


TRUSTWORTHY, PRIVACY-PRESERVING AND FUNCTIONAL DATA OUTSOURCING SYSTEMS

Ph. D. Student: Tung Le

Advisor: Dr. Thang Hoang

Computer Science Department,

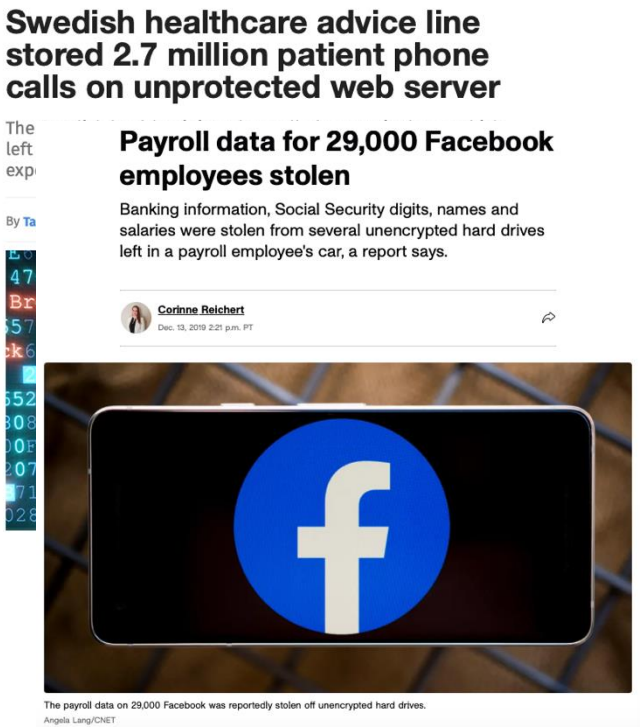
Virginia Tech

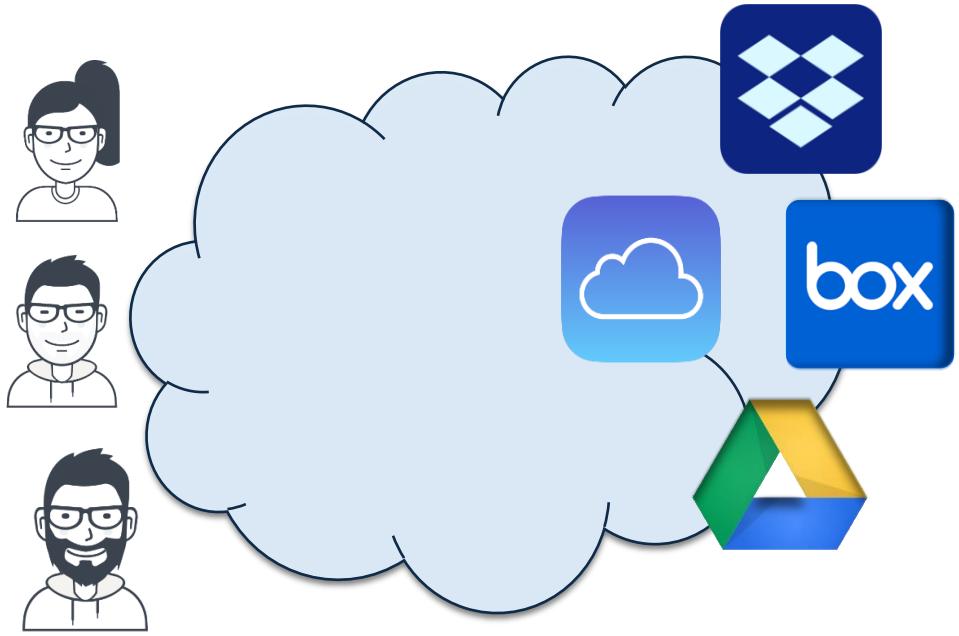
December 17, 2024

Ph. D. Committee:

1. Dr. Thang Hoang
2. Dr. Lenwood S. Heath
3. Dr. Daphne Yao
4. Dr. Wenjing Lou
5. Dr. Rouzbeh Behnia


Overview





Storage-as-a service (STaaS)

- **Misuse** of personal sensitive data (Facebook/Cambridge Analytica)

- Data **breaches** of large enterprises (Yahoo!, Sony PSN, Equifax)

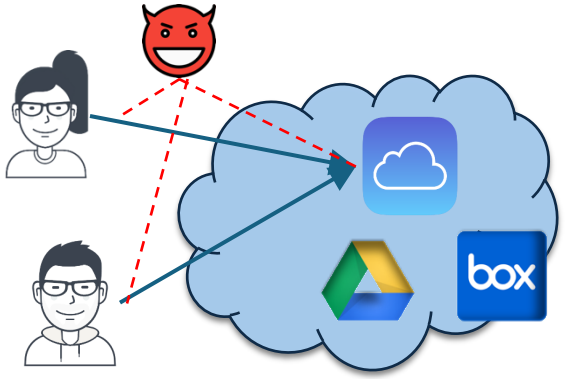
The need for encrypted storage platforms

Is Data Confidentiality Enough?

End-to-end encrypted systems are increasingly popular

Keybase PREVEIL SpiderOak sync.com tresorit

Data is always kept encrypted, however:



- Data integrity and soundness are still concerns
- Sensitive information can still be inferred from **metadata**

(e.g., query/access pattern and frequency, side-channel information)

"Metadata absolutely tells you everything about somebody's life. If you have enough metadata, you don't really need content" –

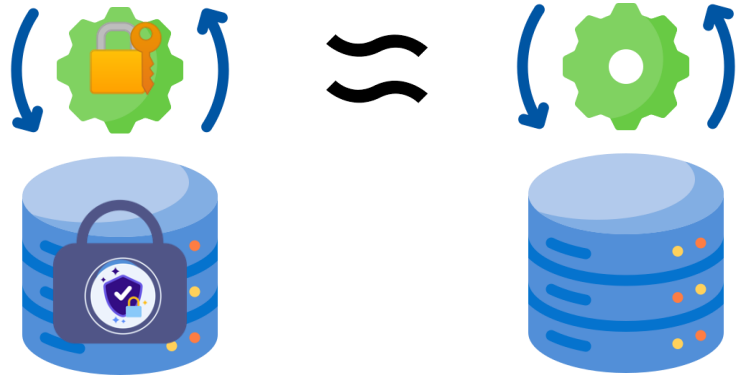
A former NSA General Counsel

- Inefficient and insecure operations that leak user data and queries

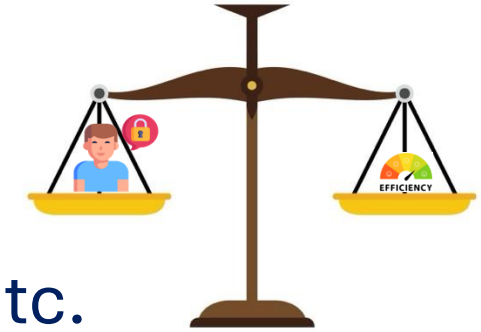
Desirable Properties

Trustworthy data outsourcing services are expected to:

