Advanced cryptography module:
Paper presentation

Presented by:

- **Ben Ayad**, Mohamed Ayoub

Supervised:

- **Moataz**, Tarik

Generic Attacks on Secure Outsourced Databases

Authors:

- **Georgios Kellaris**
- **George Kollios**
- **Kobbi Nissim**
- **Adam O’Neil**

Outline

1. General context
    a. Problematic
        i. Outsourced databases
    b. Solutions
    c. The idea of the paper
    d. Related work
2. The model: Set up
3. Attacks:
    a. Attack using the communication volume
4. Experiments
5. Conclusion

General context - Problematic

Outsourced databases model

Example: **Big companies** with strong customer base:

- e-commerce websites.
- banks.
- ensurance .

General context - Problematic

Problematic:

Potential move to the cloud ?

General context - Problematic

Questions:

>>> Security VS Efficiency <<<

General context - Solutions

Cryptographic solutions:

- FHE
- ORAM
- Searchable encryption

Practical solutions:

- Deterministic encryption | Order preserving encryption
  - CryptDB | CipherBase
  - >> More practical but leaks information <<

General context - Down sides

- Most of these systems do leak information

- Which leads to Attacks

General context - The paper

- Generic Approach for outsourced DB system
  - Implementation free

- Present an attack depending on the leakage mode.

Leakages Types

- **Access pattern:**
  - Which “encrypted” records are returned as the result of a query

encrypted query 1	id1, id5, id7, ...
encrypted query 2	id3, id5 , id1, ...
….	

What the attacker sees

Leakages Types

- **Communication volume:**
    - We know learn how many encrypted records are returned as a result of a query

# of records	# of queries
0	u0
1	u1
2	u2 = 13
3	u3
4	u4

What the attacker sees

Related work

- Previous work exploiting the access pattern leakage:
  - **M. S. Islam, M. Kuzu, and M. Kantarcioglu.**
    - **Access pattern disclosure on searchable encryption:Ramification, attack and mitigation.**
  - **J. L. Dautrich Jr and C. V. Ravishankar:**
    - **Compromising privacy in precise query protocols.**
      - **Assumptions ?**
  - **M. Naveed, S. Kamara, and C. V. Wright.**
    - **Inference attacks on property-preserving encrypted databases.**
      - **Assumptions?**
- First Attack considering the communication volume leakage..

The paper main tool

- <u>Reconstruction Attack</u> (of the search keys) :
  - Type 1: Based on access pattern
  - Type 2: Based on communication volume

- Assumptions and limitations:
  - No required information on queries or answers
  - Range queries
  - Uniform queries

General context - Range queries

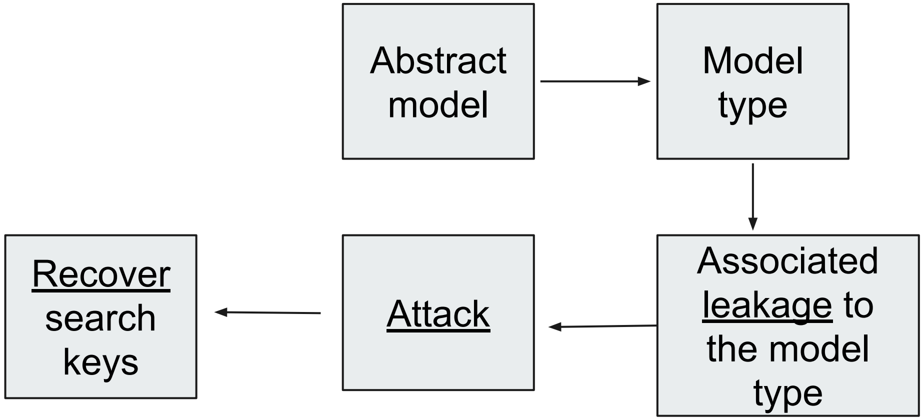
- <u>Queries Types:</u>
  - Point queries
  - Range queries

ID	First Name	Last Name	Email	Year of Birth
1	Peter	Lee	plee@university.edu	1992
2	Jonathan	Edwards	jedwards@university.edu	1994
3	Marilyn	Johnson	mjohnson@university.edu	1993
6	Joe	Kim	jkim@university.edu	1992
12	Haley	Martinez	hmartinez@university.edu	1993
14	John	Mfume	jmfume@university.edu	1991
15	David	Letty	dletty@university.edu	1995

Table: Students

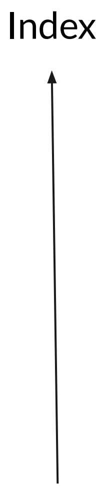
Students born between
91 -- 96 ??

