Revisiting Leakage Abuse Attacks

Laura Blackstone
BROWN

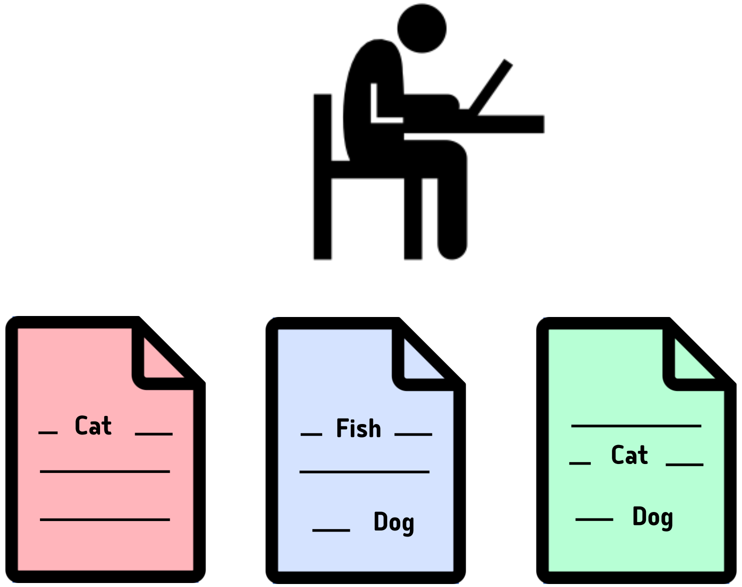
Seny Kamara
BROWN

Tarik Moataz
AROKI SYSTEMS

ENCRYPTED SYSTEMS LAB

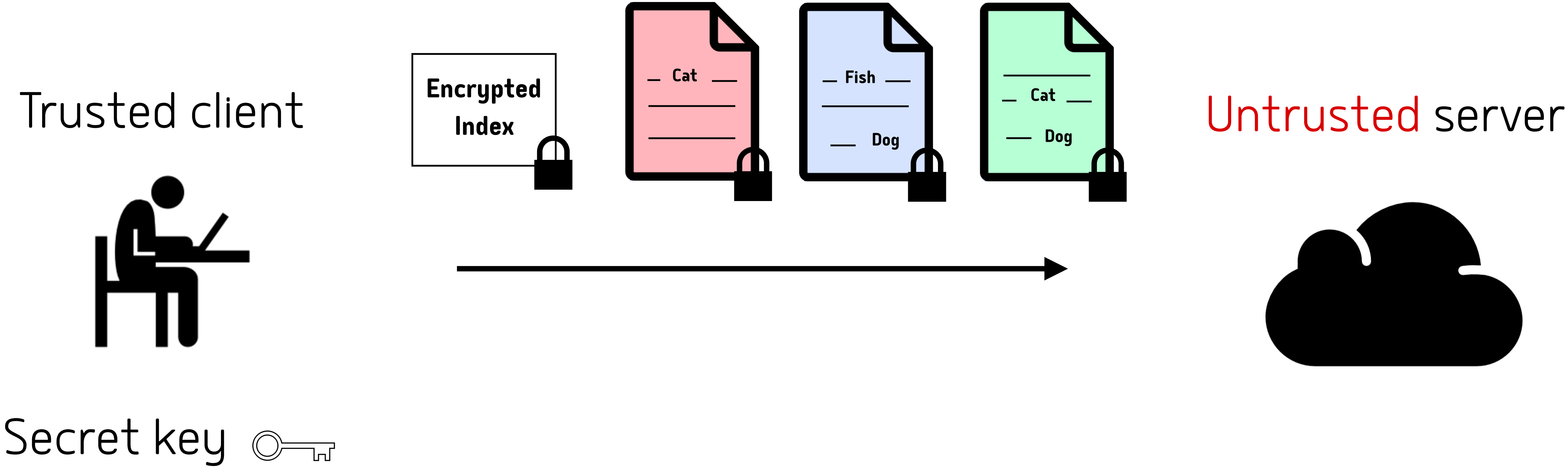
Encrypted Search

Trusted client

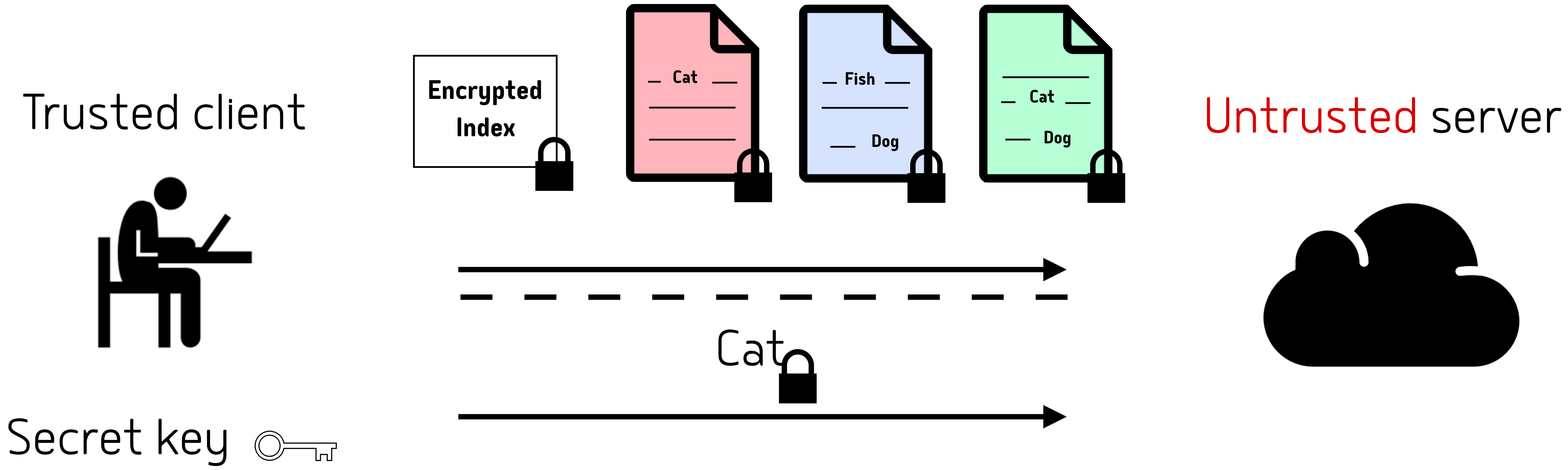


Untrusted server

Encrypted Search

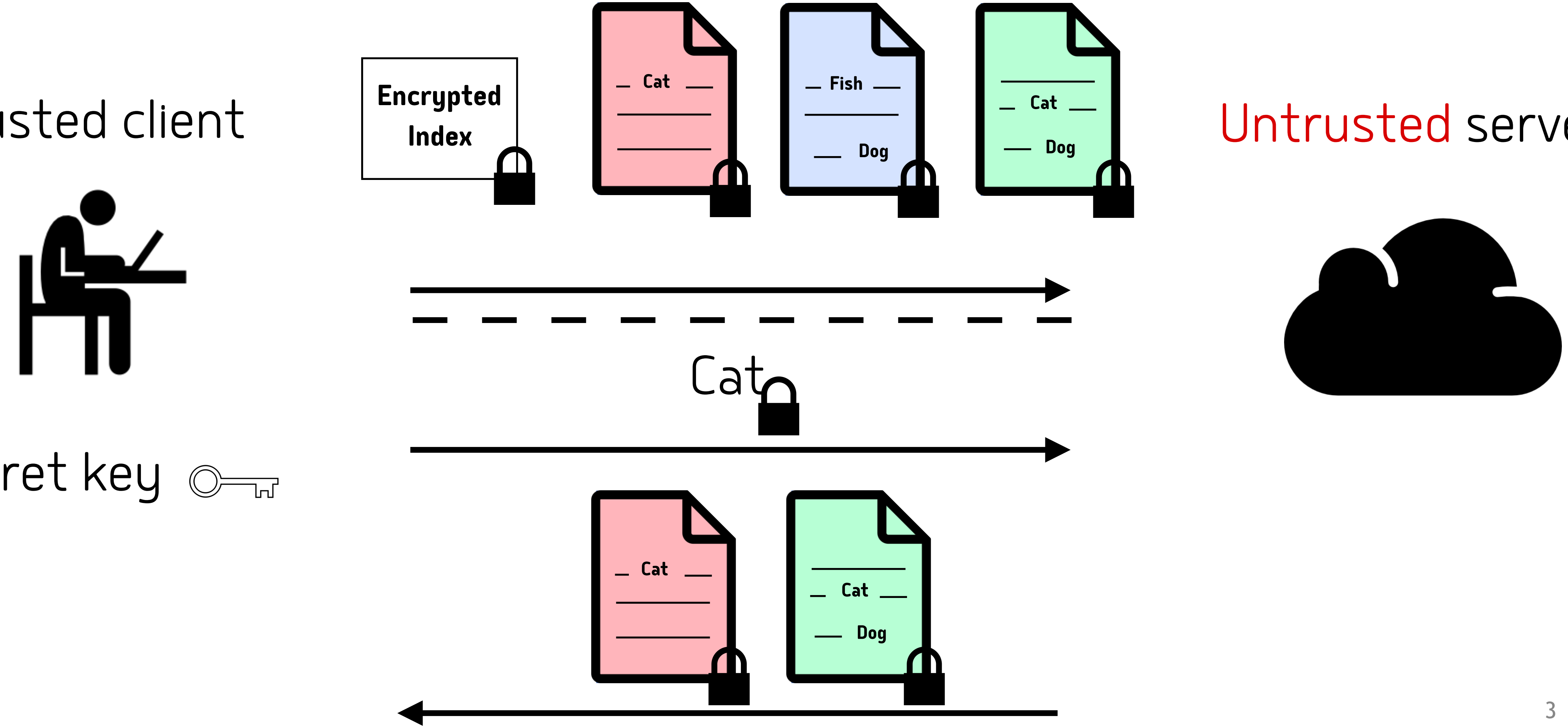


Encrypted Search



Encrypted Search

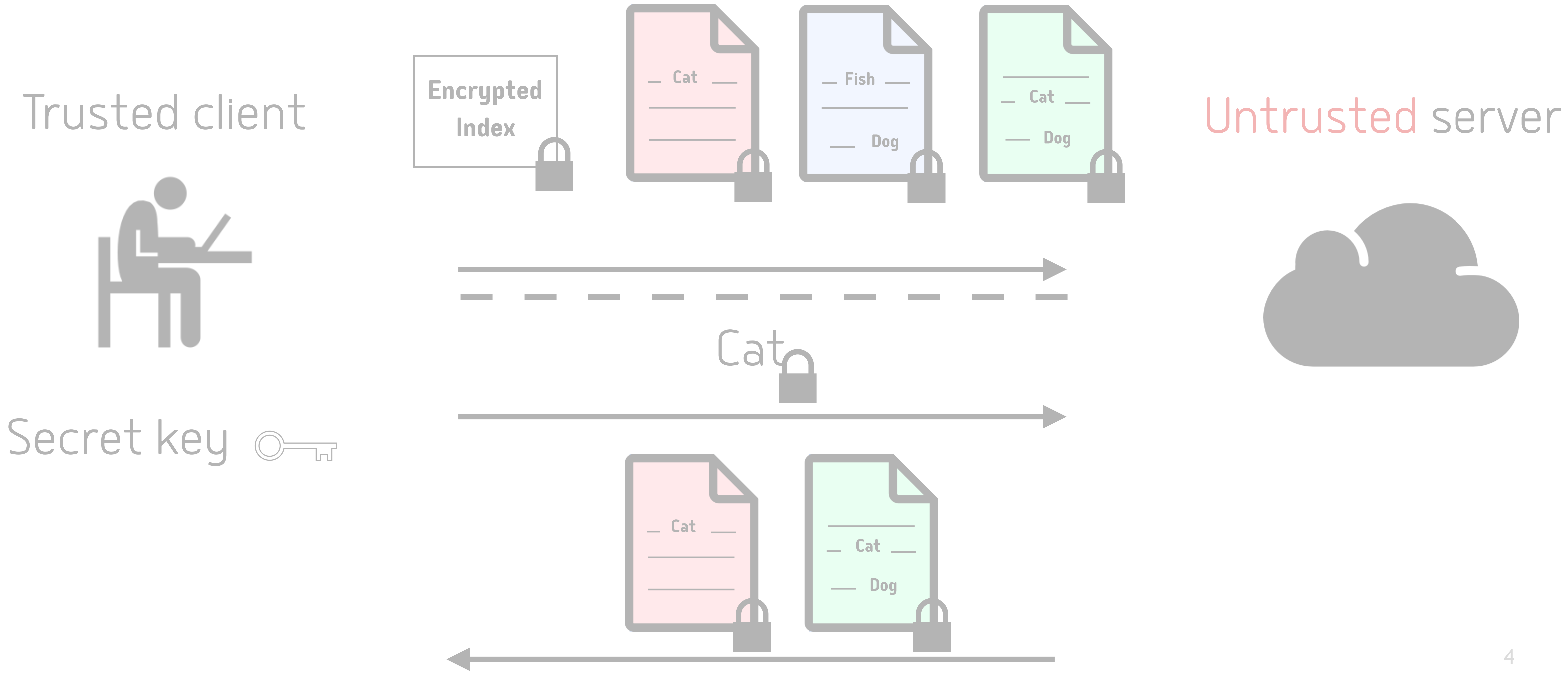
Trusted client



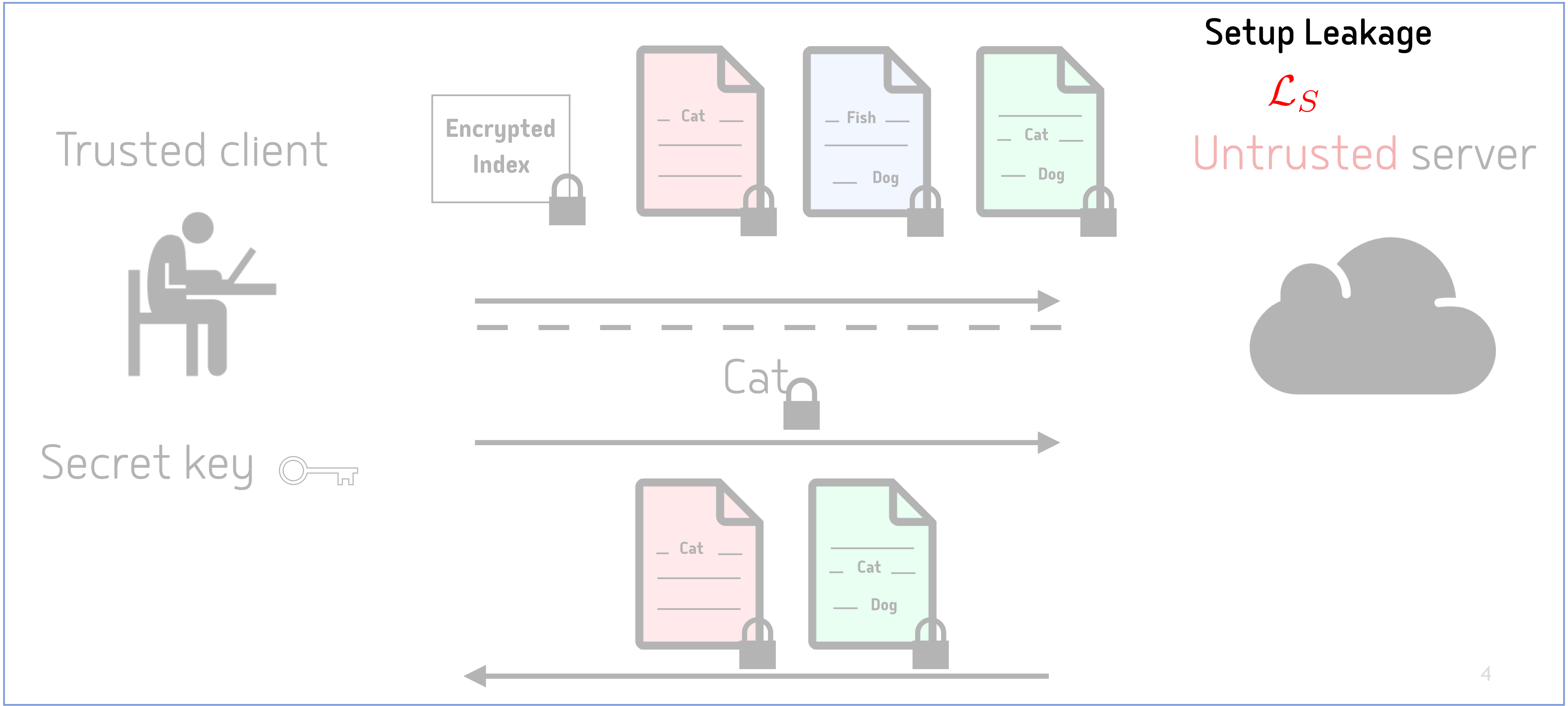
Untrusted server

Secret key

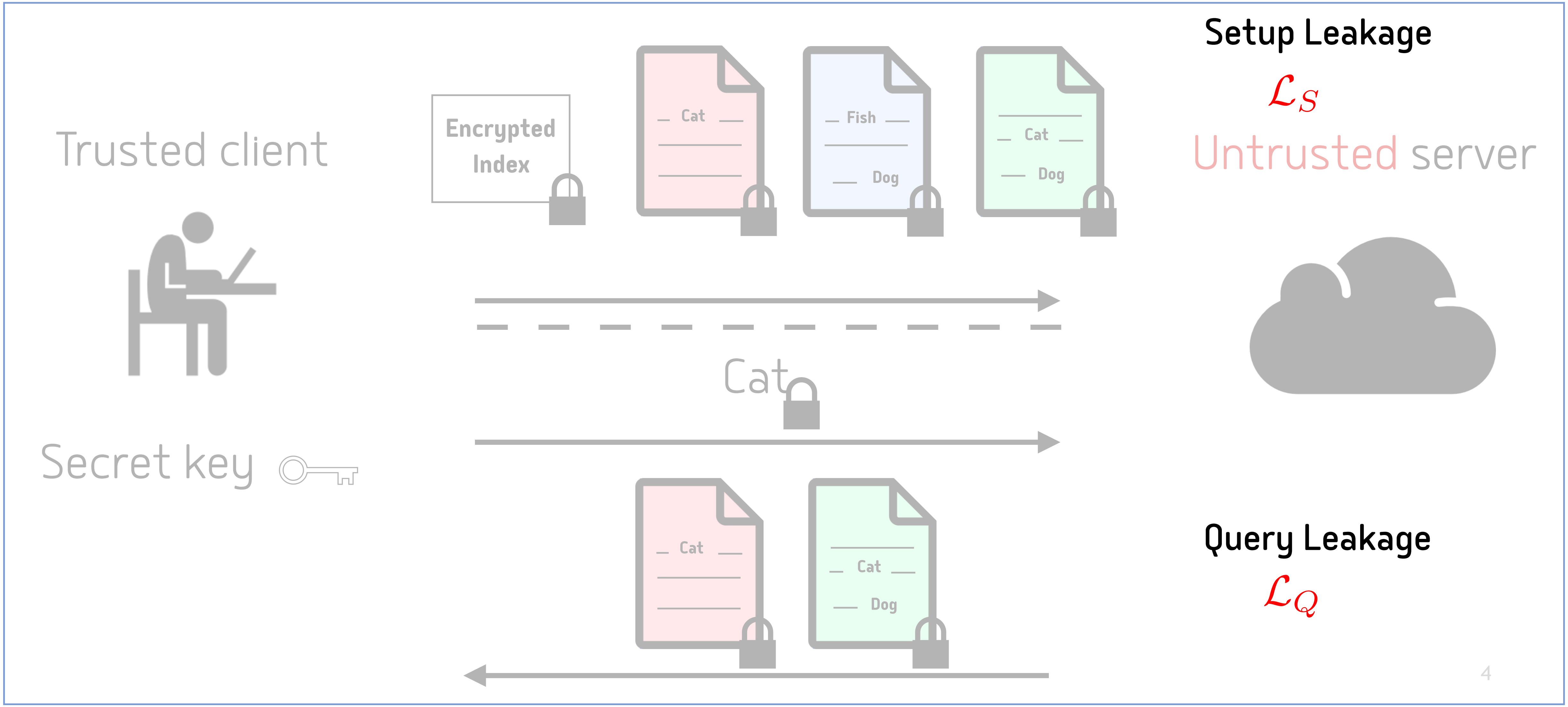
Encrypted Search



Encrypted Search



Encrypted Search



Query Leakage Terminology

- Query equality pattern (qeq)
  - If and when the search is the same (search pattern)
- Response identity pattern (rid)
  - The file identifiers matching the query (access pattern)
- Co-occurrence pattern (co-occ)
  - The number of files shared by any two queries
- Response length pattern (rlen)
  - The number of files matching a query
- Volume pattern (vol) / Total volume pattern (tvol)
  - The number of bits of each file / the sum of file sizes in bits

Q: do we leak all of these patterns “at once”?

Encrypted Search

Primitives

Property-Preserving Encryption (PPE)

Functional Encryption

Structured Encryption (STE)

Fully-Homomorphic Encryption (FHE)

Oblivious RAM (ORAM)

Encrypted Search

Primitives

- Property-Preserving Encryption (PPE)
- Functional Encryption
- Structured Encryption (STE)
- Fully-Homomorphic Encryption (FHE)
- Oblivious RAM (ORAM)