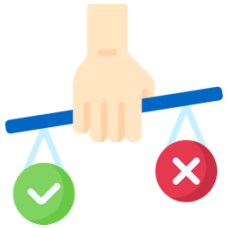
1. **Keep** user **data intact**



2. **Ensure** data and user **privacy**



3. **Provide** essential **functionalities**: querying, analytics, etc.



1. Data Intactness

- Data loss can happen due to unwanted accidents or adversarial behaviors

- A data owner/user expects the following guarantees:

  - Authenticated storage
  - Retrievability

is my data safe?

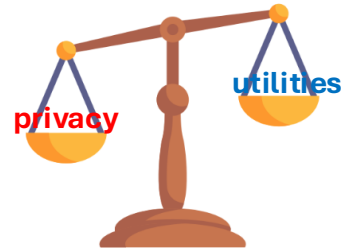
- The user can download the whole data and check it

High communication cost and significant overhead

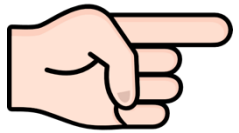
- **Proof of Retrievability** can offer the above guarantees with small user and/or server overhead

2. User/Data Privacy and Utilization Dilemma

- There is a dilemma between user/data privacy and utilization



- Data is encrypted



How to execute queries on plaintext data, such as:

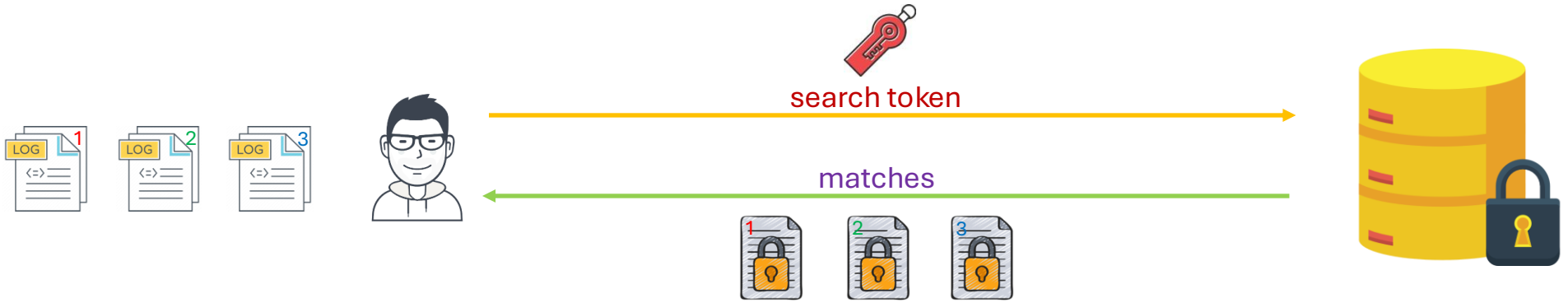
- **Search query**: obtain documents matching a specific keyword
- **Data analytics**: obtain statistical information

- There are encrypted systems with these built-in capabilities

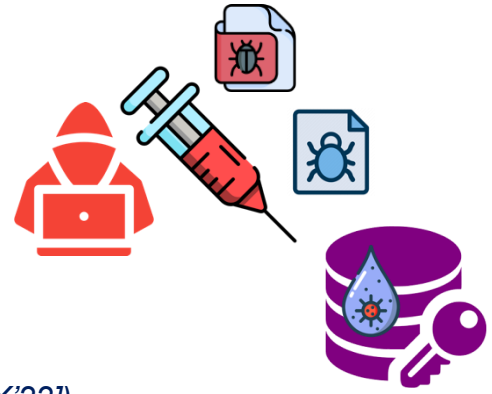
- Costly crypto tools (e.g., Multiparty Computation, Homomorphic Encryption)
- **Metadata** leakage

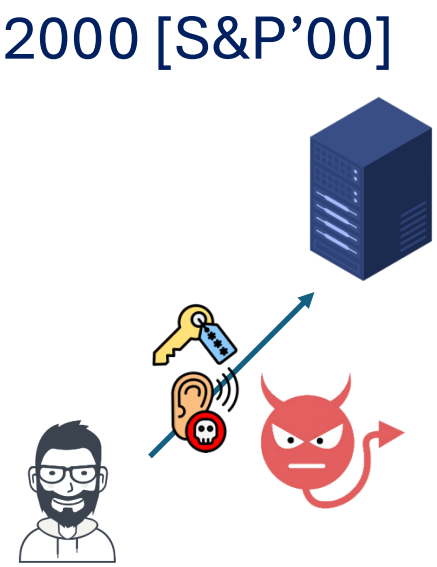
2. User/Data Privacy and Utilization Dilemma

- How to support encrypted search securely and efficiently?



- **Searchable Encryption** (SE) was first proposed in 2000 [S&P'00]
- **Vulnerable** to many types of attack:
  - File-injection attacks
  - Keyword-guessing attacks*
  - Leakage-abuse attacks





* *For public-key SE only (e.g., [EUROCRYPT'04, USENIX'22])*

Examples of Metadata Leakage Attacks

- There are potential attacks exploiting **metadata**. For example:

Access Pattern Leakage Attack

File-Injection Attack

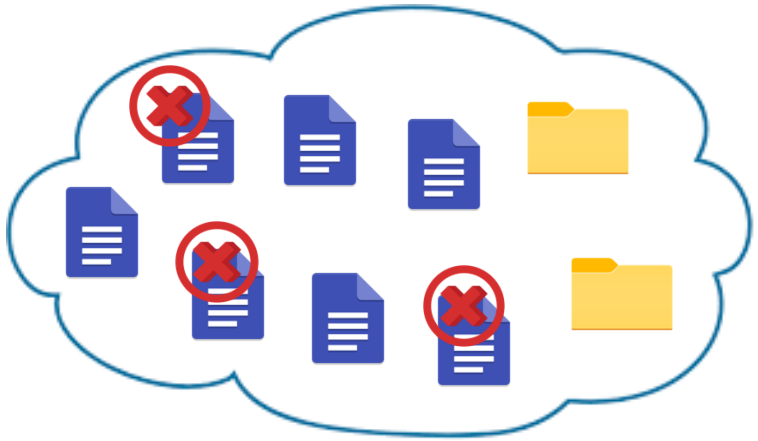
Data intactness

Data and query confidentiality

The goal of my dissertation is to efficiently resolve **security**, **privacy**, and **functionalities** issues simultaneously in data outsourcing systems

Data/Query Searchability

Authenticated Storage



Data loss can happen due to:

- Hardware failures 
- Adversarial behaviors 

Two Decades of Proof of Retrievability (PoR)

Audit Request

Audit Proof

First proposal

Proof of Retrievability [CCS'07]: Error Correcting Code (ECC), Static

Provable Data Possession [CCS'07]: Integrity of *most* data

2007

[CRYPTO'11]
Verifiable Computation
Static
Audit: $O(N)$

2011

[CCS'13]
Dynamic
Incrementally Constructible Code (ICC)
Update/Audit: $O(\log N)$
Proof size: $O(\beta + \lambda^2 \log N)$

2013

[USENIX'21]
Dynamic
Low server storage cost
Audit: $O(N)$

2021

2009

[CCSW'09]
ECC
↑ Error rate
Static

2012

[ACSAC'12]
Dynamic
Update: $O(\log\log N)$
Audit: $O(\sqrt{N})$

2017

[JoC'17]
Dynamic, ECC
Efficient update
Update/Audit: $O(\log^2 N)$

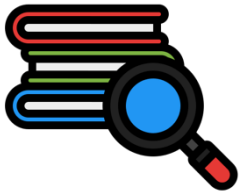
2023

[NDSS'23] (Ours)
Dynamic
ICC
Update/Audit: $O(\log N)$
Proof size: $O(\log \beta)/O(1)$

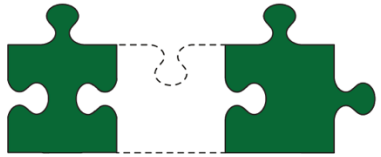
N: # blocks

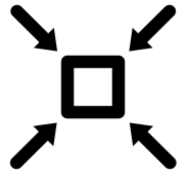
β: block size

λ: security parameter



Efficient Dynamic PoR for Cold Storage





- **Research Gap:**
  - Minimal audit cost: audit proof size and end-to-end audit latency

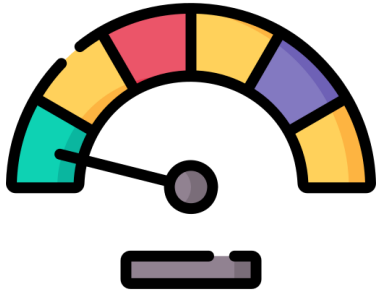
- Our **Porla** [NDSS'23]:

  - *Minimize* audit cost:
    - ✓ Audit bandwidth: $O(\log \beta)$ or $O(1)$, where:
    - ✓ Server/Client: $O(\lambda \log N)$

N: #data blocks

β: data block size

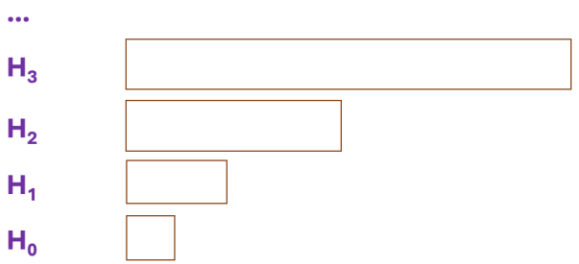
λ: security parameter



  - Maintain a reasonable update performance:
    - ✓ Server: $O(\log N)$
    - ✓ Client/Bandwidth: $O(\beta)$

[NDSS'23] **Tung Le**, Pengzhi Huang, Attila A. Yavuz, Elaine Shi, and Thang Hoang. "Efficient Dynamic Proof of Retrievability for Cold Storage." In *2023 Annual Network and Distributed System Security Symposium (ISOC NDSS 2023)*, San Diego, CA, March 2023.

Main Techniques:

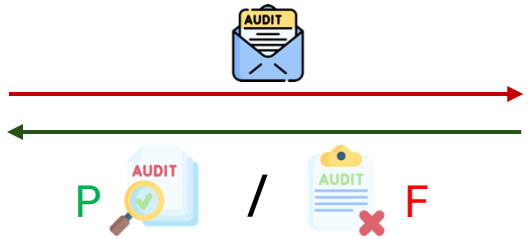


- Incrementally Constructible Code

- Homomorphic MAC

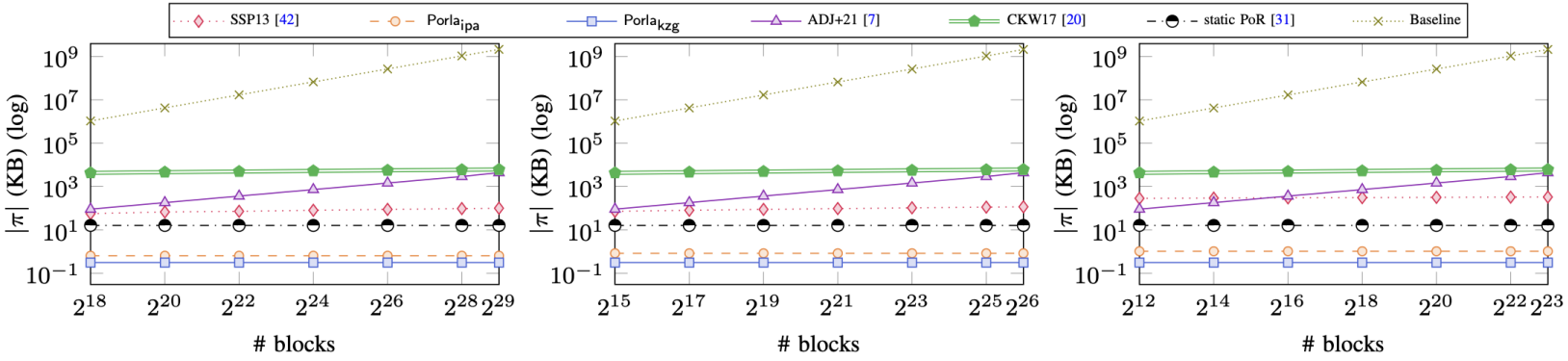
- Verifiable Computation Techniques

- Support Public Audit



Porla Achievements

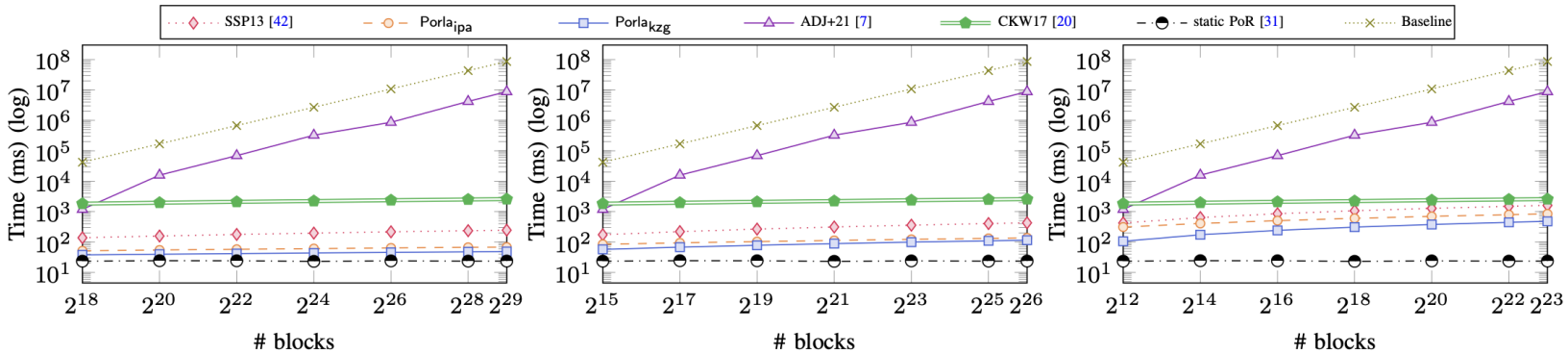
- 87 × −14,012 × smaller proof size than previous DPoR schemes



(a) $\beta = 4\,\text{KB}$ (b) $\beta = 32\,\text{KB}$ (c) $\beta = 256\,\text{KB}$

Fig. 6: Audit proof size of Porla and its counterparts.

- 4 × −18,000 × faster audit time than prior approaches



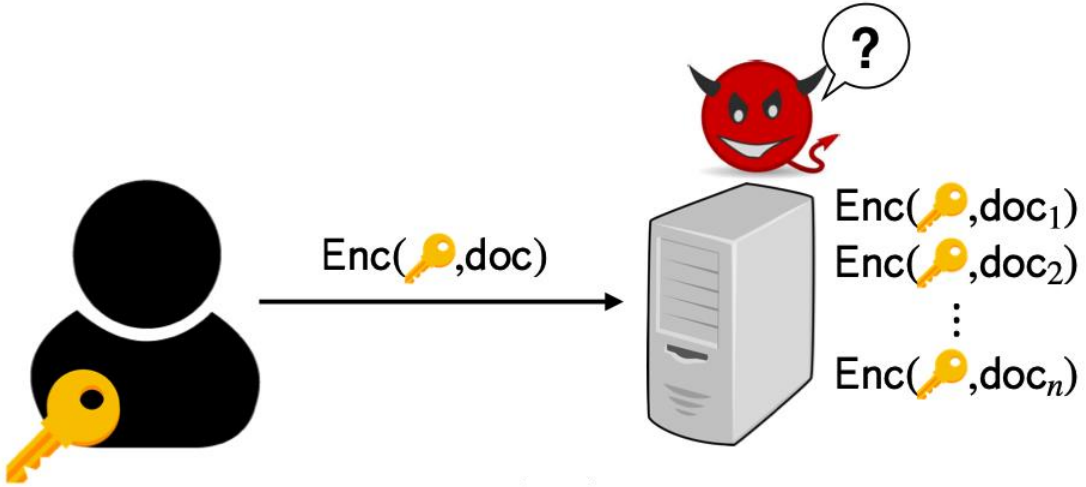
(a) $\beta = 4\,\text{KB}$ (b) $\beta = 32\,\text{KB}$ (c) $\beta = 256\,\text{KB}$

Fig. 7: End-to-end audit delay of Porla and its counterparts.

Searchable Encryption: Motivation

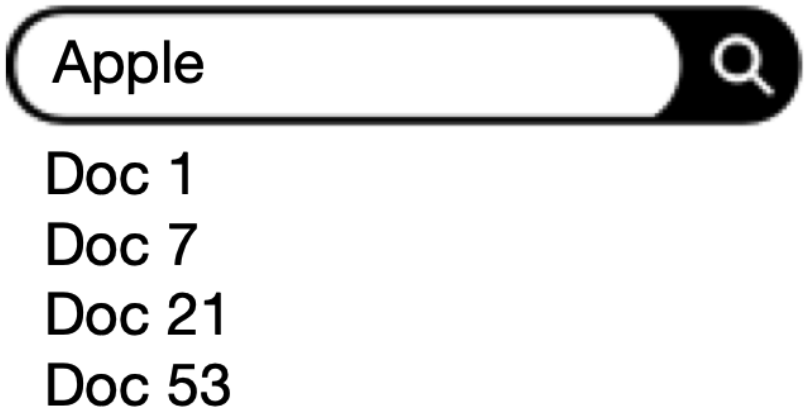
End-to-End Encryption (E2EE)

