Encrypted Search: Leakage Attacks

Seny Kamara
How do we Deal with Leakage?

• Our definitions allow us to prove that our schemes
  • achieve a certain leakage profile
  • but doesn’t tell us if a leakage profile is exploitable?
• We need more than proofs
The Methodology

- **Leakage analysis**: what is being leaked?
- **Proof**: prove that scheme leaks no more
- **Cryptanalysis**: can we exploit this leakage?
Leakage Attacks

- **Target**
  - *query recovery*: recovers information about query
  - *data recovery*: recovers information about data
- **Adversarial model**
  - *persistent*: needs EDS and tokens
  - *snapshot*: needs EDS
- **Auxiliary information**
  - *known sample*: needs sample from same distribution
  - *known data*: needs actual data
- **Passive vs. active**
  - *injection*: needs to inject data
Leakage Attacks

• Inference attacks \( \approx \) (passive) known-sample attacks
  • [Islam-Kuzu-Kantarcioglu12]*
    • persistent query-recovery vs. SSE with baseline leakage
  • [Naveed-K.-Wright15, …]
    • snapshot data-recovery vs. PPE-based encrypted databases
  • [Kellaris-Kollios-Nissim-O’Neill, …]
    • persistent query-recovery vs. encrypted range schemes
Leakage Attacks

• Leakage-abuse attacks ≈ (passive) known-data attacks
  • [Cash-Grubbs-Perry-Ristenpart15]
    • persistent query-recovery vs. SSE with baseline leakage
• Injection attacks ≈ (active) chosen-data attacks
  • [Cash-Grubbs-Perry-Ristenpart15]
    • persistent query-recovery vs. non-SSE-based solutions
  • [Zhang-Papamanthou-Katz16]
    • persistent query-recovery vs. SSE with baseline leakage
Typical Citations

• “For example, IKK demonstrated that by observing accesses to an encrypted email repository, an adversary can infer as much as 80% of the search queries”

• “It is known that access patterns, to even encrypted data, can leak sensitive information such as encryption keys [IKK]”

• “A recent line of attacks […, Count,…] has demonstrated that such access pattern leakage can be used to recover significant information about data in encrypted indices. For example, some attacks can recover all search queries [Count,…] …”
IKK Attack

[Islam-Kantarcioğlu-Kuzu12]

• Published as an inference attack
  • persistent *known-sample* query-recovery attack
  • exploits co-occurrence pattern + knowledge of 5% of queries
    • co-occur: times each pair of documents occur together
• Highly cited but significant limitations
  • experiments only for 2500 out of 77K+ keywords
  • auxiliary and test data were not independent
• [CGPR15] re-ran IKK on independent test data
  • it achieved 0% recovery
IKK as a Known-Data Attack

[Islam-Kantargioglu-Kuzu12, Cash-Grubbs-Perry-Ristenpart15]

• What if we just give IKK the client data; does it work then?

• Notation
  • \( \delta \): fraction of adversarially-known data
  • \( \varphi \): fraction of adversarially-known queries

• [CGPR15] experiments for IKK attack
  • \( \delta = 70\% + \varphi = 5\% \) recovers 5\% of queries
  • \( \delta = 95\% + \varphi = 5\% \) recovers 20\% of queries
The Count Attack
[Cash-Grubbs-Perry-Ristenpart15]

• Known-data attack (i.e., “leakage-abuse attack”)
  • Count v.1 [2015] and Count v.2 [2019]
  • exploit co-occurrence pattern + response length
• Count v.1
  • \( \delta = 80\% + \varphi = 5\% \) recovers 40% of queries
  • \( \delta = 75\% + \varphi = 5\% \) recovers 0% of queries
• Count v.2
  • \( \delta = 75\% \) recovers 40% of queries
Revisiting Leakage-Abuse Attacks

• High known-data rates ($\delta \geq 75\%$)
  • how can an adversary learn 75% of client data?
  • recall that when outsourcing, client erases plaintext
    • if client needs to outsource public data it should use PIR
• Known queries ($\varphi \geq 5\%$)
Revisiting Leakage-Abuse Attacks

- Low-vs. high selectivity keywords
  - Experiments all run on high-selectivity keywords
  - We re-ran on low-selectivity keywords and attacks failed
- Both exploit co-occurrence pattern
  - relatively easy to hide (see OPQ [Blackstone-K.-Moataz19])
Revisiting Leakage-Abuse Attacks

- Should we discount the IKK and Count attacks?
  - No! they are interesting, just not necessarily practical
- Theoretical attacks (e.g., Count, IKK)
  - rely on strong assumptions, e.g., $\delta > 20\%$ or $\varphi > 20\%$
- Practical attacks (e.g., [Naveed-K.-Wright15] vs. PPE-based)
  - weak adversarial model
  - mild assumptions (*real-world* auxiliary input)
Q: can we do better than IKK & Count?
## New Known-Data Attacks

[Blackstone-K.-Moataz19]

