surveys using a single modern workstation.

Blind Seer(BLoom filter INDex SEarch of Encrypted Results) Goal

- Almost all earlier applications considered limited queries.
- Provided weak guarantee of privacy.
- We have other solutions for privacy like FHE, ORAM and MPC which are way too expensive to be applied to practical uses (polynomial time).
- Therefore we try to solve : **Query privacy in secure DBMS** keeping scalability, functionality (efficient sublinear search for arbitrary Boolean queries) and usability in mind.

Model


Server
Index Server
0. Index construction
5. Modifications
4. Decryption keys
1. Search
3. Encrypted records
2. Enforce policy
Client
Query Checker

Approach

- In large problem, we find small privacy critical parts & solve them securely.
- Large problem (encrypted search on large DB) is done by traversing an encrypted search tree.
- Input is secure; intermediate and output are in plain text revealing some information (to make search sublinear).
- Search tree traversal decision is made by Secure Function Evaluation (SFE) using Yao's GC protocol.
- Bloom filters are used to store keywords in each tree node. It gives constant time access and invariant traversal patterns.

Literature:

Notations. Let $[n] = \{1, \ldots, n\}$. For ℓ-bit strings a and b, let $a \vee b$ (resp., $a \wedge b$ and $a \oplus b$) denote the bitwise-OR (resp. bitwise-AND and bitwise-XOR) of a and b. Let $S = (i_1, i_2, \ldots, i_\eta)$ be a sequence of integers. We define a projection of $a \in \{0,1\}^\ell$ on S as $a\downarrow_S = a_{i_1} a_{i_2} \cdots a_{i_\eta}$; for example, with $S = (2,4)$, we have $0101\downarrow_S = 11$. We also define a filtering of $a = a_1 a_2 \ldots a_\ell$ by S as $a\ddagger_S = b_1 b_2 \ldots b_\ell$ where $b_j = a_j$ if $j \in S$, or $b_j = 0$ otherwise; for example, with $S = (2,4)$, we have $1110\ddagger_S = 0100$.

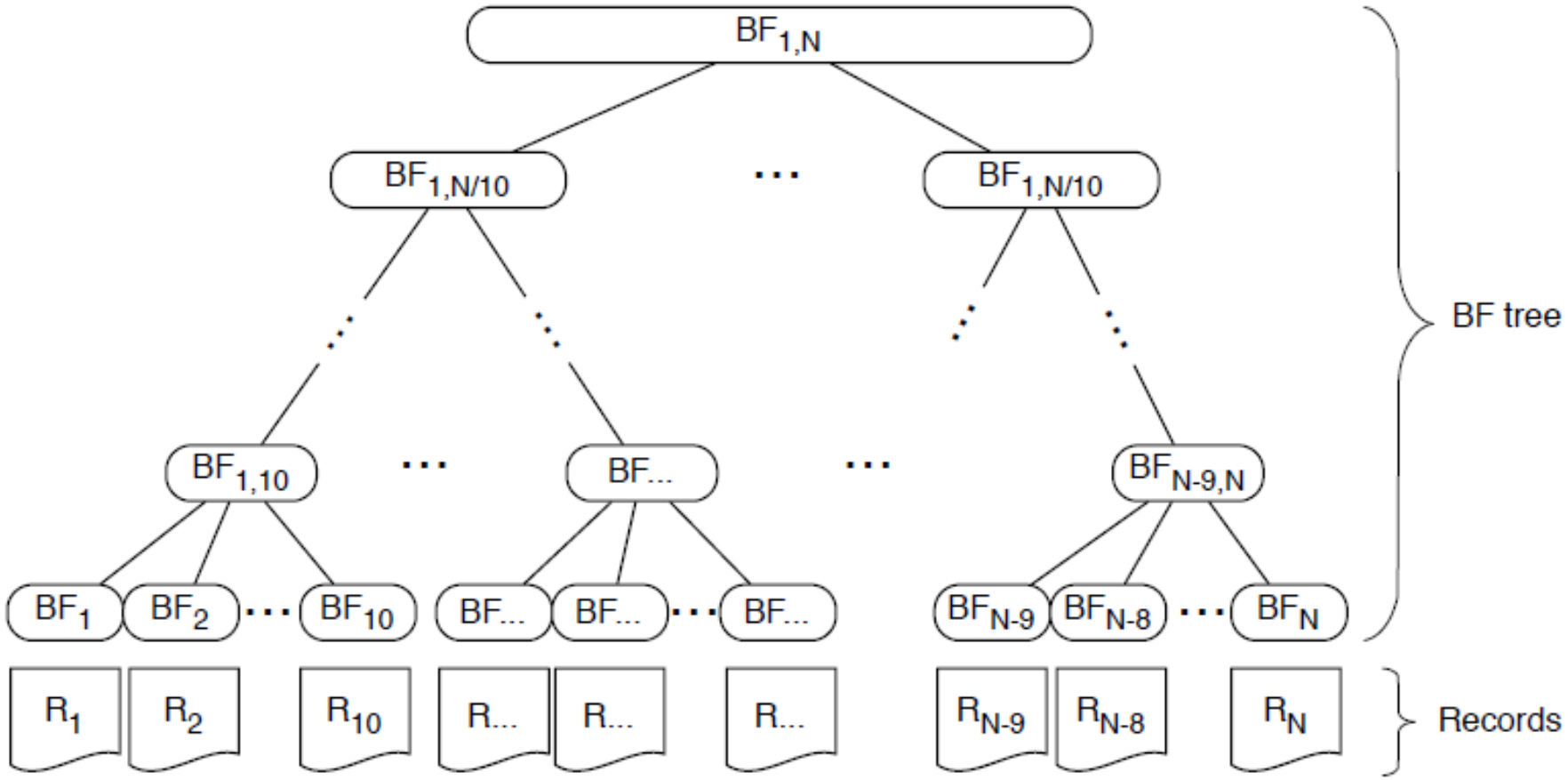
Literature:

Bloom filter (BF). A Bloom filter is a well-known data structure that facilitates efficient search. The filter B is a string initialized with 0^ℓ and associated with a set of η different hash functions $\mathcal{H} = \{h_i : \{0,1\}^* \to [\ell]\}_{i=1}^{\eta}$. For a keyword $\alpha \in \{0,1\}^*$, let $\mathcal{H}(\alpha)$ the sequence of the hash results of α, i.e.,

$$\mathcal{H}(\alpha) = (h_1(\alpha), h_2(\alpha), \ldots, h_\eta(\alpha)).$$

To add a keyword α to the filter, the hash result $\mathcal{H}(\alpha)$ is added to it, that is, $B := B \vee (1^\ell \ddagger_{\mathcal{H}(\alpha)})$.

Bloom Filter Search tree



Bloom Filter Search tree

- Key data structure enabling sublinear search is a BF search tree for the database records
- There is only one global search tree for the entire database
- In the search tree, each leaf is associated with each database record, and each node v is associated with a Bloom filter B_v
- Filter B_v contains all the keywords from the (leaf) records that the node v have (as itself or as its descendants)
- Example, if a node contains a record that has Jeff in the fname field, a keyword = ‘fname:Jeff’ is inserted to B_v

Search using BF search tree

Search($\alpha \wedge \beta, v$):
  If the BF B_v contains α and β, then
    If v is a leaf, then output $\{v\}$.
    Otherwise, output $\bigcup_{c:\text{ children of } v}$ Search($\alpha \wedge \beta, c$).
  Otherwise, output $\emptyset$.

- α and β are the keywords to search
- Example: Search(α , β ,root) will give all entries with the 2 keywords

Preprocessing – Encrypting DB

- Server S shuffles and encrypts its records
- S creates the BF search tree using the new records.
- Each node gets a Bloom filter B_v which is encrypted using PRF F with key k.
- S sends the encrypted DB and search tree to IS.
- Client gets the PRF key k and the hash functions for the Bloom filter generation.

Traversing Search tree privately

- Search procedure starts with transforming the query into Boolean circuit
- Client and IS compute the circuit via secure computation.
- If the circuit outputs true, it will again evaluate on that node recursively till it reaches a leaf node or else it terminates at that node.
- If the client gets the leaf node successfully, IS will send it $(\psi(i), r_i, \tilde{R}_i)$.
- Client sends $\psi(i)$ to S and gets $s'_{\psi(i)} = s_i r_i$.
- C decrypts $\tilde{R}_i$ using s_i and obtains the output record.

Summary

- Instead of complete privacy we go for a weaker privacy as a trade off for greater efficiency.
- Peroformance of just 1.2x to 3x slowdown was achieved.
- The performance is due to the search tree traversal access patterns being revealed to the parties.

References

- Blind Seer: A Scalable Private DBMS, SP14
- ANONIZE: A Large-Scale Anonymous Survey System, SP14

Thank You!