General Road Map of the paper



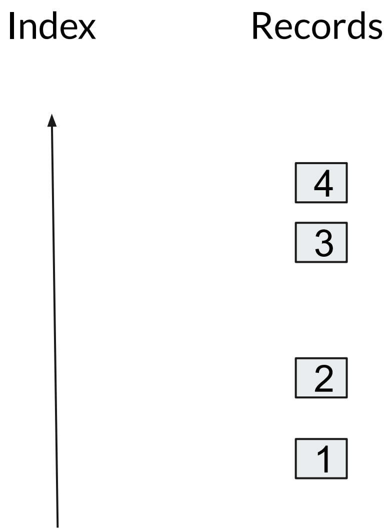
Attack #2: Using <u>communication volume</u>

<u>The setting</u>



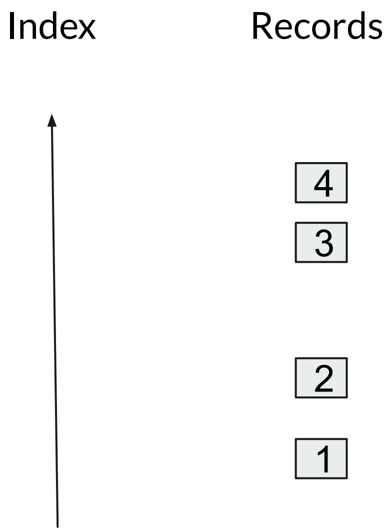
Attack #2: Using communication volume

The setting



Attack #2: Using communication volume

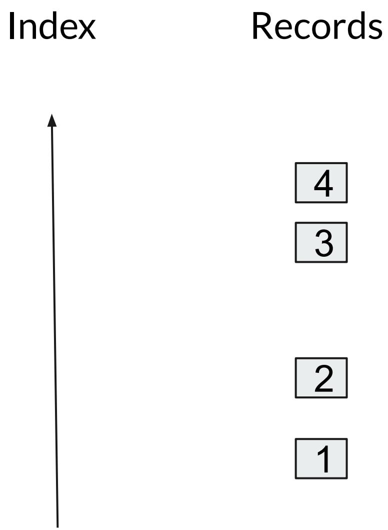
The setting



Goals:
- Ordering
- Position

Attack #2: Using **communication volume**

The setting

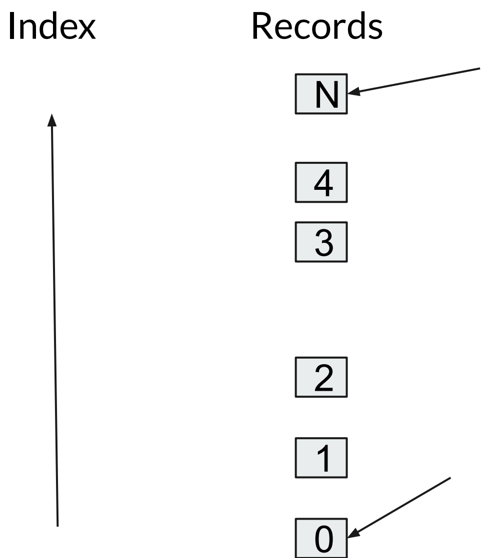


Goals:
- Ordering
- Position

# of records	# of queries
0	u0
1	u1
2	u2
3	u3
4	u4

Attack #2: Using communication volume

The setting



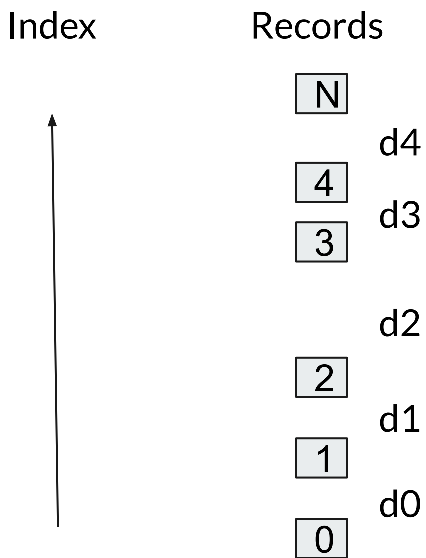
Goals:

- Ordering
- Position

# of records	# of queries
0	u0
1	u1
2	u2
3	u3
4	u4

Attack #2: Using communication volume

The setting

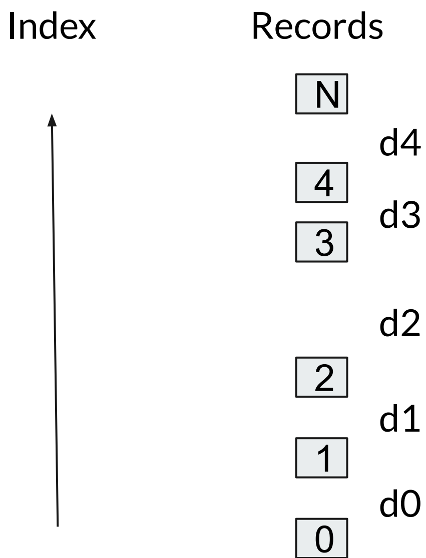


Goals:
- Ordering
- Position

# of records	# of queries
0	u0
1	u1
2	u2
3	u3
4	u4

Attack #2: Using <u>communication volume</u>

<u>The setting</u>



# of records	# of queries
0	u0
1	u1
2	u2
3	u3
4	u4

Knowing <u>**di**</u> for i: 0 -> N

Knowing the **positions** and **ordering**

Attack #2: Using <u>communication volume</u>

Knowing <u>**di**</u> for i: 0 -> N

↓

Knowing the **positions** and **ordering**

# of records	# of queries
0	u0
1	u1
2	u2
3	u3
4	u4

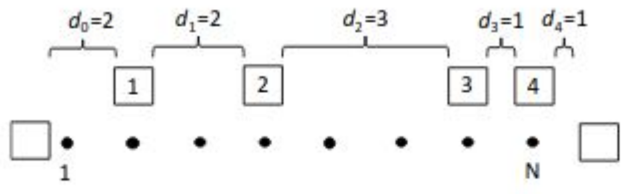
How can we do it starting from ui ??!!

Attack #2: Using communication volume

# of records	# of queries
0	u0
1	u1
2	u2
3	u3
4	u4

How can we do it starting from ui ??!!

Attack #2: Using communication volume



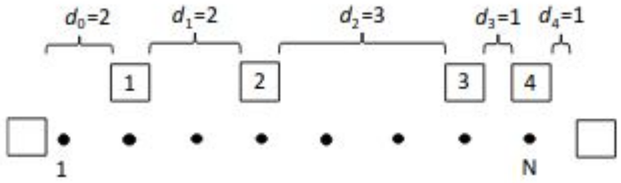
We can see that :

- **N = 8**

We have this system:

$$d_0 \cdot d_n = \mathbf{u_n}$$
$$d_0 \cdot d_{n-1} + d_1 \cdot d_n = \mathbf{u_{n-1}}$$
$$d_0 \cdot d_{n-2} + d_1 \cdot d_{n-1} + d_2 \cdot d_n = \mathbf{u_{n-2}}$$
$$\ldots$$
$$d_0 \cdot d_1 + d_1 \cdot d_2 + \cdots + d_{n-1} \cdot d_n = \mathbf{u_1}$$
$$(d_0)\text{^}2 + \ldots + (d_n)\text{^}2 = 2 \cdot \mathbf{u_0} + N + 1$$

Attack #2: Using communication volume



We can see that :