Encrypted Search

STE- & ORAM- based schemes

co-occ

qeq

vol

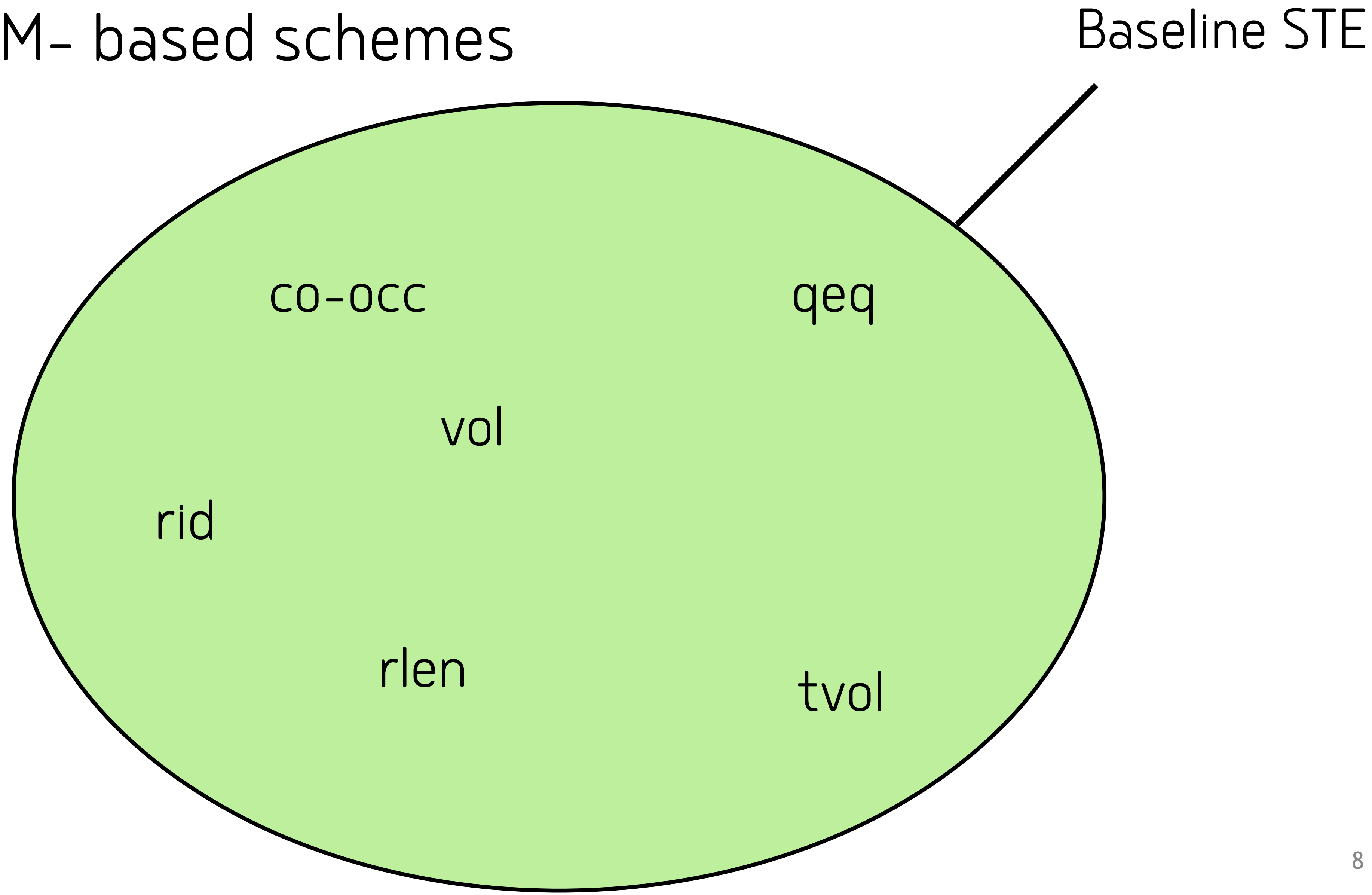
rid

rlen

tvol

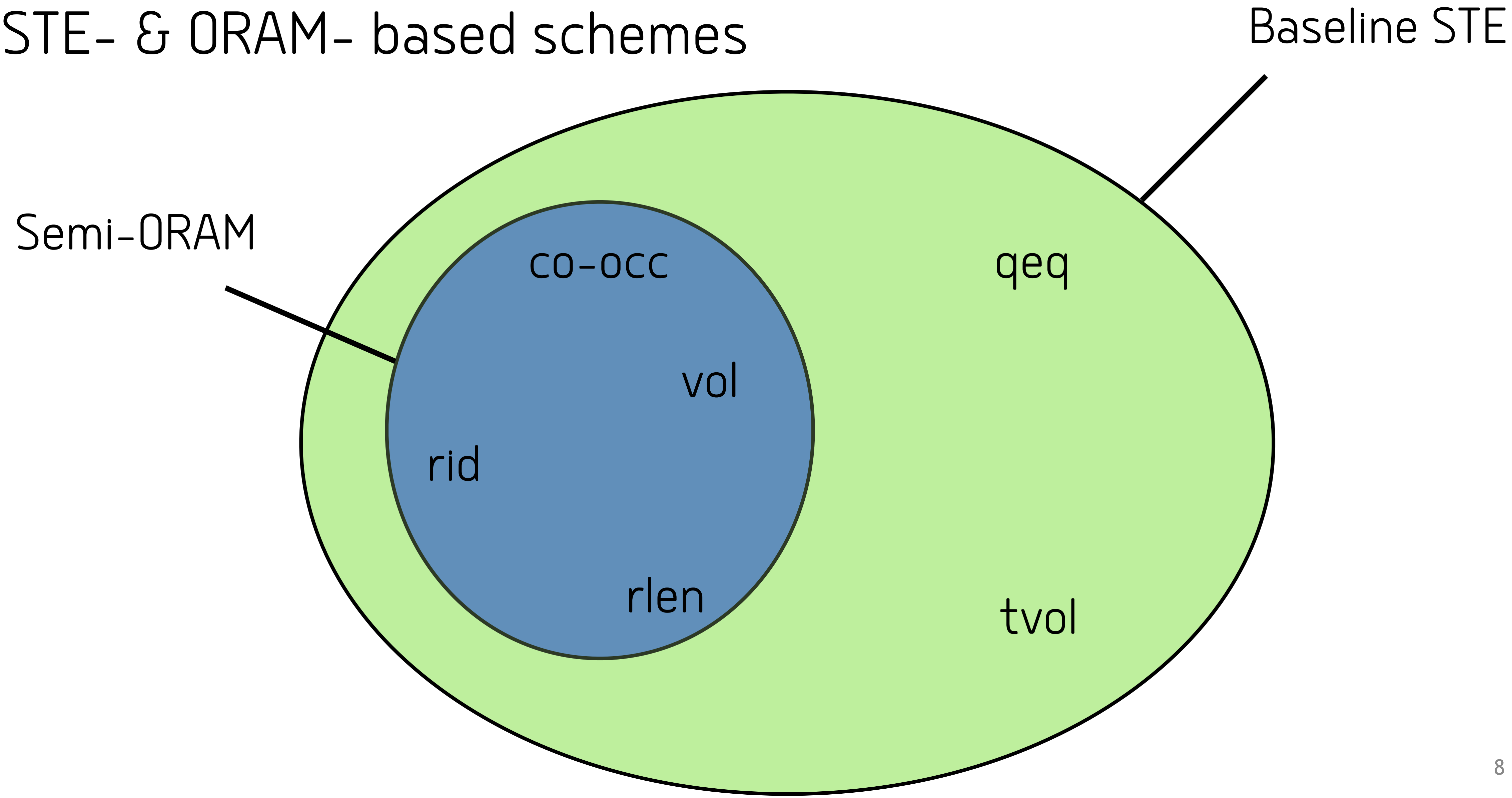
Encrypted Search

STE- & ORAM- based schemes



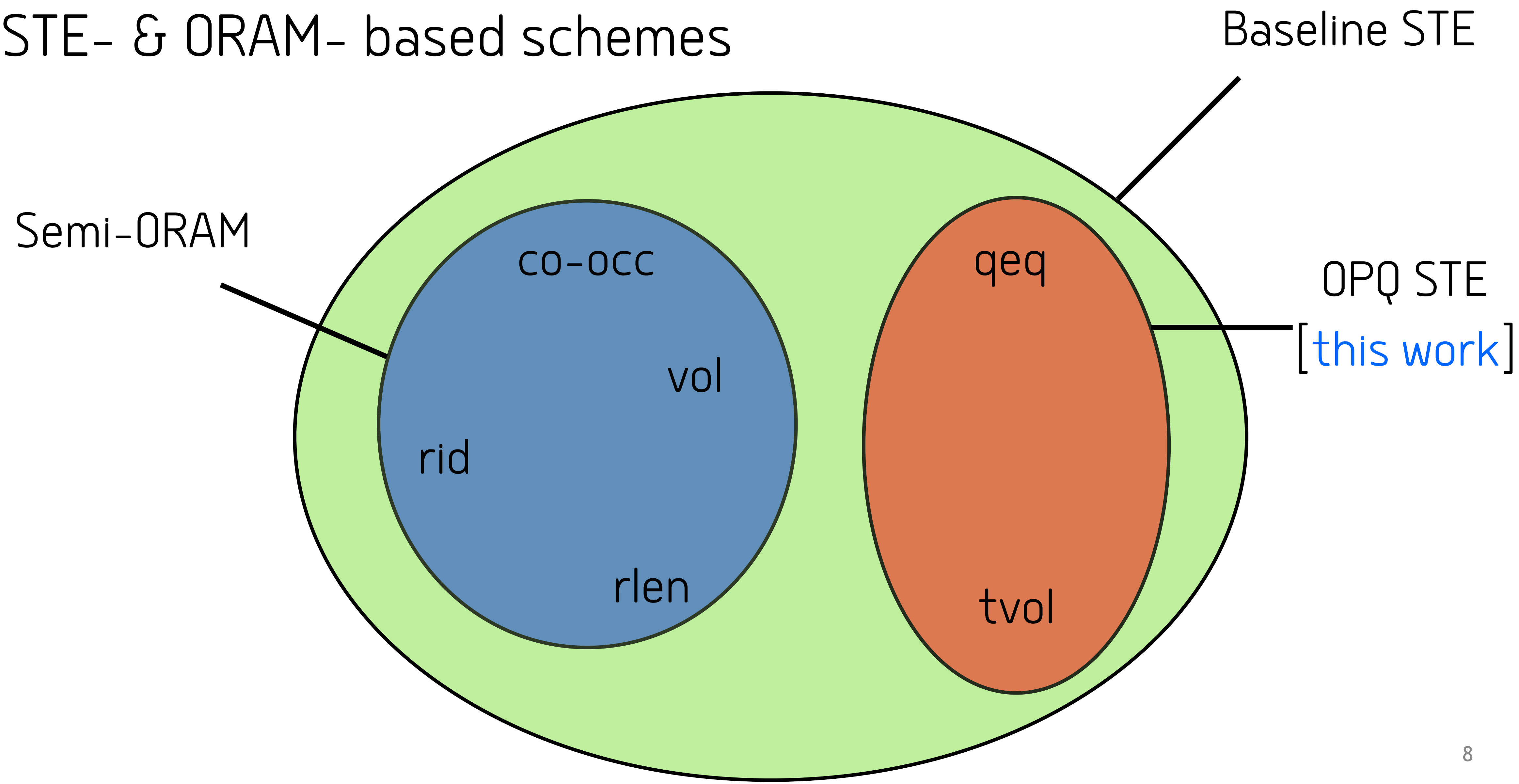
Encrypted Search

STE- & ORAM- based schemes



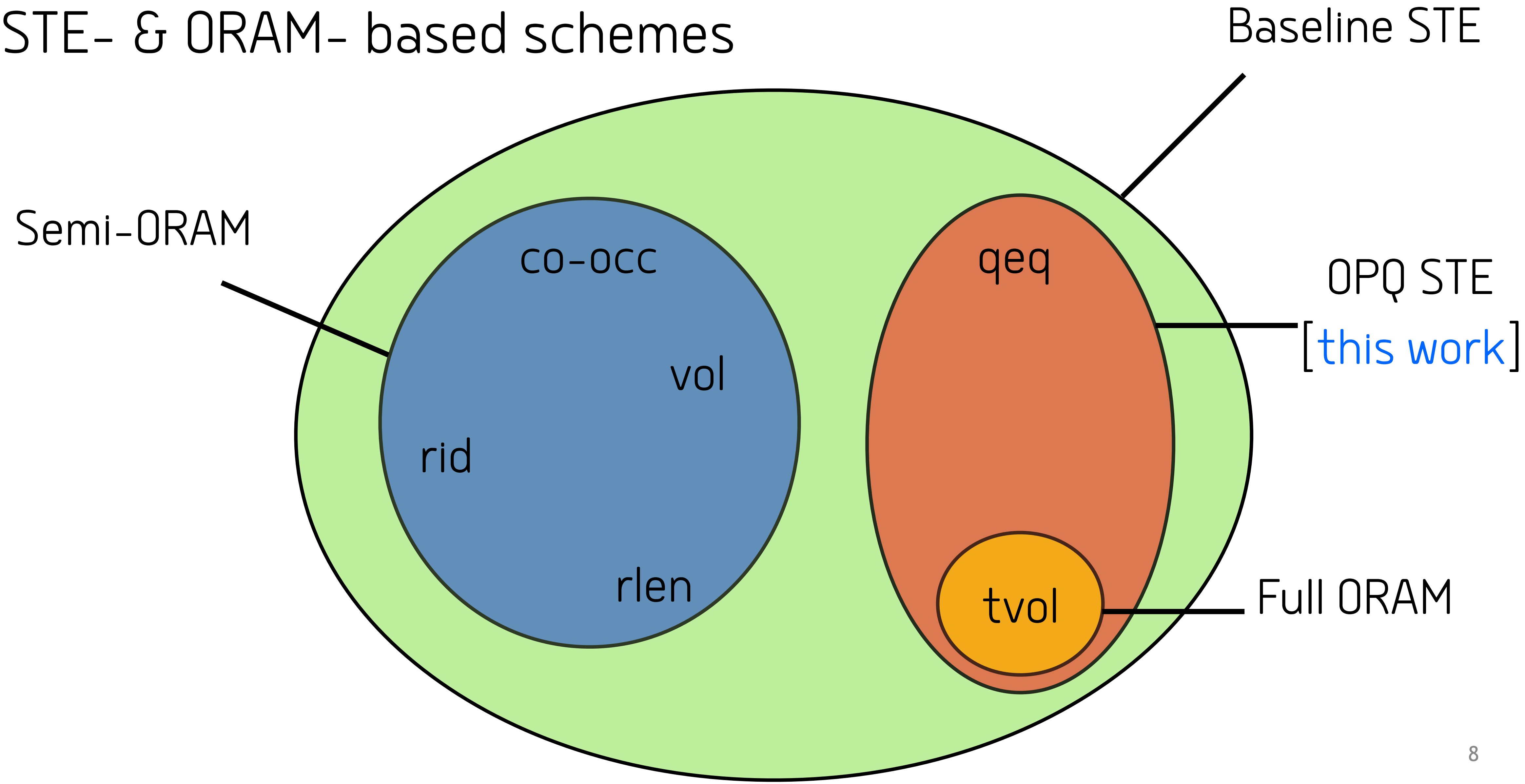
Encrypted Search

STE- & ORAM- based schemes



Encrypted Search

STE- & ORAM- based schemes



Q: can we use the disclosed leakage to recover user's data?

Leakage Attacks

Input

One or more leakage pattern

→ Leakage Attack →

Output

User's query or data recovery

Assumptions

- Type of adversary
- Type of auxiliary data
- Type of actions
- ...

Leakage Attacks

Assumptions

Leakage Attacks

Assumptions

- **Adversarial model**
  - persistent: needs encrypted index, documents and queries
  - snapshot: needs encrypted index and documents

Leakage Attacks

Assumptions

- **Adversarial model**
  - persistent: needs encrypted index, documents and queries
  - snapshot: needs encrypted index and documents

- **Auxiliary information**
  - known sample: needs sample from same distribution
  - known data: needs actual data or/and user queries
    - δ: fraction of adversarially-known data

Leakage Attacks

Assumptions

- **Adversarial model**
  - persistent: needs encrypted index, documents and queries
  - snapshot: needs encrypted index and documents

- **Auxiliary information**
  - known sample: needs sample from same distribution
  - known data: needs actual data or/and user queries
    - δ: fraction of adversarially-known data

- **Passive vs. active**
  - injection (chosen-data): needs to inject data

Leakage Attacks

IKK Attack [Islam-Kuzu-Kantarcioglu12]

Leakage Attacks

IKK Attack [Islam-Kuzu-Kantarcioglu12]

Input

co-occ

IKK Attack

Output

Query recovery

Assumptions

- Persistent adversary
- Passive
- Known sample*
- Known queries

Leakage Attacks

IKK Attack [Islam-Kuzu-Kantarcioglu12]

Input

co-occ

IKK Attack

Output

Query recovery

Assumptions

- Persistent adversary
- Passive
- Known sample*
- Known queries

Vulnerable schemes

- Baseline STE
- Semi-ORAM

Leakage Attacks

Count Attack [Cash-Grubbs-Perry-Ristenpart15]

Leakage Attacks

Count Attack [Cash-Grubbs-Perry-Ristenpart15]

Input

co-occ + rlen

→ **Count Attack** →

