Searchable Encryption & its attacks

Kaitai Liang EEMCS, TU Delft, The Netherlands

OUTLINE

- What is SSE?
- What are the attacks?
- Future works?

Cloud-based data

Profile Information

Medical History

E-Banking Record

Genome Data

a great amount of various, dynamic, non-uniform data

TU Delft

Cloud-based data – confidentiality

Profile Information

Medical History

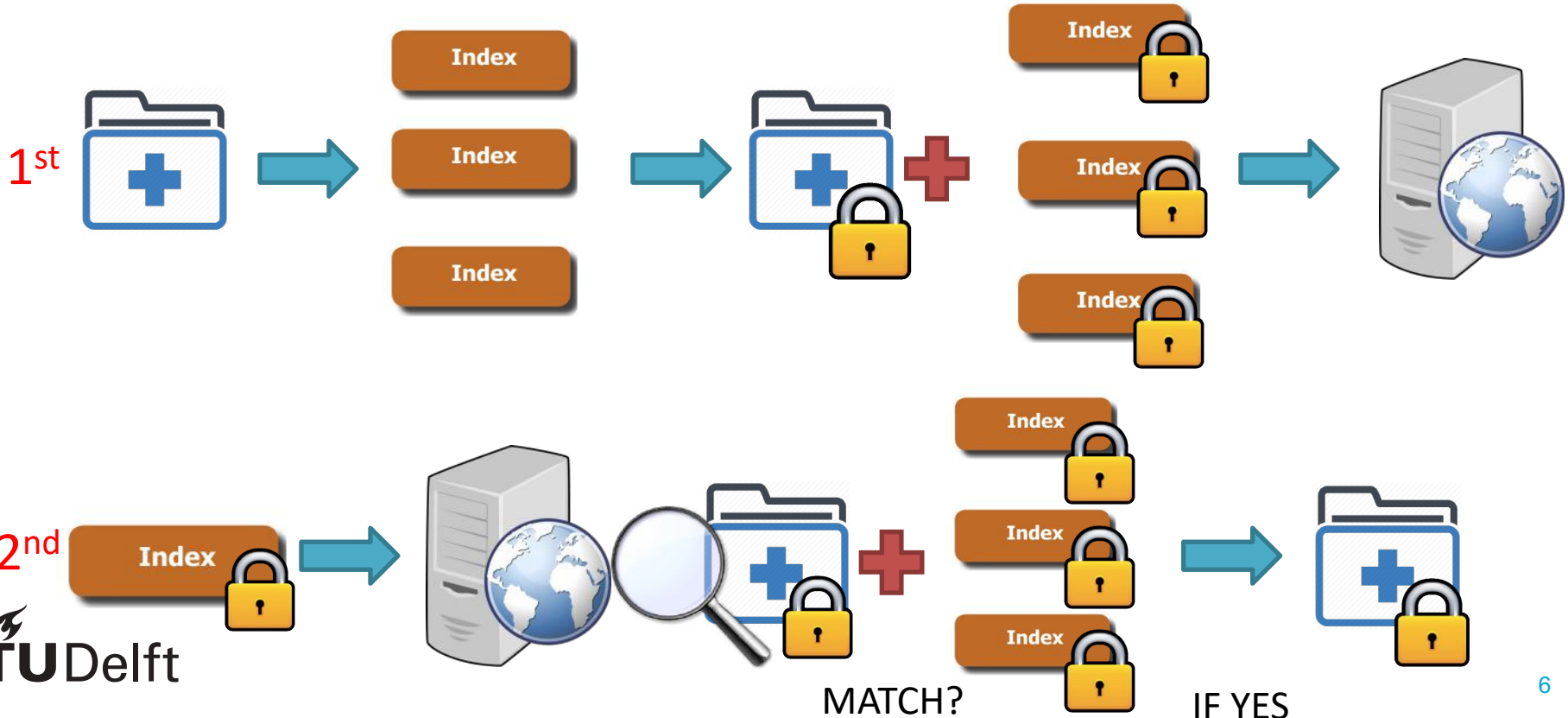
E-Banking Record

Genome Data

Cloud-based data – confidentiality

Encryption enhances privacy;
BUT it also limits the scalability of some behaviors.

Searchable Encryption



TU Delft

Searchable Encryption

1. Symmetric SE (SSE):

[Goh03], [CM05], [CGKO06], [LSD+10], [CJJ+13]…..

symmetric tools, fast, find a way to tighten index structure and ciphertext,

Simulation based proof, leakage function

Many attacks.