E2EE provides strong security guarantees if attacker compromises server

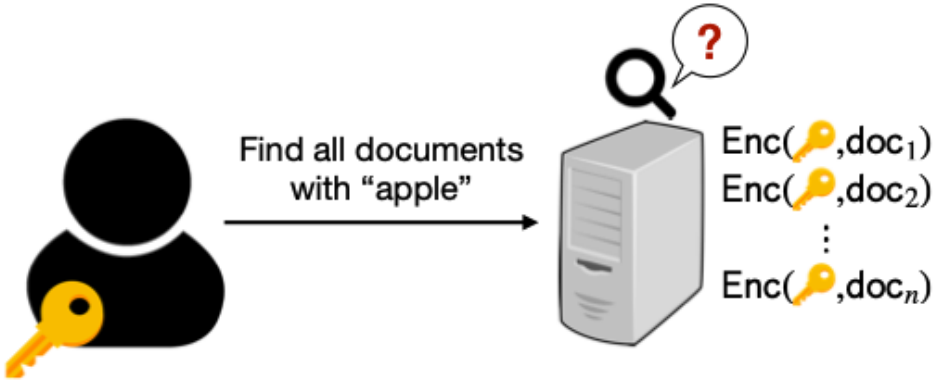


Searchable Encryption: Motivation

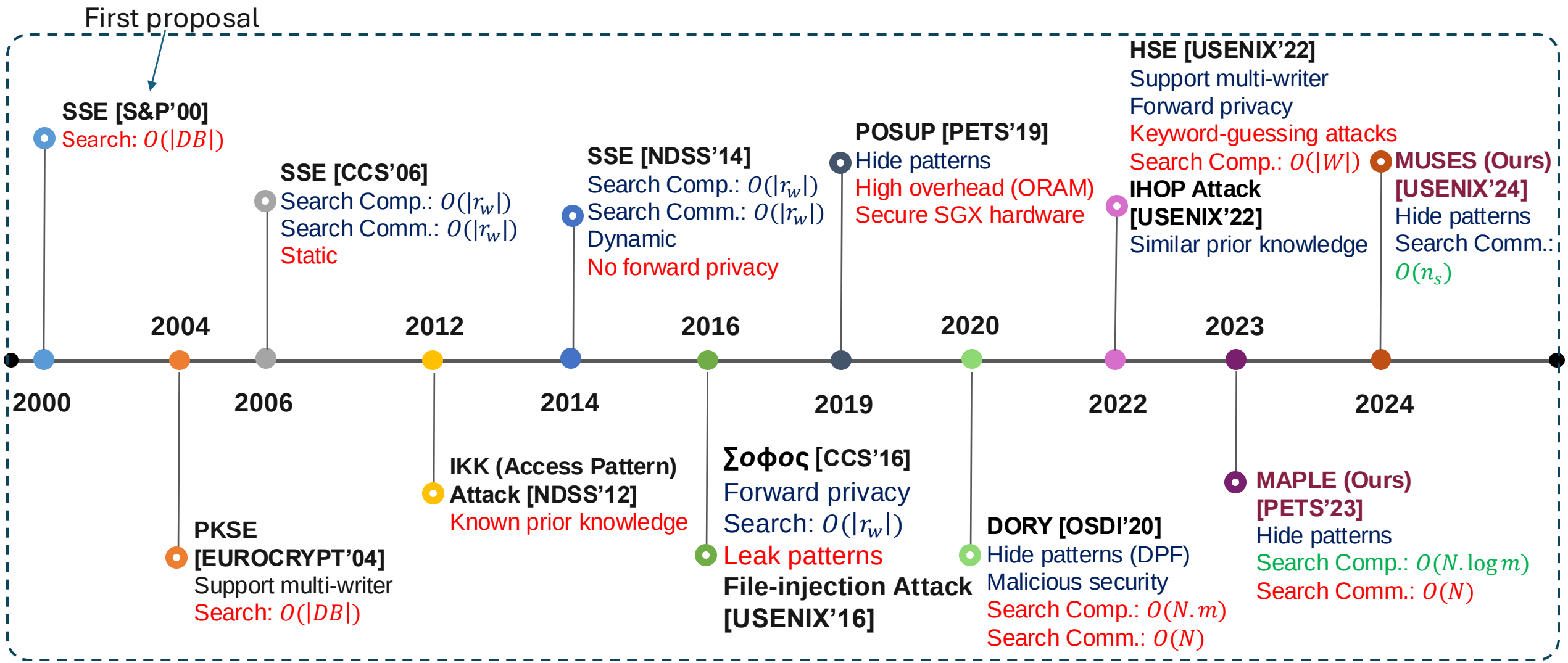
Users expect the ability to execute search



Challenge: server cannot decrypt data to search



20 Years+ of Searchable Encryption (SE)

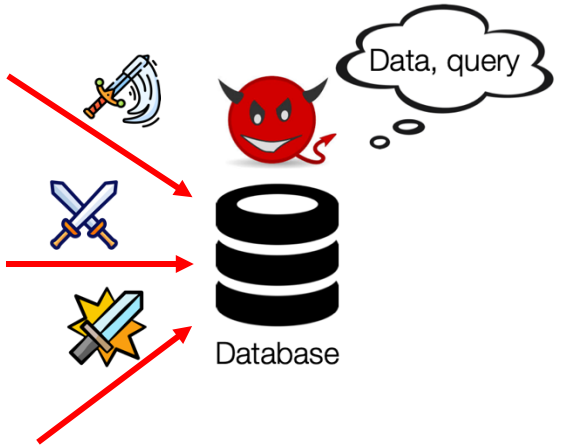


Improve communication, computation efficiency, and security

DB: database N: #documents W: keyword universe set m: keyword representation size n_s: search result size r_w: #matches

Numerous Leakage-Abuse Attacks in Searchable Encryption:

- **Search Pattern:** Repetition in search queries [USENIX'21, USENIX'22, CCS'23, USENIX'24]



- **Result Pattern:** Repetition in matching documents [NDSS'12, CCS'15, CCS'16, NDSS'20, CCS'21, NDSS'22, USENIX'22]

- **Volume Pattern:** Repetition in the number of matching documents [CCS'15, USENIX'21, CCS'23, USENIX'24]

Our MAPLE [PETS'23]

Research Gap:

- Hide search result pattern with search complexity $O(N.m)$, where N is the number of documents and m is the keyword representation size
- Limited multi-user support: assume all users are trusted or control access policies based on access level

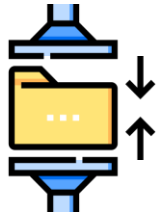
Our **MAPLE** [PETS'23]:

- Server search complexity: $O(N.\log m)$
- Hide *all* metadata: search, result and volume patterns

- Multi-user with fine-grained access control

[PETS'23] **Tung Le**, and Thang Hoang. "MAPLE: A Metadata-Hiding Policy-Controllable Encrypted Search Platform with Minimal Trust." In 2023 Privacy Enhancing Technologies Symposium *(PETS 2023)*, Lausanne, Switzerland, July 2023.