Output

Query recovery

Assumptions

- Persistent adversary
- Passive
- Known data

Leakage Attacks

Count Attack [Cash-Grubbs-Perry-Ristenpart15]

Input

co-occ + rlen

→ **Count Attack** →

Output

Query recovery

Assumptions

- Persistent adversary
- Passive
- Known data

Vulnerable schemes

- Baseline STE
- Semi-ORAM

Impact of IKK & Count

- “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”

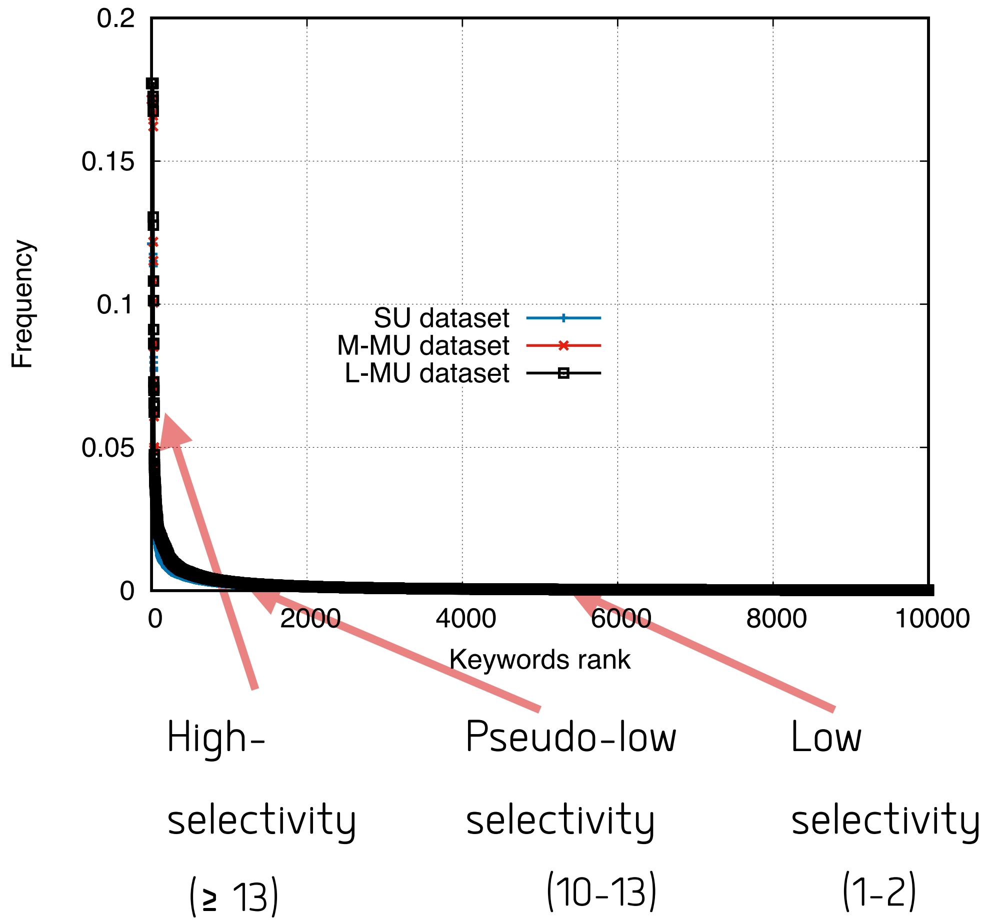
- “It is known that access patterns, to even encrypted data, can leak sensitive information such as encryption keys [IKK]”

- “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,…] …”

A closer look at IKK & Count attacks

Non-trivial limitations

- High known-data rates
  - Count v1 requires more than **80%** and **5%** of the queries
  - IKK requires more than **95%** and **5%** of the queries
  - Count v2 requires more than **60%**
  - Practical vs. Theoretical?
- Low-vs. high selectivity keywords
  - Experiments all run on high-selectivity keywords
    - Keywords that are frequent in the user's data
  - Re-ran on low-selectivity keywords and failed
- Both exploit co-occurrence
  - relatively easy to hide (using OPQ SSE)



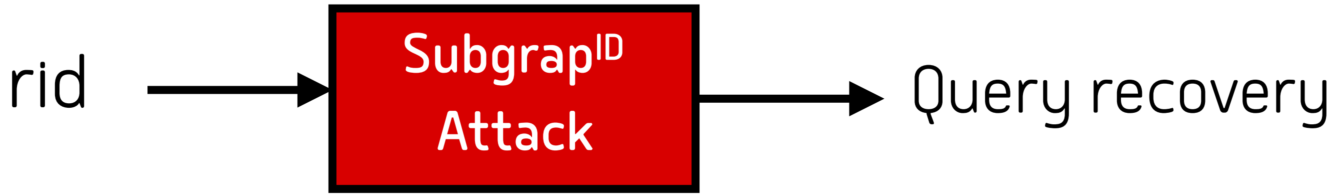
Q: can we de better than IKK & Count?

Summary of our Attacks

Known-Data attacks

Summary of our Attacks

Known-Data attacks



Summary of our Attacks

Known-Data attacks

Vulnerable schemes

rid → **Subgrap^{ID} Attack** → Query recovery

- Baseline STE
- Semi-ORAM

Summary of our Attacks

Known-Data attacks

Vulnerable schemes

rid → Subgrap^{ID} Attack → Query recovery

- Baseline STE
- Semi-ORAM

vol → Subgraph^{VL} Attack → Query recovery

- Baseline STE
- Semi-ORAM

Summary of our Attacks

Known-Data attacks

Vulnerable schemes

rid → **Subgrap^{ID} Attack** → Query recovery

- Baseline STE
- Semi-ORAM

vol → **Subgraph^{VL} Attack** → Query recovery

- Baseline STE
- Semi-ORAM

tvol → **VolAn & SelVolAn Attacks** → Query recovery

- Baseline STE
- Semi-ORAM
- OPQ STE
- Full ORAM

Summary of our Attacks

Injection attacks

Vulnerable schemes

tvol → **Decoding & Binary attacks** → Query recovery

- Baseline STE
- Semi-ORAM
- OPQ STE
- Full ORAM

First injection attack was by [Zhang-Katz-Papamanthou16] and works against Baseline STE and Semi-ORAM