2. Asymmetric SE (ASE):

[BCOP04], [Abdalla+08], [BNS06], [Khader10], [ZI09], [BW07],[ZXA14]……

Asymmetric tools, slow, expressiveness, natural bound,

Game based proof, less attacks.

ASE worries

- Inside/outside – online/offline KGA. Due to the encryption algorithm is publicly known also the public key.
- In general, that is tackled[1]

1. 2020 Privacy preserving search services against online attack

SSE concerns

- File/document content security
- Keyword privacy (related to query)
  - Search pattern
  - Access pattern
  - Volume pattern

SSE attacks

Leakage abuse attacks

Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation (`12)
Leakage-Abuse Attacks Against Searchable Encryption (`15)
Revisiting Leakage Abuse Attacks (`20)

Search pattern leakage in searchable encryption: Attacks and new construction (`14)
The Shadow Nemesis: Inference Attacks on Efficiently Deployable, Efficiently Searchable Encryption (`16)
Leakage-Hiding the Access Pattern is Not Enough: Exploiting Search Pattern Leakage in Searchable Encryption (`21)

Injection attack

All Your Queries Are Belong to Us: The Power of File-Injection Attacks on Searchable Encryption (`16)
Revisiting Leakage Abuse Attacks (`20)

Attacks aspects

- Goal
  - Query recovery: query keyword information
- Attack Mode
  - Passive: no malicious operations on dataset/ listener
  - Active: injection malicious data
- Target mode
  - Snapshot: just to encrypted dataset
  - Persistent: query transcript + encrypted dataset
- Knowledge
  - Full plain dataset, keyword index, file identifier…..

Leakage abuse attacks

- Current main static/passive attack mode
- Leverage 1-2 pattern(s) + pre-dataset knowledge
- Strong attack condition and assumption, depending on “high” amount of knowledge to obtain high attack accuracy.

2011 Poster-Inference Attacks against Searchable Encryption Protocols

2012 Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation - IKK

2015 Leakage-Abuse Attacks Against Searchable Encryption - Count

2020 Revisiting Leakage Abuse Attacks

TU Delft

Leakage abuse attacks

- Leakage mode:
  - Identifier pattern (access pattern)
  - Query equality pattern (search pattern)