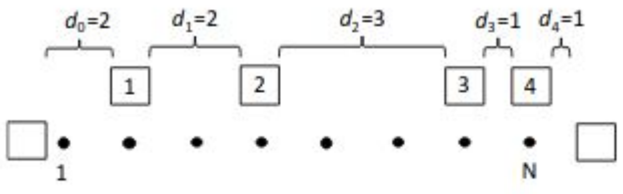
- **N = 8**

We have this system:

$$
\begin{gathered}
d_0 \cdot d_n = \mathbf{u_n} \\
d_0 \cdot d_{n-1} + d_1 \cdot d_n = \mathbf{u_{n-1}} \\
d_0 \cdot d_{n-2} + d_1 \cdot d_{n-1} + d_2 \cdot d_n = \mathbf{u_{n-2}} \\
\ldots \\
d_0 \cdot d_1 + d_1 \cdot d_2 + \cdots + d_{n-1} \cdot d_n = \mathbf{u_1} \\
(d_0)^2 + \ldots + (d_n)^2 = 2 \cdot \mathbf{u_0} + N + 1
\end{gathered}
$$

How to solve this system ?

Attack #2: Using communication volume



We can see that :

- **N = 8**

We have this system:

$$d_0 \cdot d_n = \mathbf{u_n}$$
$$d_0 \cdot d_{n-1} + d_1 \cdot d_n = \mathbf{u_{n-1}}$$
$$d_0 \cdot d_{n-2} + d_1 \cdot d_{n-1} + d_2 \cdot d_n = \mathbf{u_{n-2}}$$
$$\ldots$$
$$d_0 \cdot d_1 + d_1 \cdot d_2 + \cdots + d_{n-1} \cdot d_n = \mathbf{u_1}$$
$$(d_0)^2 + \ldots + (d_n)^2 = 2 \cdot \mathbf{u_0} + N + 1$$

We define these two polynomials:

$$d(x) = d_0 + d_1 x + d_2 x^2 + \cdots + d_n x^n$$
$$d^R(x) = d_n + d_{n-1} x + d_{n-2} x^2 + \cdots + d_0 x^n;$$
$$F(x) = d(x) \cdot d^R(x)$$

Attack #2: Using communication volume

We have this system:

$$d_0 \cdot d_n = \mathbf{u_n}$$
$$d_0 \cdot d_{n-1} + d_1 \cdot d_n = \mathbf{u_{n-1}}$$
$$d_0 \cdot d_{n-2} + d_1 \cdot d_{n-1} + d_2 \cdot d_n = \mathbf{u_{n-2}}$$
$$\ldots$$
$$d_0 \cdot d_1 + d_1 \cdot d_2 + \cdots + d_{n-1} \cdot d_n = \mathbf{u_1}$$
$$(d_0)\text{^}2 + \ldots + (d_n)\text{^}2 = 2 \cdot \mathbf{u_0} + N + 1$$

We define these two polynomials:

$$d(x) = d_0 + d_1 x + d_2 x^2 + \cdots + d_n x^n$$
$$d^R(x) = d_n + d_{n-1} x + d_{n-2} x^2 + \cdots + d_0 x^n,$$
$$F(x) = d(x) \cdot d^R(x)$$

Turns out that, F(x) equals:

$$F(x) = u_n x^{2n} + u_{n-1} x^{2n-1} + \cdots + u_0 x^n + \cdots + u_{n-1} x + u_n$$

Attack #2: Using communication volume

We have this system:

$$d_0 \cdot d_n = \mathbf{u_n}$$
$$d_0 \cdot d_{n-1} + d_1 \cdot d_n = \mathbf{u_{n-1}}$$
$$d_0 \cdot d_{n-2} + d_1 \cdot d_{n-1} + d_2 \cdot d_n = \mathbf{u_{n-2}}$$
$$\ldots$$
$$d_0 \cdot d_1 + d_1 \cdot d_2 + \cdots + d_{n-1} \cdot d_n = \mathbf{u_1}$$
$$(d_0)\text{^}2 + \ldots + (d_n)\text{^}2 = 2 \cdot \mathbf{u_0} + N + 1$$

We define these two polynomials:

$$d(x) = d_0 + d_1 x + d_2 x^2 + \cdots + d_n x^n$$
$$d^R(x) = d_n + d_{n-1} x + d_{n-2} x^2 + \cdots + d_0 x^n,$$
$$F(x) = d(x) \cdot d^R(x)$$