The Subgraph^{VL} Attack

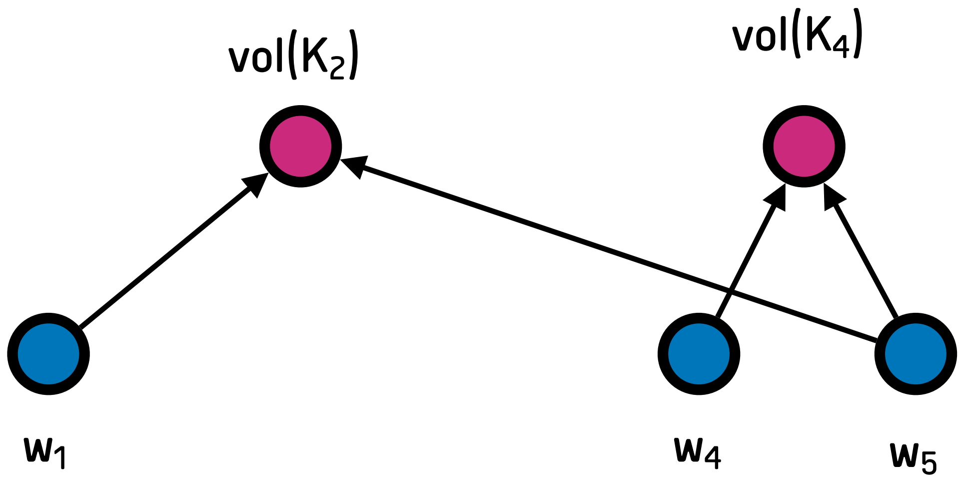
The Subgraph^{VL} Attack

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

The Subgraph^{VL} Attack

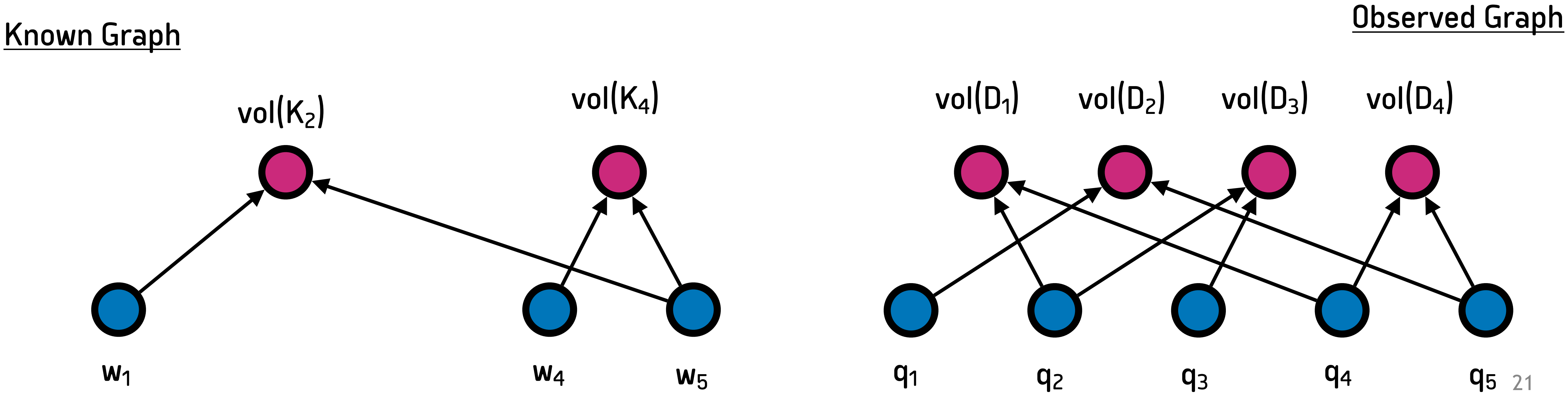
- Let $\mathbf{K} \subseteq \mathbf{D}$ be set of known documents
  - $\mathbf{K} = (K_2, K_4)$ and $\mathbf{D} = (D_1, \ldots, D_4)$

Known Graph



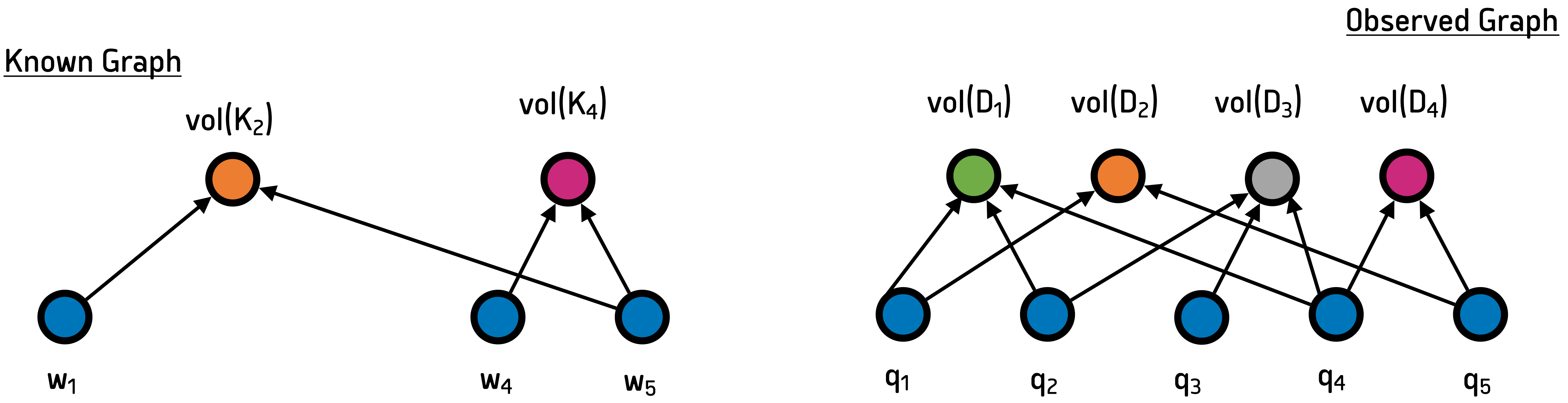
The Subgraph^{VL} Attack

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



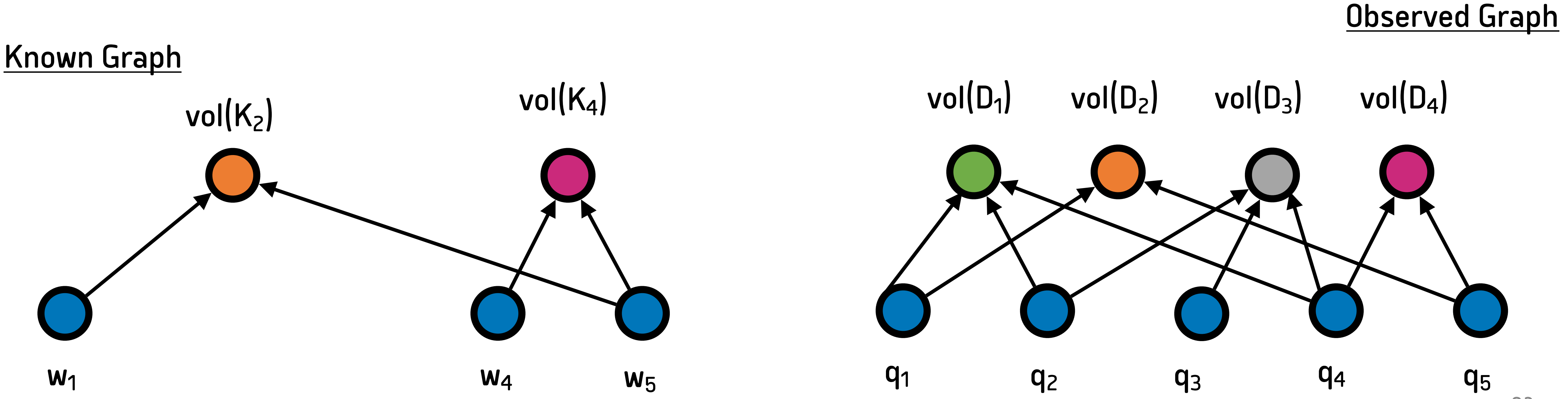
The Subgraph^{VL} Attack

- We need to match $\mathbf{q_i}$ to some $\mathbf{w_j}$
- The volumes are the ground of truth



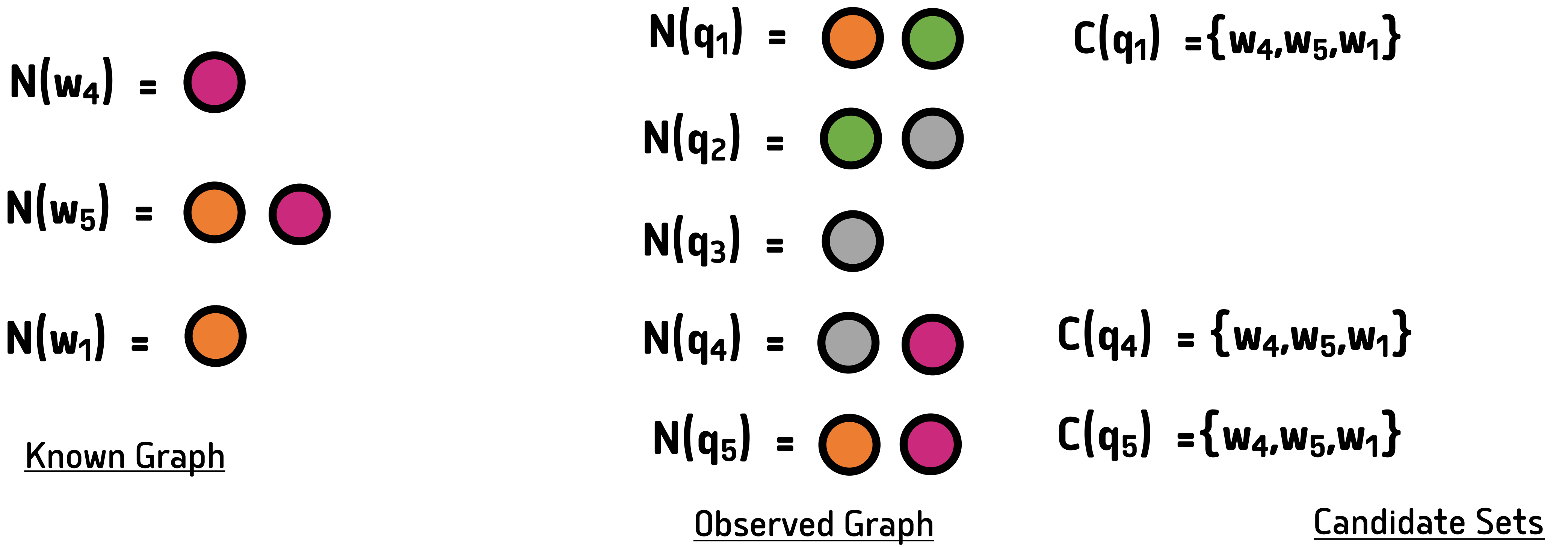
The Subgraph^{VL} Attack

- **Observations**: if $\mathbf{q_i = w_j}$ then
  - $N(w_j) \subseteq N(q_i)$ and $\#N(w_j) \approx \delta \cdot \#N(q_i)$



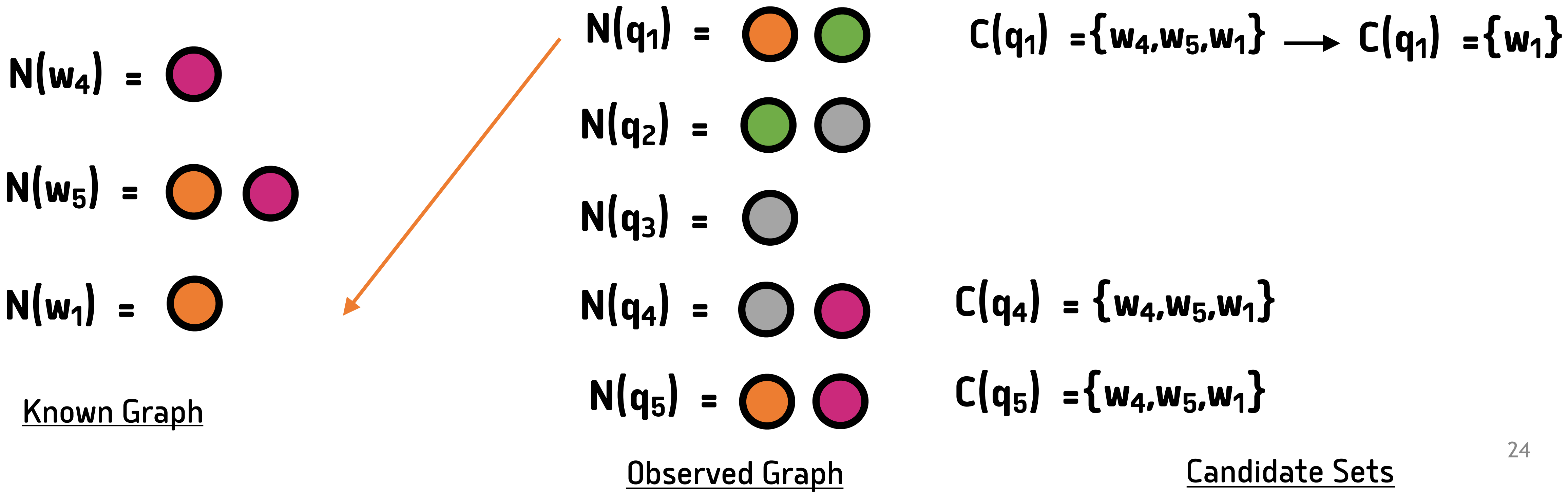
The Subgraph^{VL} Attack

- Each query q starts with a candidate set $C_q = \mathbb{W}$
  - remove all words s.t. either $N(w_j) \not\subseteq N(q_i)$ or $\#N(w_j) \not\approx \delta \cdot N(q_i)$