Main techniques:

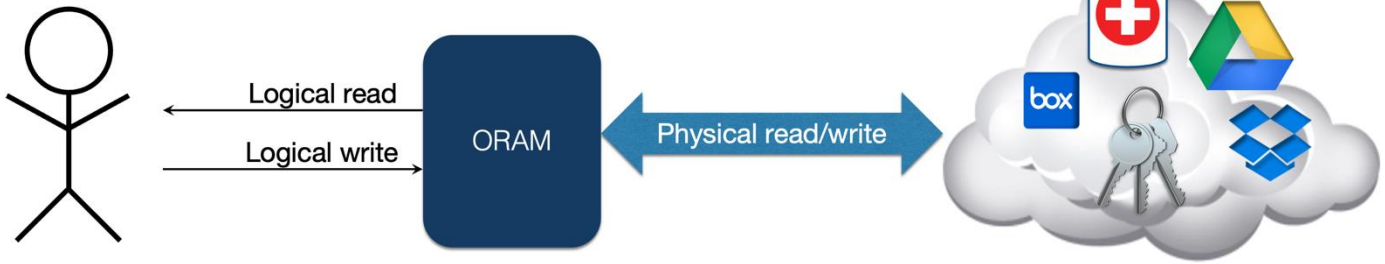


- Bloom Filter to compress search index

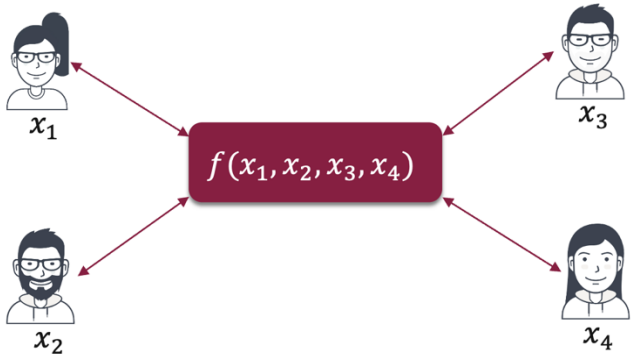
	 "amazon"	 "google"	 "netflix"	...	 "apple"
doc 1	1	0	1	0	1
doc 2	0	1	1	0	0
doc 3	1	1	0	0	0
...	0	1	0	1	1
doc N	1	0	0	1	1

Bitmap for keywords in doc 2

- Oblivious Random Access Machine (ORAM)
  - Circuit ORAM
  - Oblivious Table



- Multiparty Computation



MAPLE Achievements

- MAPLE is $2.6\times - 10.7\times$ slower than DORY with BF size $\leq 2^{14}$, and starts to outperform when BF size $\geq 2^{16}$

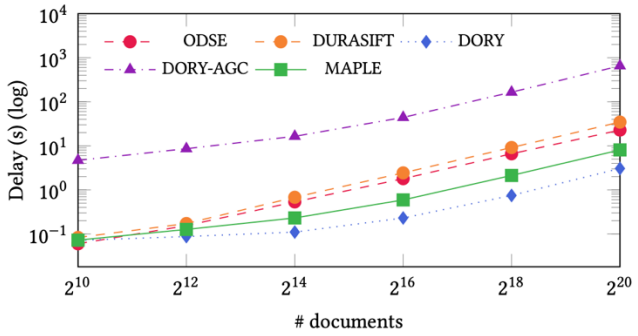


Figure 5: Search delay of MAPLE and its counterparts.

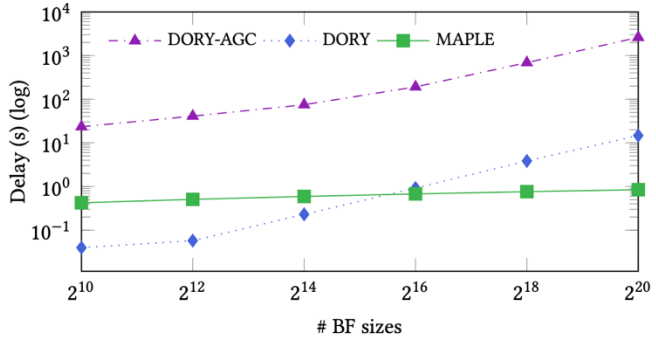
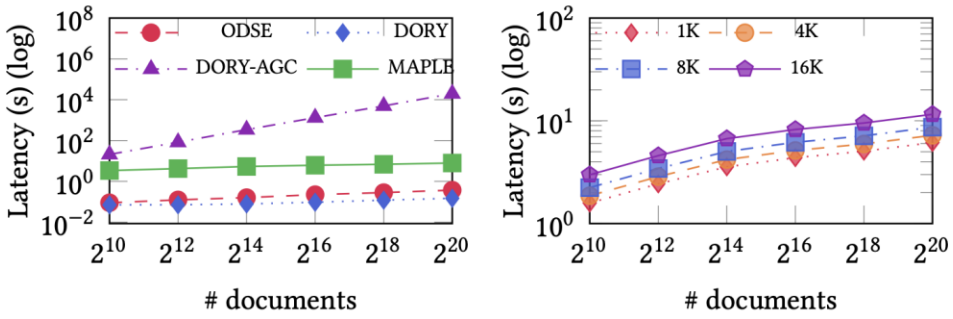


Figure 6: Search delay with varied BF sizes.

- MAPLE is 3.3s – 7.8s slower to achieve oblivious update

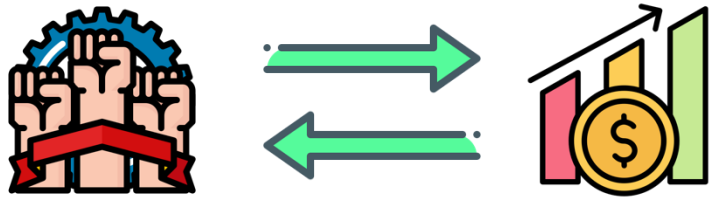


(a) Document update (b) Permission update

Figure 8: Update delay of MAPLE and its counterparts.

