CERIAS

The Center for Education and Research in Information Assurance and Security

PURDUE UNIVERSITY

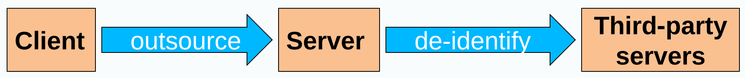
Query Processing in Private Data Outsourcing Using Anonymization

Ahmet Erhan Nergiz
anergiz@cs.purdue.edu

Chris Clifton
clifton@cs.purdue.edu

Motivation

Data outsourcing benefits:
- Cheaper (affordable by small companies)
- High availability (24/7 access guarantee)
- Service quality (having experts on the field)



Patient	Age	City	Disease
Ike	20	Dayton	Cold
Eric	22	Richmond	Fever
…….	…..	……..	…….

Identifier — Quasi-identifiers — Sensitive info

Age	City	Disease
20	Dayton	Cold
22	Richmond	Fever
…..	……..	…….

Only Ike's age is 20 in Dayton

Re-identify

Ike has Cold

Our Contribution

Encrypting whole database is not the answer
- Value-added services: Data analysis, Data cleaning
- "Sell" the data (or analysis of the data): Make the service free to the client!

Why not use existent work on data publishing (anatomy based on l-diversity)?

Patient	Age	City	Disease
Ike	41	Dayton	Cold
Eric	22	Richmond	Fever
Olga	30	Lafayette	Flu
Kelly	35	Lafayette	Cough

Anonymize

Age	City	GID
41	Dayton	1
22	Richmond	1
30	Lafayette	2
35	Lafayette	2

GID	City
1	Cold
1	Fever
2	Flu
2	Cough

Related Work

Bucketization (Hacigumus et al.):

Patient	Age	City	Disease
Ike	41	Dayton	Cold
Eric	22	Richmond	Fever
…….	…..	……..	…….

Patient	Age	City	Disease	etuple
23	6	3	2	E(tuple1)
46	4	9	9	E(tuple2)
…….	…..	……..	…….	E(tuple3)

How We Do It

Patient	Age	City	GID	SEQ
Ike	41	Dayton	1	1
Eric	22	Richmond	1	2
Olga	30	Lafayette	2	3
Kelly	35	Lafayette	2	4
Faye	24	Richmond	3	5
Mike	47	Richmond	3	6
Jason	45	Lafayette	4	7
Max	31	Lafayette	4	8

$Patient_{IT}$ = Identifying table

H(SEQ)	GID	Disease
$H_{k2}(1)$	1	Cold
$H_{k2}(2)$	1	Fever
$H_{k2}(3)$	2	Flu
$H_{k2}(4)$	2	Cough
$H_{k2}(5)$	3	Flu
$H_{k2}(6)$	3	Fever
$H_{k2}(7)$	4	Cough
$H_{k2}(8)$	4	Flu

$Patient_{ST}$ = Sensitive table

Proposal: Server does partial query processing on the anatomized database, client opens the encrypted link and joins IT and ST to complete the query

Selection

Idea: Convert selection into subqueries that can be processed on the IT and ST

SELECT * FROM Patient
WHERE Age > 40 AND (Disease = Flu OR Disease = Cough) AND
(Disease = Cough OR Address = Lafayette)

P	Age	City	G	S
Ike	41	Dayton	1	1
Mike	47	Rich	3	6
Jason	45	Laf	4	7

P	Age	City	G	S
Mike	47	Rich	3	6
Jason	45	Laf	4	7

P	Age	City	G	S
Jason	45	Laf	4	7

H(S)	G	Disease
$H_{k2}(3)$	2	Flu
$H_{k2}(4)$	2	Cough
$H_{k2}(5)$	3	Flu
$H_{k2}(7)$	4	Cough
$H_{k2}(8)$	4	Flu

H(S)	G	Disease
$H_{k2}(5)$	3	Flu
$H_{k2}(7)$	4	Cough
$H_{k2}(8)$	4	Flu

H(S)	G	Disease
$H_{k2}(7)$	4	Cough
$H_{k2}(8)$	4	Flu

Client joins the two tables and get:

Patient	Age	Address	G	S	H(S)	G	Disease
Jason	45	Lafayette	4	$H_{k2}(7)$	$H_{k2}(7)$	4	Cough

Group By

SELECT AVG(Age) FROM Physician, Patient GROUP BY Gender, City

Doctor	Gender	G	S
Alice	Female	1	1
Carol	Female	1	2
Bob	Male	2	3
Dave	Male	2	4
Carol	Female	3	5
Alice	Female	3	6
Dave	Male	4	7
Carol	Female	4	8

$Physician_{IT}$

H(S)	G	Patient
$H_{k1}(1)$	1	Ike
$H_{k1}(2)$	1	Eric
$H_{k1}(3)$	2	Olga
$H_{k1}(4)$	2	Kelly
$H_{k1}(5)$	3	Faye
$H_{k1}(6)$	3	Mike
$H_{k1}(7)$	4	Jason
$H_{k1}(8)$	4	Max

$Physician_{ST}$

Patient	Age	City	G	S
Ike	41	Dayton	1	1
Eric	22	Richmond	1	2
Olga	30	Lafayette	2	3
Kelly	35	Lafayette	2	4
Faye	24	Richmond	3	5
Mike	47	Richmond	3	6
Jason	45	Lafayette	4	7
Max	31	Lafayette	4	8

$Patient_{IT}$

Server processes the group-by partially and send the partial result to the client with the rest of the tuples:

Gender	City	AVG(Age)	COUNT(*)
Female	Dayton	41	1
Female	Richmond	31	3
Male	Lafayette	32.5	2

Client processes the group by query on the rest of the tuples by decrypting the link and joining the tables:

Gender	City	AVG(Age)	COUNT(*)
Male	Lafayette	45	1
Female	Lafayette	31	1

Client merges the two results to get the actual result

Gender	City	AVG(Age)
Female	Dayton	41
Female	Lafayette	31
Female	Richmond	31
Male	Lafayette	36.67

Work In Progress

Implement it on top of a DBMS
Support data manipulation (insert, delete, update)
Challenges:
- Need incremental anatomization
- Previous works support only insert and delete w/o incremental anatomization
- Incremental anatomization needs to consider all inference channels existing between snapshots of the database

Future Work

Integrate it to safe grouping for transactional datasets

This publication was made possible by the support of the NPRP grant 09-256-1-046 from the Qatar National Research Foundation. Partial support for this work was provided by MURI award FA9550-08-1-0265 from the Air Force Office of Scientific Research.

Discovery Park e-Enterprise Center