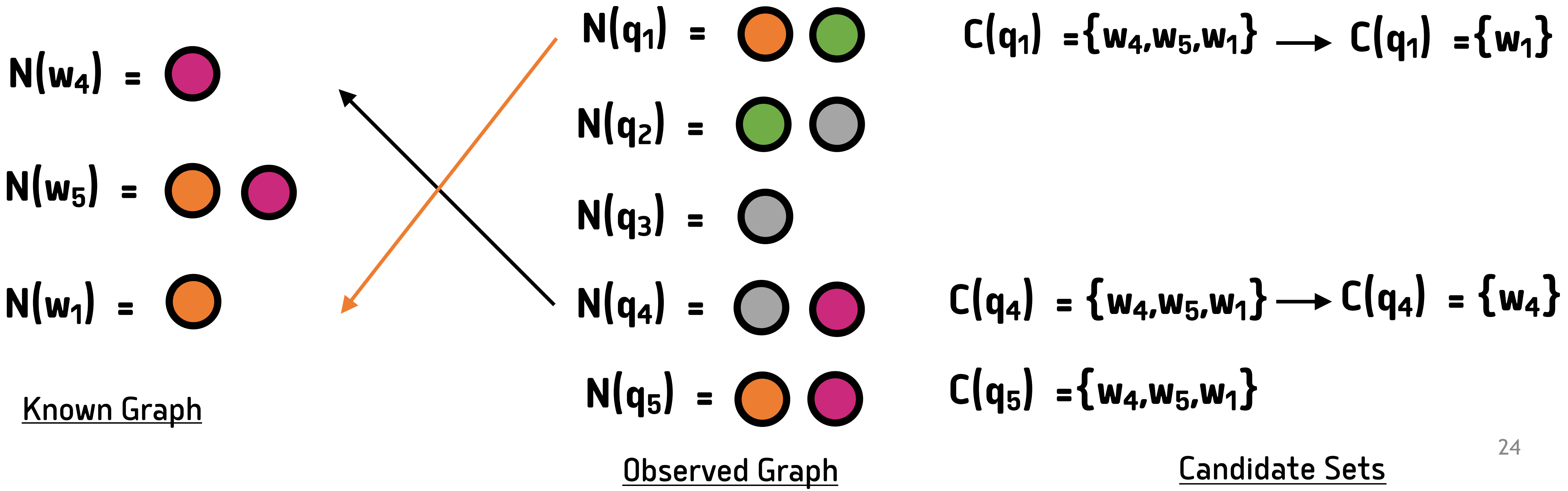
The Subgraph^{VL} Attack

- Each query q starts with a candidate set $C_q = \mathbb{W}$
  - remove all words s.t. either $N(w_j) \not\subseteq N(q_i)$ or $\#N(w_j) \not\approx \delta \cdot N(q_i)$



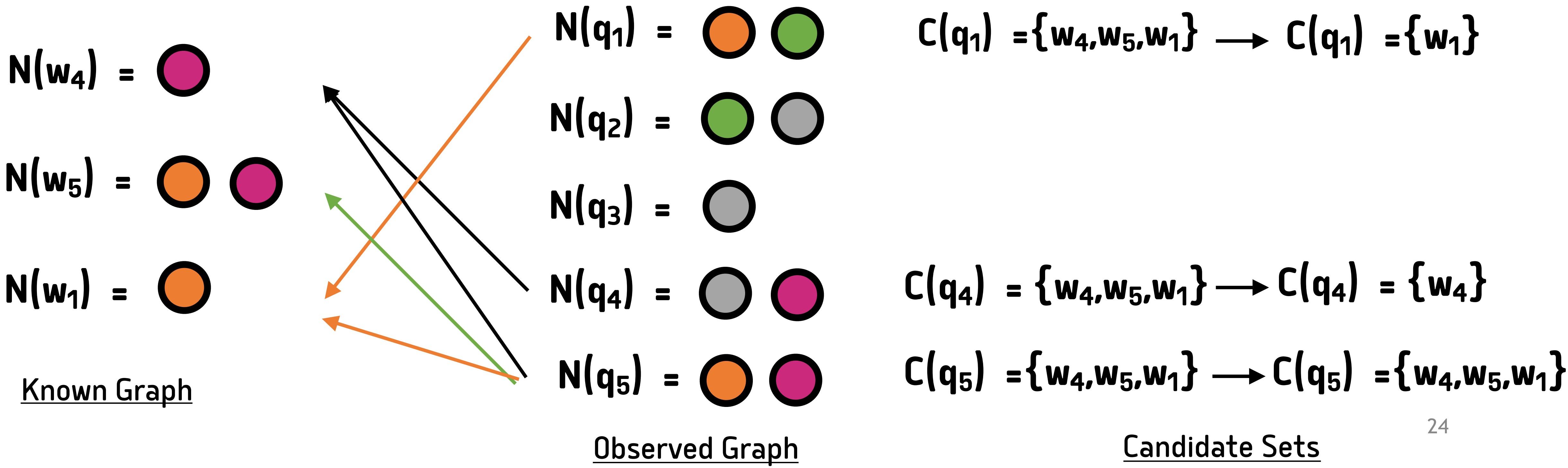
The Subgraph^{VL} Attack

- Each query q starts with a candidate set $C_q = \mathbb{W}$
  - remove all words s.t. either $N(w_j) \not\subseteq N(q_i)$ or $\#N(w_j) \not\approx \delta \cdot N(q_i)$



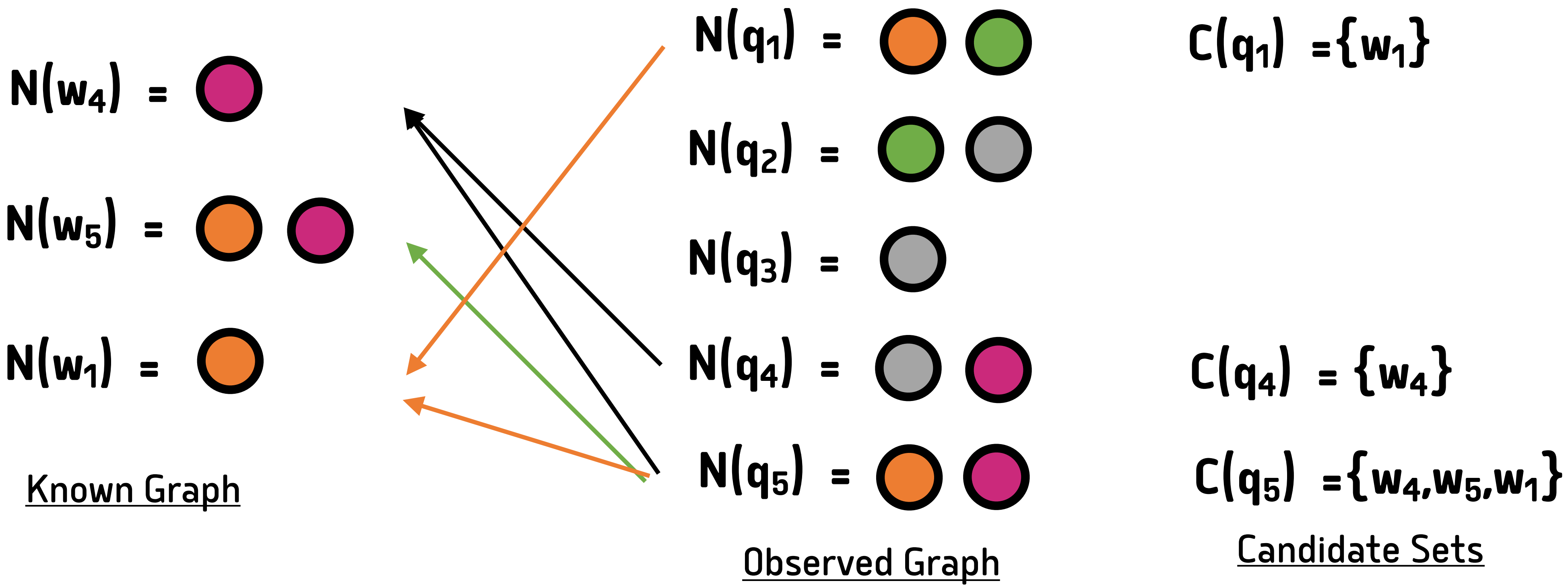
The Subgraph^{VL} Attack

- Each query q starts with a candidate set $C_q = \mathbb{W}$
  - remove all words s.t. either $N(w_j) \not\subseteq N(q_i)$ or $\#N(w_j) \not\approx \delta \cdot N(q_i)$



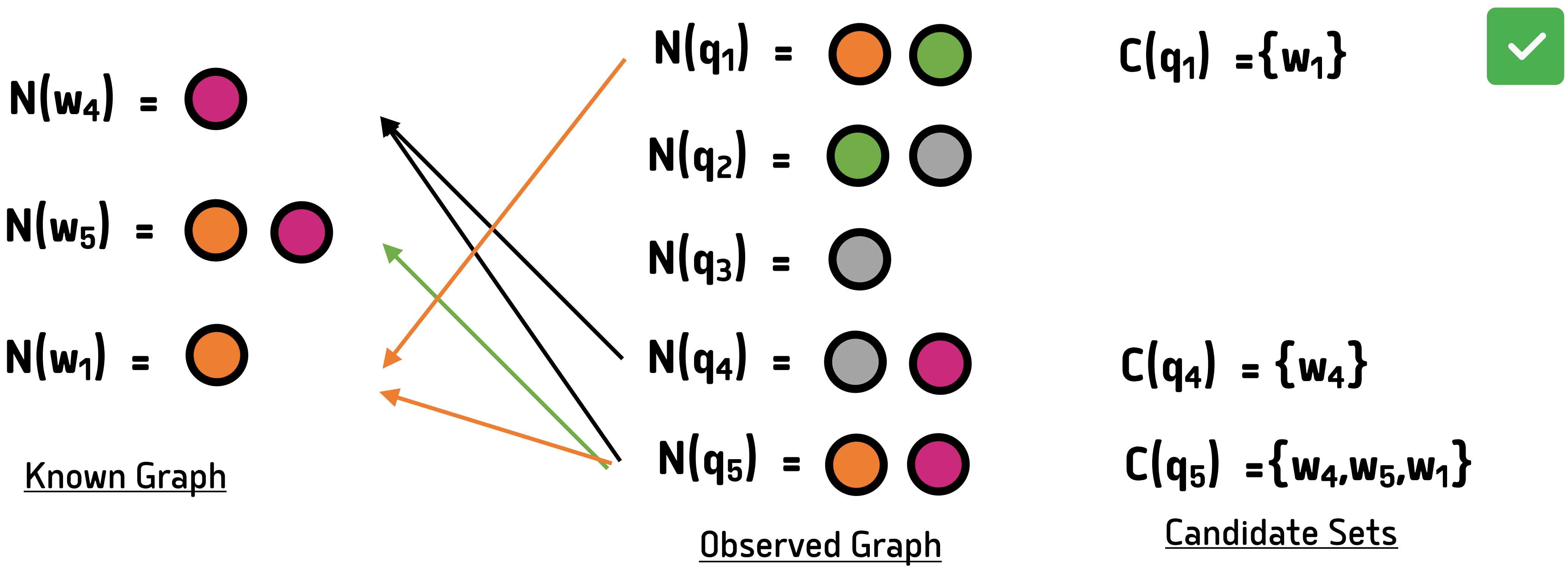
The Subgraph^{VL} Attack

- If a single word is left that's the match
- Remove it from other queries' candidate sets