Our MUSES [USENIX'24]

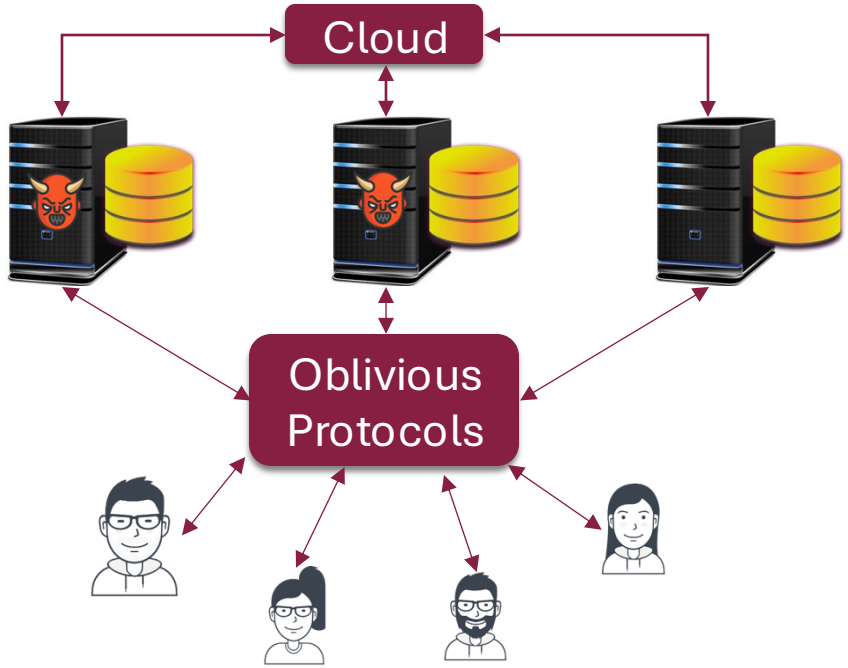
- Generic MPCs are powerful but expensive



- Distributed computations specifically designed for a particular computation task are more efficient

Our **MUSES** [USENIX'24]:

- Hide *all* statistical information: search, result, and volume patterns
- Minimal user overhead for search and permission revocation

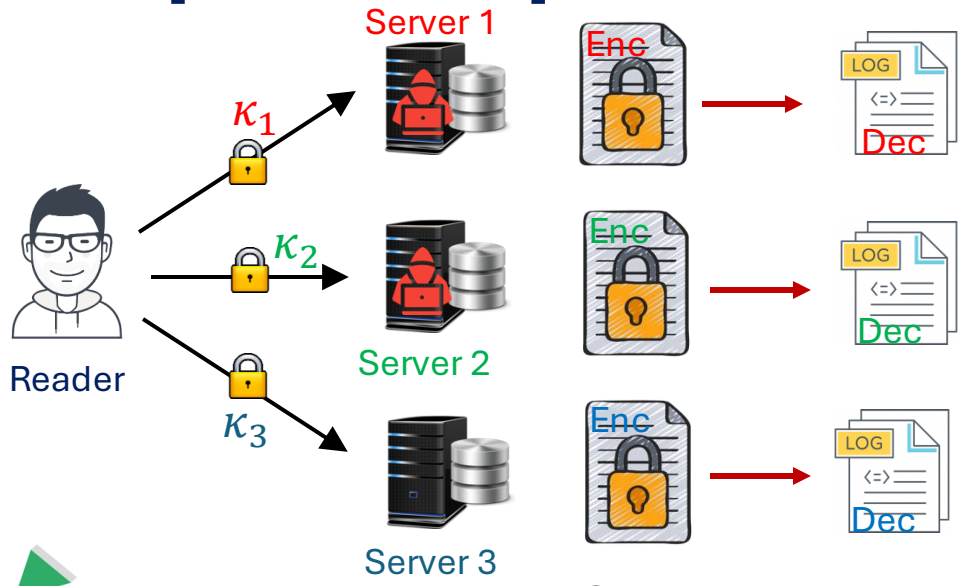


[USENIX'24] **Tung Le**, Rouzbeh Behnia, Jorge Guajardo, and Thang Hoang. "MUSES: Efficient Multi-User Searchable Encrypted Database." In USENIX Security Symposium *(USENIX Security 2024)*, Philadelphia, PA, August 2024.

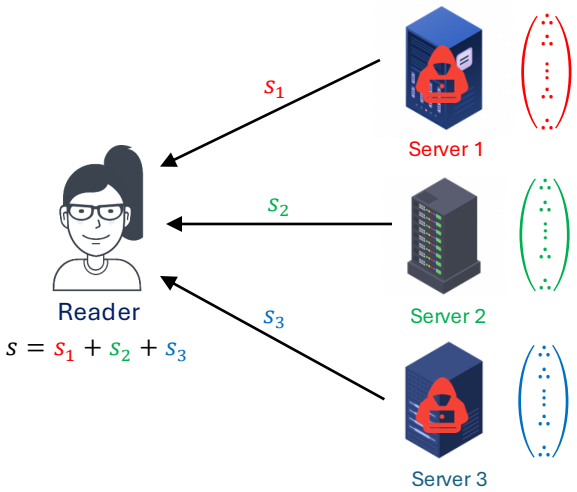
Main techniques:

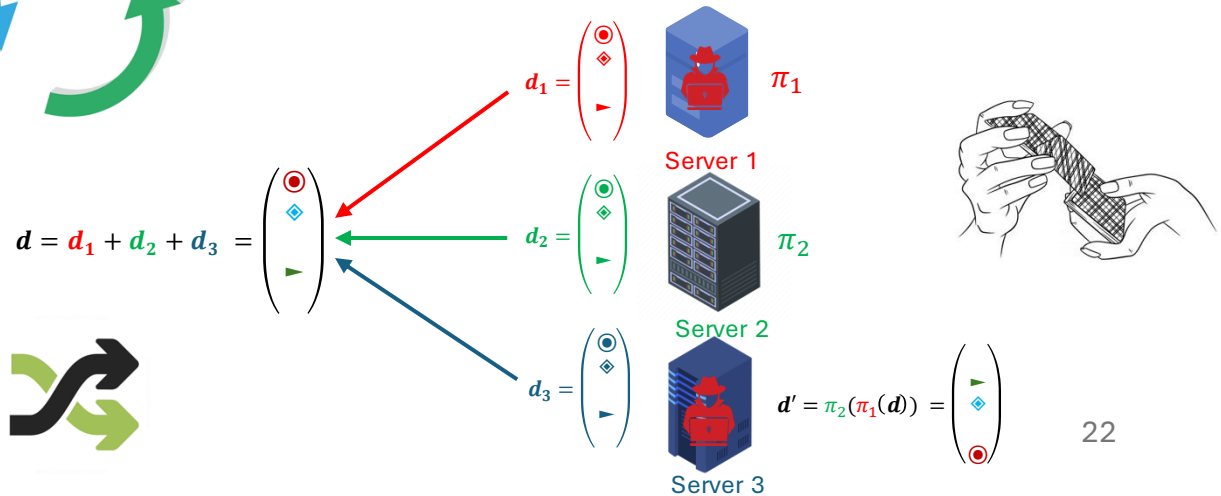
- Key-Homomorphic Pseudorandom Function [CRYPTO'13]