<table>
<thead>
<tr>
<th>Attack</th>
<th>Type</th>
<th>Pattern</th>
<th>Known Queries</th>
<th>$\delta$ for HS</th>
<th>$\delta$ for PLS</th>
<th>$\delta$ for LS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IKK</td>
<td>known-data</td>
<td>co</td>
<td>Yes</td>
<td>$\geq 95%$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Count</td>
<td>known-data</td>
<td>rlen</td>
<td>Yes/No</td>
<td>$\geq 80%$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Injection</td>
<td>injection</td>
<td>rid</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SubgraphID</td>
<td>known-data</td>
<td>rid</td>
<td>No</td>
<td>$\geq 5%$</td>
<td>$\geq 50%$</td>
<td>$\geq 60%$</td>
</tr>
<tr>
<td>SubgraphVL</td>
<td>known-data</td>
<td>vol</td>
<td>No</td>
<td>$\geq 5%$</td>
<td>$\geq 50%$</td>
<td>$\delta=1$ recovers $&lt;10%$</td>
</tr>
<tr>
<td>VolAn</td>
<td>known-data</td>
<td>tvol</td>
<td>No</td>
<td>$\geq 85%$</td>
<td>$\geq 85%$</td>
<td>$\delta=1$ recovers $&lt;10%$</td>
</tr>
<tr>
<td>SelVolAn</td>
<td>known-data</td>
<td>tvol, rlen</td>
<td>No</td>
<td>$\geq 80%$</td>
<td>$\geq 85%$</td>
<td>$\delta=1$ recovers $&lt;10%$</td>
</tr>
<tr>
<td>Decoding</td>
<td>injection</td>
<td>tvol</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Apply to ORAM

$\delta$ needed for RR $\geq 20\%$

HS $\geq 13$

PLS = 10-13

LS = 1-2
The Subgraph\textsuperscript{VL} Attack
[Blackstone-K.-Moataz19]

- Let $K \subseteq D$ be set of known documents
  - $K = (K_2, K_4)$ and $D = (D_1, \ldots, D_4)$

Known Graph

- $\text{vol}(K_2)$
- $\text{vol}(K_4)$

Observed Graph

- $\text{vol}(D_1)$
- $\text{vol}(D_2)$
- $\text{vol}(D_3)$
- $\text{vol}(D_4)$

$w_1$, $w_4$, $w_5$, $q_1$, $q_2$, $q_3$, $q_4$, $q_5$
The Subgraph\textsuperscript{VL} Attack
[Blackstone-K.-Moataz19]

- We need to match $q_i$ to some $w_j$
- Observations: if $q_i = w_j$ then
  - $N(w_j) \subseteq N(q_i)$ and $|N(w_j)| \approx \delta N(q_i)$
  - $w_j$ cannot be a match for $q_z$ for $z \neq i$
The Subgraph\textsuperscript{VL} Attack

[Blackstone-K.-Moataz19]

• Each query \( q \) starts with a candidate set \( C_q = \mathbb{W} \)
  • remove all words that have been matched to other queries
  • remove all words s.t. either \( N(w_j) \not\subseteq N(q_i) \) or \( \#N(w_j) \neq \delta N(q_i) \)
  • if a single word is left that’s the match
    • remove it from other queries’ candidate sets
Revisiting Leakage-Abuse Attacks
[Blackstone-K.-Moataz19]

• ORAM-based search is also vulnerable to known-data attacks
• Subgraph attacks are practical for high-selectivity queries
  • can exploit rid or vol
  • need only $\delta \geq 5$
• Countermeasures
  • for $\delta < 80\%$ use OPQ [Blackstone-K.-Moataz19]
  • for $\delta \geq 80\%$ use PBS [K.-Moataz-Ohrimenko18]
  • or use VLH or AVLH [K-Moataz19]
File Injection Attacks
[Zhang-Katz-Papamanthou16]

- Adversary tricks client into adding files
- For $i = 1$ to $\log(\#\mathbb{W})$
  - inject document $D_i = \{\text{all keywords with } i^{\text{th}} \text{ bit equal to } 1\}$
- Observation
  - if $D_i$ is returned then adversary knows $i^{\text{th}}$ bit of keyword is 1
  - otherwise $i^{\text{th}}$ bit of keyword is 0
- When client makes a query,
  - if $D_4, D_8, D_{10}$ are returned then $w = 0001000101$
File Injection Attacks

[Zhang-Katz-Papamanthou16]

• Requires injecting documents of size
  • $2^{\log(#W)} - 1 = \#W/2$ keywords

• What if client refuses to add documents of size $\geq \#W/2$?
  • just target a smaller set of queries $\mathcal{Q}$ s.t. $\#\mathcal{Q} = \#W-2$

• Hierarchical injection attack
  • more sophisticated attack recovers sets larger than $\#W/2$…
    • …even when client uses threshold
Attacks on Encrypted Range Search

- [Kellaris-Kollios-Nissim-O’Neill16]
  - recovers values by exploiting response id + volume
  - requires $O(N^4 \cdot \log N)$ queries
  - assumes uniform queries
- [Grubbs-Lacharite-Minaud-Paterson19]
  - recovers $\varepsilon N$-approximation by exploiting response identity
  - requires $O(\varepsilon^{-2} \log \varepsilon^{-1})$ queries
- [Grubbs-Lacharite-Minaud-Paterson19]
  - recovers $\varepsilon N$-approximate order by exploiting response identity
  - requires $O(\varepsilon^{-1} \log \varepsilon^{-1})$ queries