Leakage abuse attacks examples

IKK

background matrix

	W1	W2	W3	W4
W1		8	4	9
W2	8		5	7
W3	4	5		11
W4	9	7	11	

Co-occurrence matrix

	Q1	Q2	Q3	Q4
Q1		11	4	5
Q2	11		7	9
Q3	4	7		8
Q4	5	9	8	

Mapping from one to another; query to keyword
But too high pre-knowledge 95%

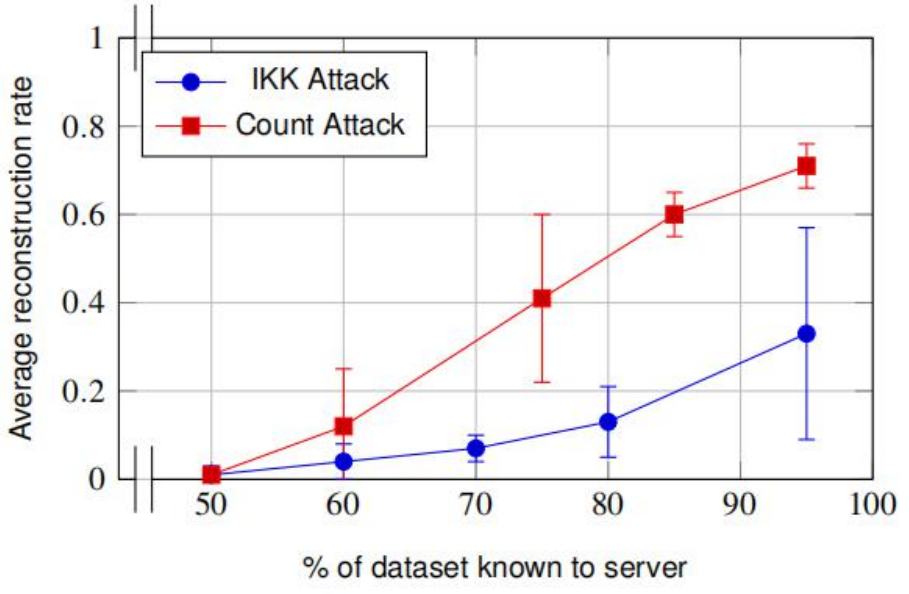
TU Delft

Leakage abuse attacks examples

count

① Pre-knowledge

keyword	M. count
W1	8
W2	1
W3	5
W4	5
W5	1
W6	2
W7	1



③ matching reflection in matrix

	W5	W6	W7
	1	1	1
	0	0	0
	1	0	0
	0	1	1
		0	0
	0		1
	0	1	

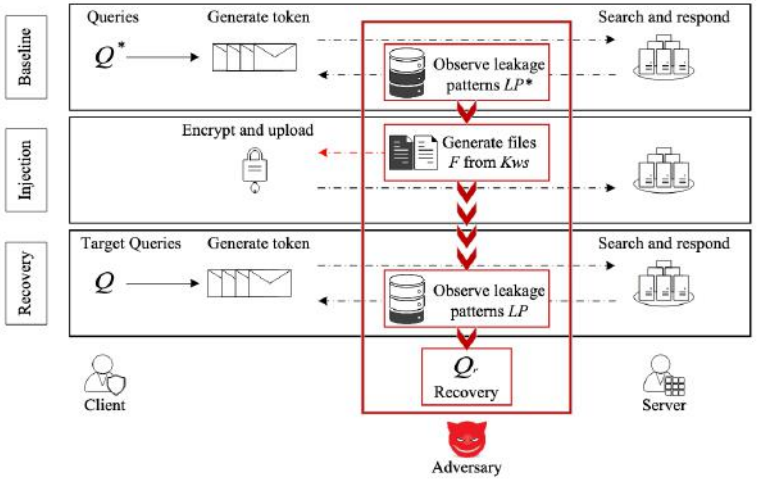
Q2	1		0	3
Q3	0	0		1
Q4	1	3	1	

Q1 -> W6
Q4 -> W1

Q2 -> W4
Q3 -> W5

Injection attacks

- Inject malicious file (make good use of volume pattern)
- Less condition and assumption (related to access pattern)
- High successful rate – close to 100% depending on the injection amount