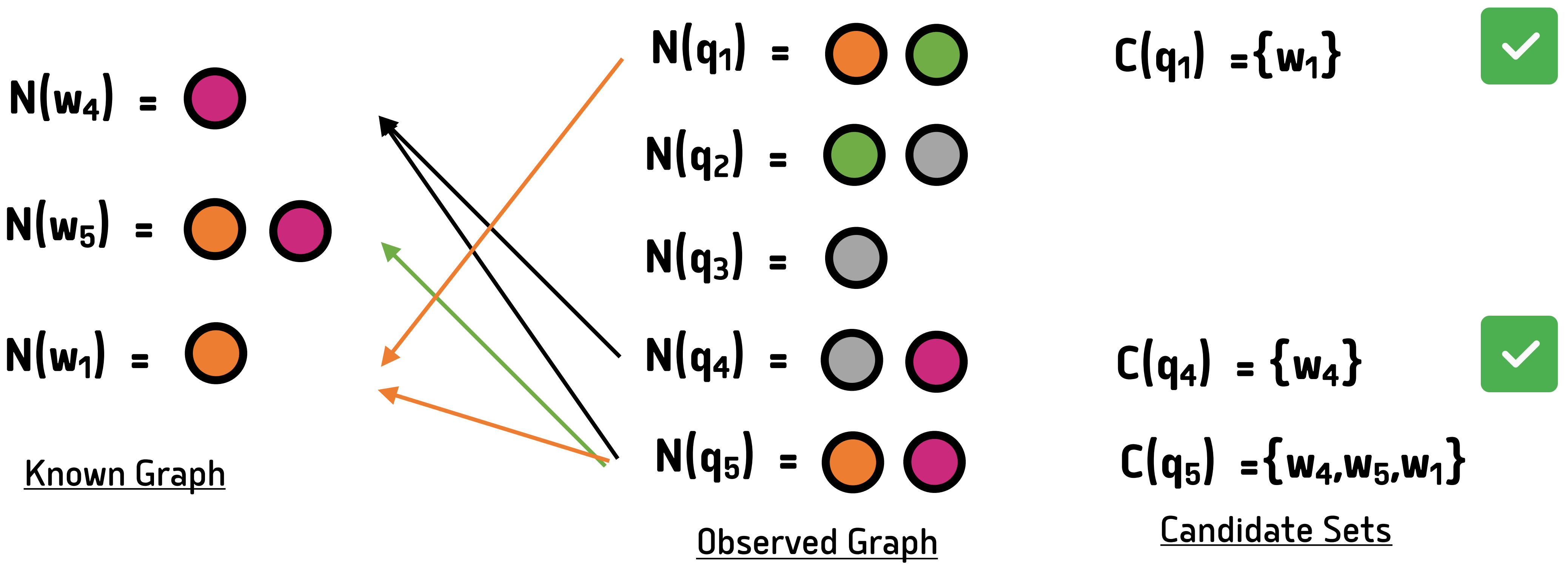
The Subgraph^{VL} Attack

- If a single word is left that's the match
- Remove it from other queries' candidate sets



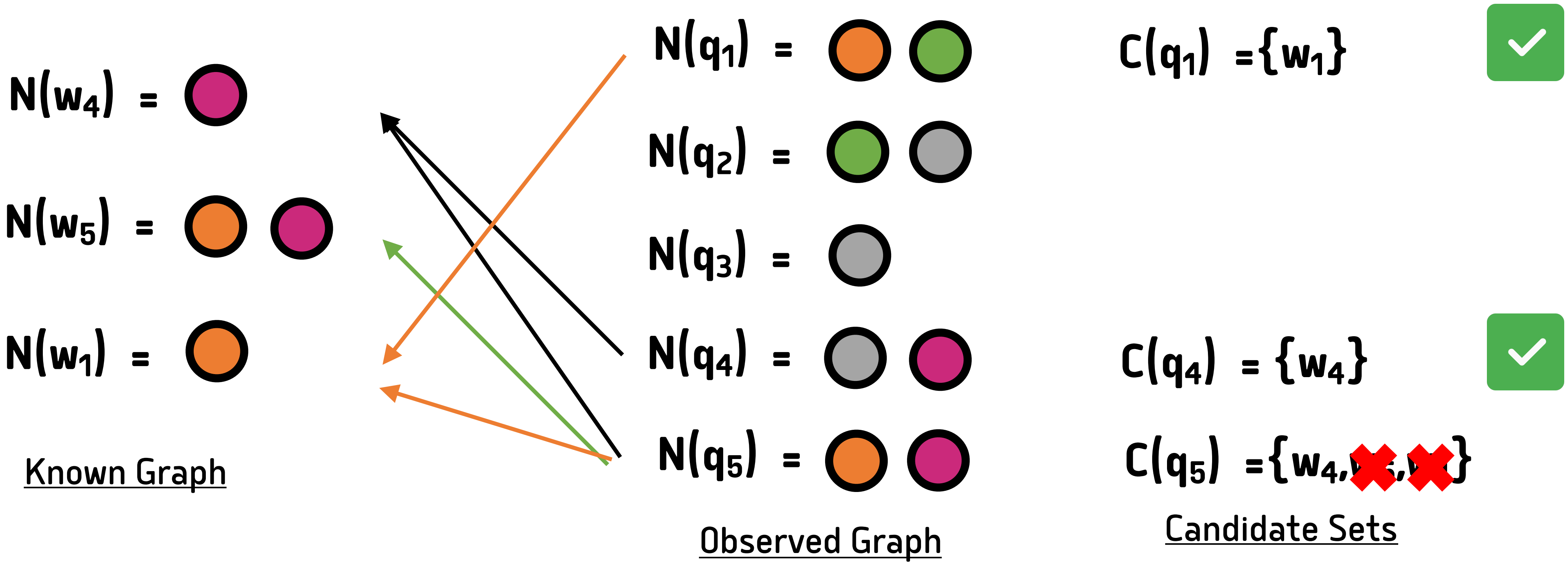
The Subgraph^{VL} Attack

- If a single word is left that's the match
- Remove it from other queries' candidate sets



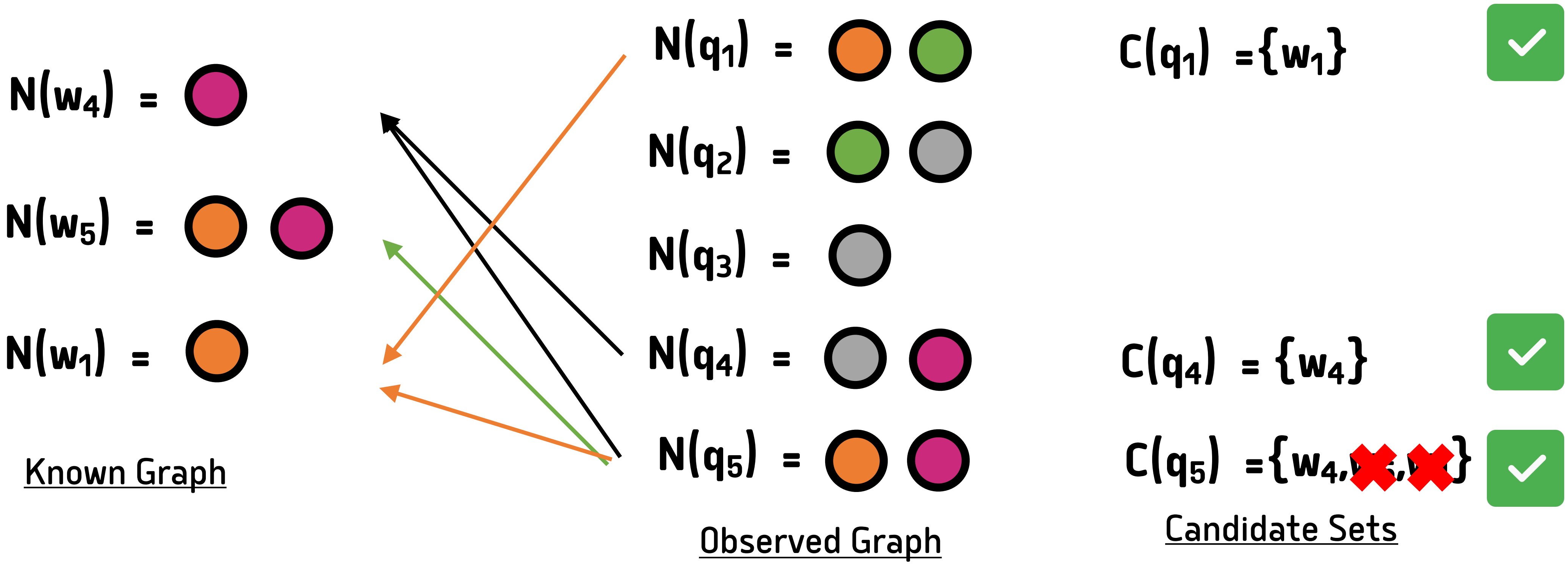
The Subgraph^{VL} Attack

- If a single word is left that's the match
- Remove it from other queries' candidate sets



The Subgraph^{VL} Attack

- If a single word is left that's the match
- Remove it from other queries' candidate sets