Turns out that, F(x) equals:

$$F(x) = u_n x^{2n} + u_{n-1} x^{2n-1} + \cdots + u_0 x^n + \cdots + u_{n-1} x + u_n$$

# of records	# of queries
0	u0
1	u1
2	u2
3	u3
4	u4

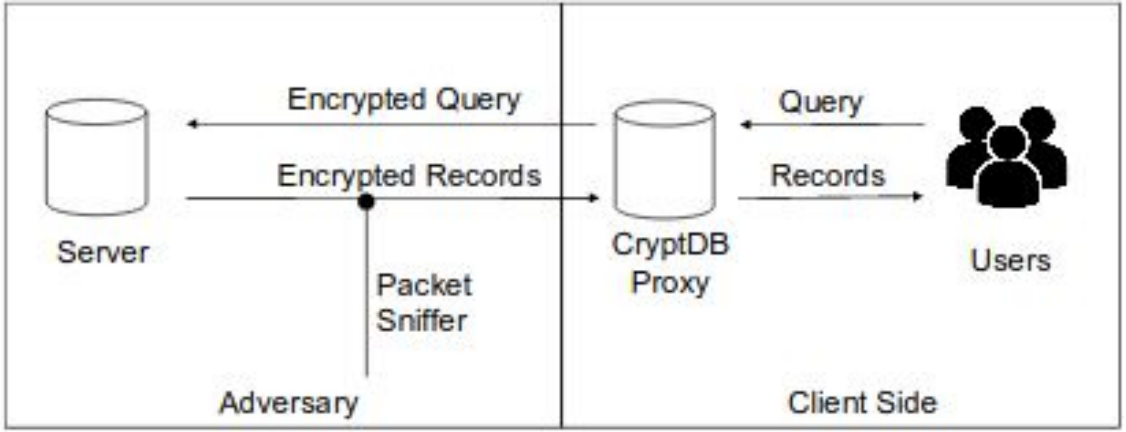
Complexity

T is the domain size:

- For order recovery:
  - $O((T^2) * \log(T))$ queries are needed

- For full reconstruction:
  - $O(T^4)$ queries are needed

Experiments - Set up



Experiments - Datasets

Source	Datasets	Index	N	n(max)	n(avg)
PUDF	518	Mortality Risk	4	55,605	5,612
		Age (<18)	6	20,454	1,170
		Age ($\geq$ 18)	16	34,162	4,130
		Age (All)	22	50,626	5,300
		Length of Stay	365	55,605	5,612
NIS	1049	Age (<18)	18	16,954	1,195
		Age ($\geq$ 18)	107	106,252	6,240
		Age (All)	125	121,663	7,435
		Length of Stay	365	121,663	7,435

Experiments - Attack 1

Source	Index	Ordering	Positions	Dense
PUDF	Mortality Risk	1 ms	1 ms	85%
	Age (<18)	1 ms	1 ms	34.1%
	Age ($\geq$ 18)	1 ms	1 ms	67.3%
	Age (All)	1 ms	1 ms	32.2%
	Length of Stay	43 ms	4.2 sec	0%
NIS	Age (<18)	1 ms	1 ms	31.5%
	Age ($\geq$ 18)	1 ms	202 ms	0%
	Age (All)	1 ms	356 ms	0%
	Length of Stay	5 ms	3.4 sec	0%

Experiments - Attack 2

Source	Index	Factor ($n \leq 150$)	BruteForce
PUDF	Mortality Risk	11 min	22 ms
	Age (<18)	39 sec	1.7 ms
	Age ($\geq$ 18)	4.3 min	15 ms
	Age (All)	4.1 min	390 ms
	Length of Stay	3 min	22 ms
NIS	Age (<18)	3.3 min	2 ms
	Age ($\geq$ 18)	6 min	34 ms
	Age (All)	5.1 min	189 ms
	Length of Stay	4 min	44 ms

Conclusion

- Generic model to capture outsourced databases.
- Two attacks depending on the type of leakage
  - Pattern access
  - Communication volume
- Their efficiency on real life databases
- Outsourced databases should avoid:
  - Being static non storage inflating
  - Being with fixed communication overhead

Open questions

- The case of non-uniform queries.
- Models that leaks communication volume.
- The case where communication volume is perturbed:

Thank you