Injection attacks examples

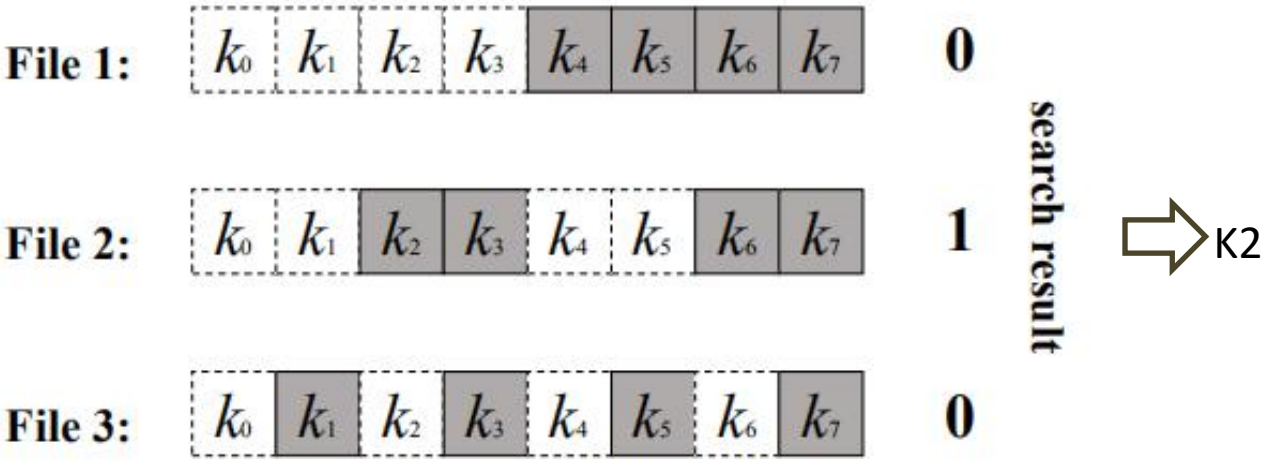
BSA

K = {k0, ..., k7}

Assume: Each file contains half of K keywords

Injection attacks examples

BSA

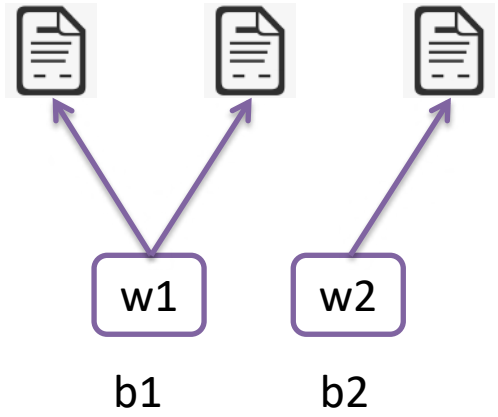


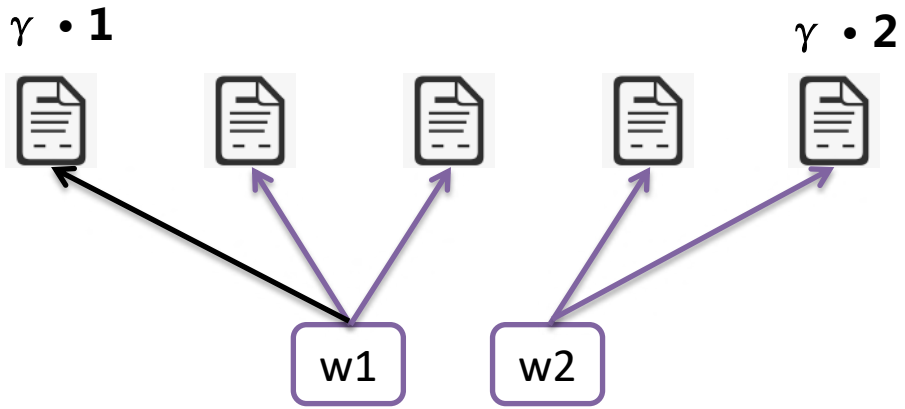
K = {k0, ..., k7}

Assume: Each file contains half of K keywords

Injection attacks examples

For some keyword w_i, inject a file only with w_i and size: γ i



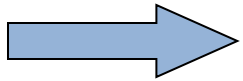


$$\gamma = \min\left\{\gamma \in \mathbb{N} : \forall i, j \in [m], \gamma \not| \, b_i - b_j\right\}.$$

Injection attacks examples

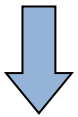
Observing -> (v_1, ... , v_t)
queries q= (q_1, ..., q_t)

$$v_i = b_j + u \cdot \gamma$$



Recover the u and match w_u

v1 = b2+ γ • **2**



Fewer injections high accuracy

Attack	Leakage	Type		Injection volume		Round[2]
		Passive	Injection	Length	Size	
IKK [25]	ap	✓	×	–	–	–
Count [7]	ap,rlp	✓	×	–	–	–
SelVolAn, Subgraph [3]	ap,vp	✓	×	–	–	–
LEAP [34]	ap	✓	×	–	–	–
SAP [35]	sp,rlp	✓	×	–	–	–
ZKP [51]	aip	×	✓	$O(\log \#W)$	$O(\#W \log \#W)$	1
Multiple-round[1][38]	rlp	×	✓	$O(k\#W \log_k \#W)$	$O(k\#W^2)$	$\#W \log_k \#W$
Single-round [38]	rlp	×	✓	$O(m\#W)$	$O(m\#W^2)$	1
Decoding [3]	rsp	×	✓	$O(\#W)$	$O(offset \cdot \#W^2)$	1
Search[1][3]	rsp	×	✓	$O(\#W \log \#W)$	$O(\#W^2)$	$\#W \log \#W$
BVA	rsp	×	✓	$O(\log \#W)$	$O(\gamma\#W)$	1
BVMA	vp, sp[3]	×	✓	$O(\log \#W)$	$O(\#W \log \#W)$	1

[1] Unlike those schemes easily restoring multiple queries, search [3] and multiple-round [38] attacks can only recover a keyword at a time by running many attack rounds. This means that the client should make many queries, and the queries must include the target query at each attack round. The multiple-round is a strategy that depends on query replay, requiring the adversary to evoke the same query repeatedly by controlling the client. These two attacks commit more rounds and injected files (than others in the table) to recover multiple queries.

[2] We here say that an attack round consists of (1) an active and complete injection and then (2) an observation within a specific period. We will present a formal definition in Appendix C.

[3] BVMA mainly investigates the vp to recover the queries. Thus, the sp is an optional and non-essential leakage for the attack (see Appendix E).