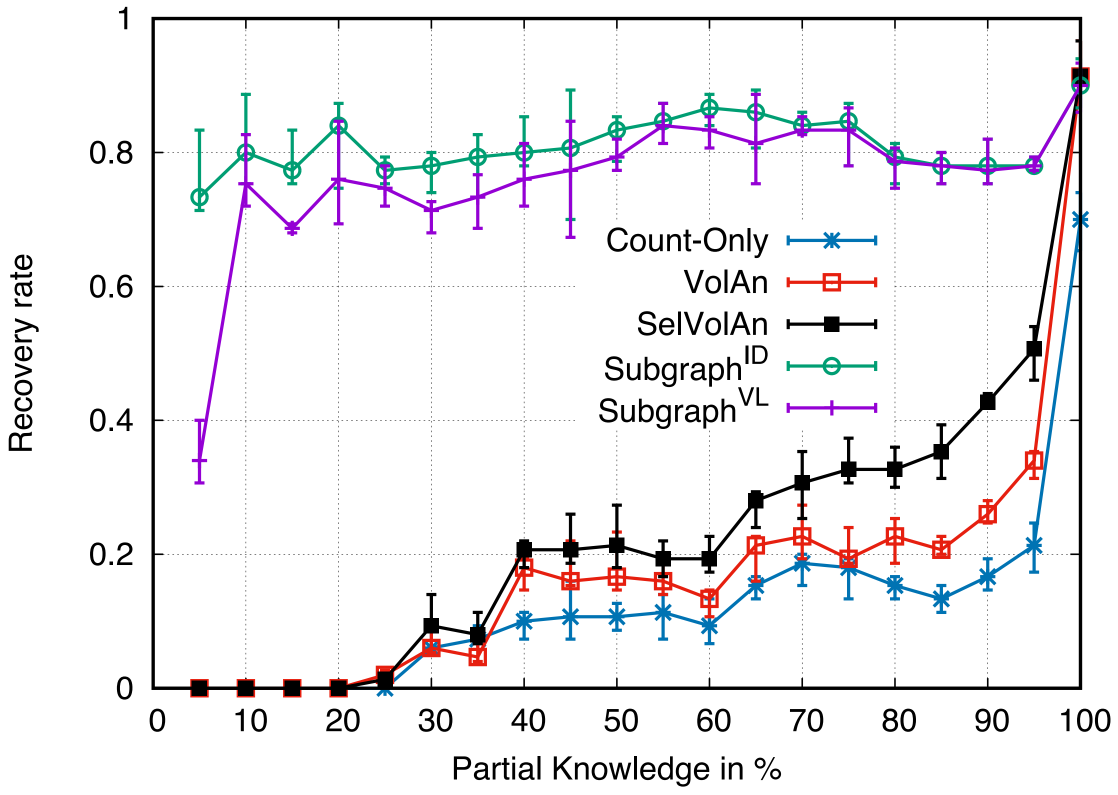


Evaluation of our Attacks
Setting

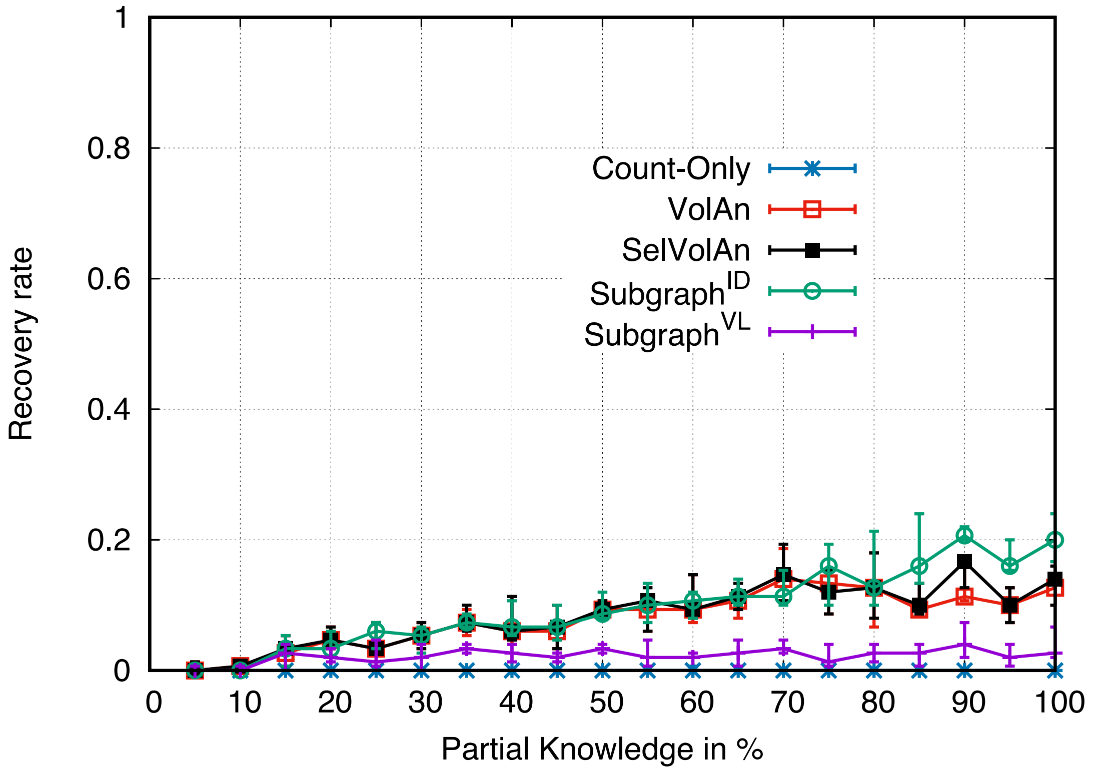
- Enron dataset:
  - ~500K emails
  - Folder for every employee
- Creation of different document collections
  - One user setting
  - Multiple user setting
- Size of the query space: 500 & 5000
- Composition of the query space
- Query frequency::high, pseudo-low, low

Evaluation of our Attacks

Single User - 500 Keywords - Entire composition



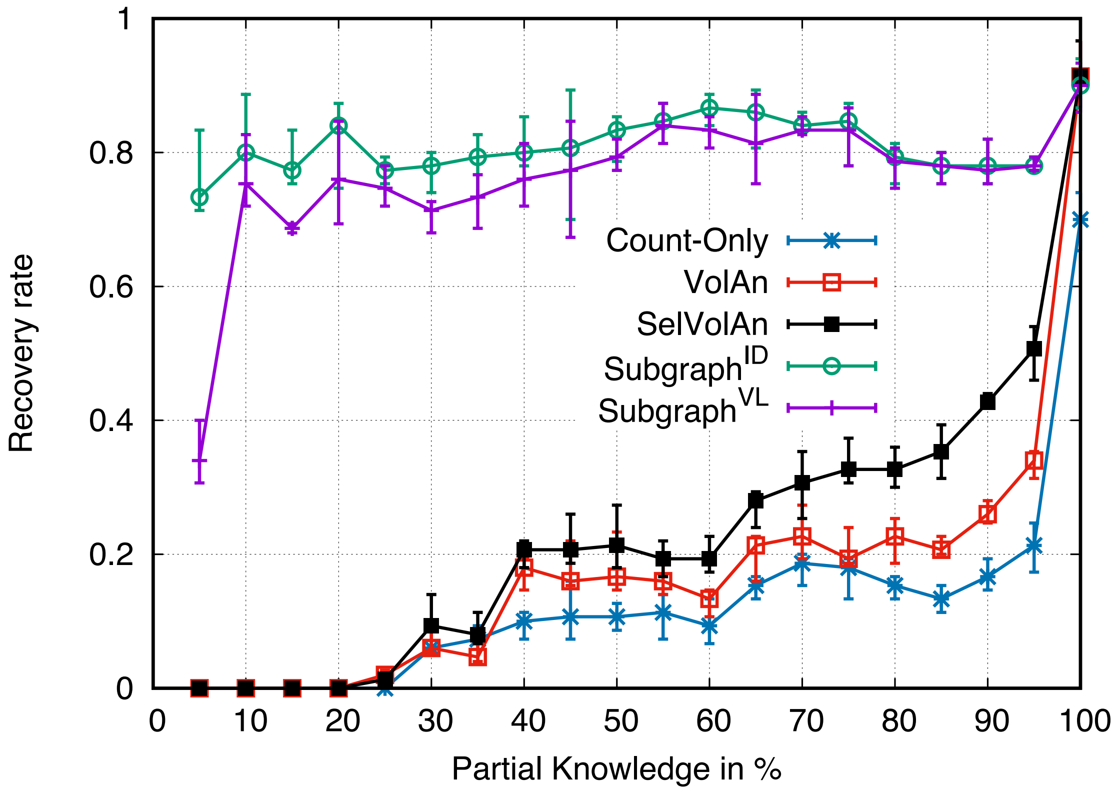
High-selectivity



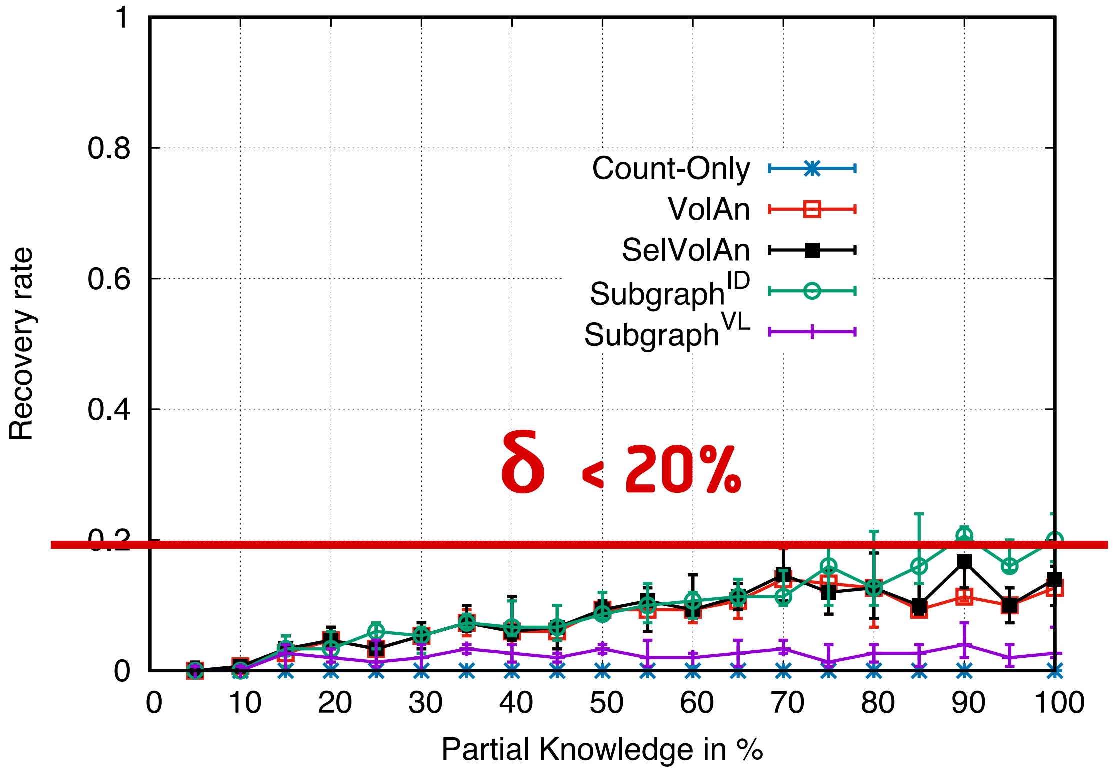
Low selectivity

Evaluation of our Attacks

Single User - 500 Keywords - Entire composition



High-selectivity



Low selectivity

Summary of our Attacks

Against Enron Dataset

δ needed for RR ≥ 20%

Attack	Type	Pattern	Known Queries	δ for HS	δ for PLS	δ for LS
IKK	known-data	co	Yes	≥95%	?	?
Count	known-data	rlen	Yes/No	≥80%	?	?
ZKP	injection	rid	No	N/A	N/A	N/A
Subgrap^{ID}	known-data	rid	No	≥5%	≥50%	≥60%
Subgraph^{VL}	known-data	vol	No	≥5%	≥50%	δ=1 recovers<10%
VolAn	known-data	tvol	No	≥85%	≥85%	δ=1 recovers<10%
SelVolAn	known-data	tvol, rlen	No	≥80%	≥85%	δ=1 recovers<10%
Decoding	injection	tvol	No	N/A	N/A	N/A
Binary	injection	Tvol	No	N/A	N/A	N/A

Very theoretical

Theoretical

Practical

Takeaways

- Cryptanalysis in Encrypted search should be more "**nuanced**" – there is a lot more to learn!
- Baseline STE is still **OK** for low-selectivity queries
- ORAM-based search is also vulnerable to volume-based known-data attacks
- ORAM-based search is also vulnerable to injection attacks
- Subgraph attacks are practical for high-selectivity queries
  - need only $\delta \geq 5\%$
- **Countermeasures**
  - for $\delta < 80\%$ use OPQ [this work]
  - for $\delta \geq 80\%$ use PBS [Kamara-**M**-Ohrimenko18] or use VLH or AVLH [Kamara-**M**19]

Thank you!

https://eprint.iacr.org/2019/1175