Fewer injections high accuracy

Table 5: Running time for query recovery ($10 \times 1,000$ queries for Enron and Lucene, $10 \times 5,000$ queries for Wikipedia). Note the cost of the BVA is within the time range by varying $\gamma = [\#W/2, offset/4]$.

Running time	Decoding	Single-round ($m = 1$)	Single-round ($m = \#W$)	BVA	BVMA
Enron	2.48s	0.01s	0.02s	(1.81s, 2.46s)	19.16s
Lucene	3.54s	0.01s	0.02s	(2.53s, 3.15s)	33.10s
Wikipedia	3min 42s	0.05s	0.06s	(2min 52s, 3min 37s)	15min 34s

Table 6: Performance under the extended SEAL. The overhead (× / No.) represents (the ratio between the extra overhead of padding and "no padding"; the number of files). Setup&Fill and Inj&Fill are the storage overheads on the database setup and file injection, respectively. S-Query and I-Query are the average query bandwidth overheads before and after Inj&Fill, respectively.

Extended SEAL	Overhead (× / No.)				Recovery rate %	
	Setup&Fill	S-Query	Inj&Fill	I-Query	BVA	BVMA
no padding	0 / 30k	0 / 730	0 / 12 (injection)	0 / 735	70	87
$x = 2$	0.5 / 45k	0.4 / 1,058	2,730 / 33k	1.8 / 2,106	< 1	< 1
$x = 4$	1.6 / 79k	1.2 / 1,595	16,384 / 197k	7.6 / 6,316	< 1	< 1
$x = 16$	2.0 / 91k	4.7 / 4,147	81,920 / 983k	88 / 66,027	< 1	< 1

Open problem

- Key compromise (tackled by “The power of bamboo - Fast and Secure SSE against Key Compromise”)
- More-is-better attacks
- More practical attack assumptions: always less knowledge.
- Handle active frequent updates, and low frequency keywords.
- More practical countermeasures (TC, SEAL) – usually over-killing, e.g., padding [SEAL].

Core SSE attacks references

- 2012 Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation
- 2014 Search pattern leakage in searchable encryption: Attacks and new construction
- 2015 Leakage-Abuse Attacks Against Searchable Encryption
- 2016 The Shadow Nemesis: Inference Attacks on Efficiently Deployable, Efficiently Searchable Encryption
- 2016 All Your Queries Are Belong to Us: The Power of File-Injection Attacks on Searchable Encryption
- 2020 Revisiting Leakage Abuse Attacks
- 2021 Leakage-Hiding the Access Pattern is Not Enough: Exploiting Search Pattern Leakage in Searchable Encryption
- 2021 A Highly Accurate Query-Recovery Attack against Searchable Encryption using Non-Indexed Documents
- 2021 LEAP: Leakage-Abuse Attack on Efficiently Deployable, Efficiently Searchable Encryption with Partially Known Dataset
- 2022 VAL: Volume and Access Pattern Leakage-Abuse Attack with Leaked Documents
- 2023 High Recovery with Fewer Injections: Practical Binary Volumetric Injection Attacks against Dynamic Searchable Encryption

- 2013 An Efficient Attack on A Fuzzy Keyword Search Scheme over Encrypted Data
- 2017 No Such Thing as a Small Leak: Leakage-Abuse Attacks Against Symmetric Searchable Encryption
- 2017 Practical passive leakage-abuse attacks against symmetric searchable encryption
- 2017 Query Recovery Attacks on Searchable Encryption Based on Partial Knowledge
- 2018 Leakage Models and Inference Attacks on Searchable Encryption for Cyber-Physical_ Social Systems
- 2018 Practical Attacks on Relational Databases Protected via Searchable Encryption
- 2019 Inference Attacks on Fuzzy Searchable Encryption Schemes
- 2019 Passive Attacks Against Searchable Encryption
- 2020 Improved File-injection Attacks on Searchable Encryption Using Finite Set Theory

PARTNERS


UBITECH
ubiquitous solutions


MARTEL
innovate


Suite5
We Deliver Intelligence


netcompany
intrasoft


SPACE


United Technologies
Research Center


uni systems


BIBA
Bremer Institut für Produktion und Logistik


CITY OF
ATHENS


TECHNISCHE
UNIVERSITÄT
DARMSTADT


TU Delft


DTU


UNIVERSITY OF
SURREY


NVIDIA

THANKS

PROJECT-ASSURED.EU

@Project_Assured

ASSURED project is funded by the EU's Horizon2020 programme under Grant Agreement number 952697