  Delegating Decryption



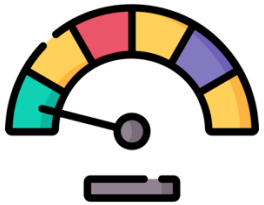
- Our Multiparty Oblivious Counting





- Our Multiparty Oblivious Shuffling

MUSES Achievements: Keyword Search

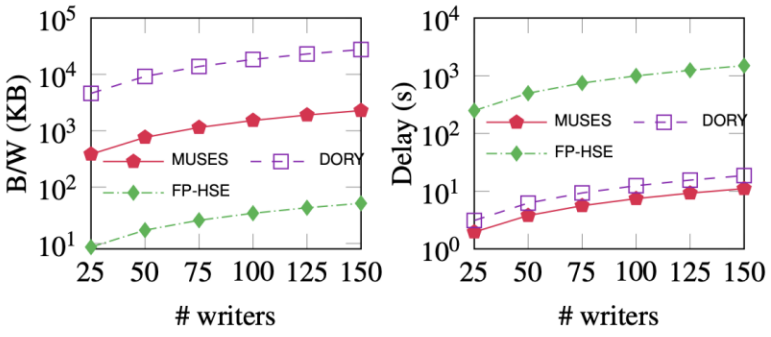


Reader's bandwidth:

12 × –97 × smaller than DORY (hide patterns), 6 × larger than FP-HSE (leak patterns)

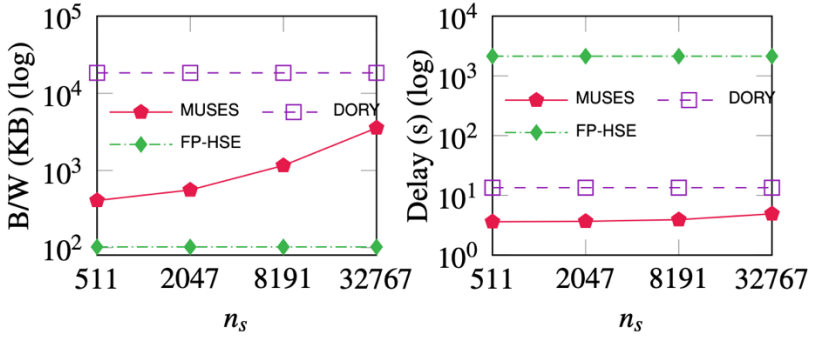
End-to-end latency:

2 × –4 × faster than DORY, 127 × –632 × faster than FP-HSE



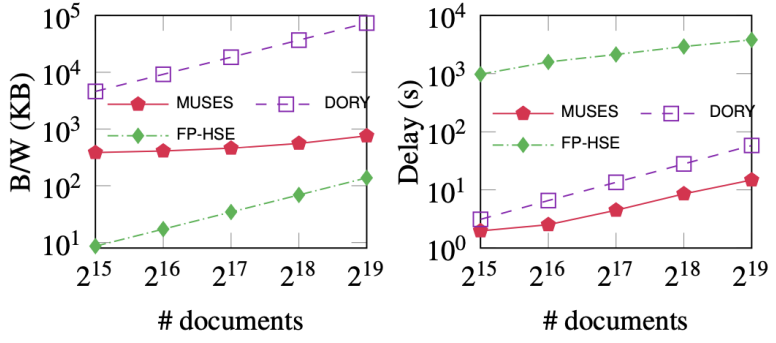
(a) Reader's bandwidth (b) E2E delay

Figure 6: Keyword search performance (log scale on y-axis).



(a) Reader's bandwidth (b) E2E delay

Figure 10: Keyword search performance with varying n_s.



(a) Reader's bandwidth (b) E2E delay

Figure 11: Keyword search performance w/ varying database sizes.

MUSES Achievements: Permission Revocation

Writer's bandwidth:

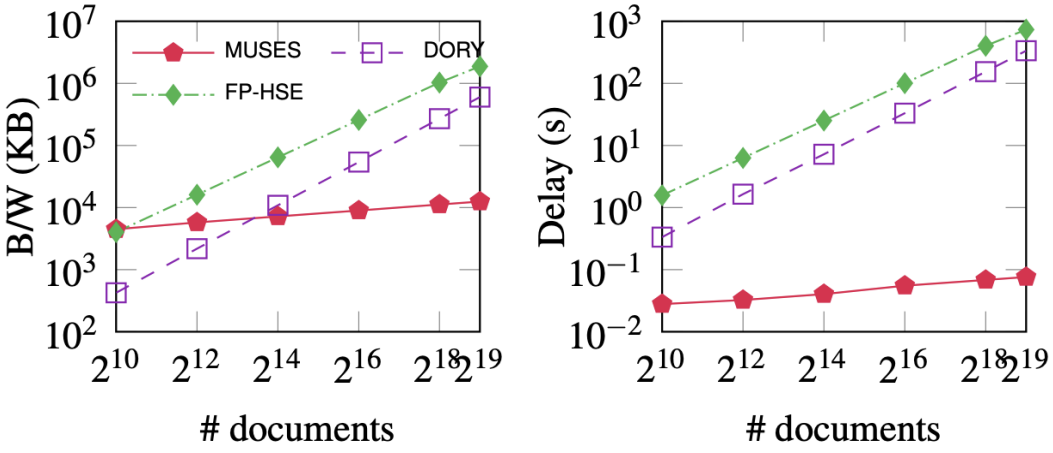
$2\times - 150\times$ smaller than DORY/FP-HSE

End-to-end latency:

$2\times - 6\times$ faster than DORY/FP-HSE

Writer's latency:

$12\times - 9600\times$ faster than DORY/FP-HSE



(a) Writer's bandwidth (b) Writer's latency

Figure 7: Permission revocation performance (log scale on y-axis).

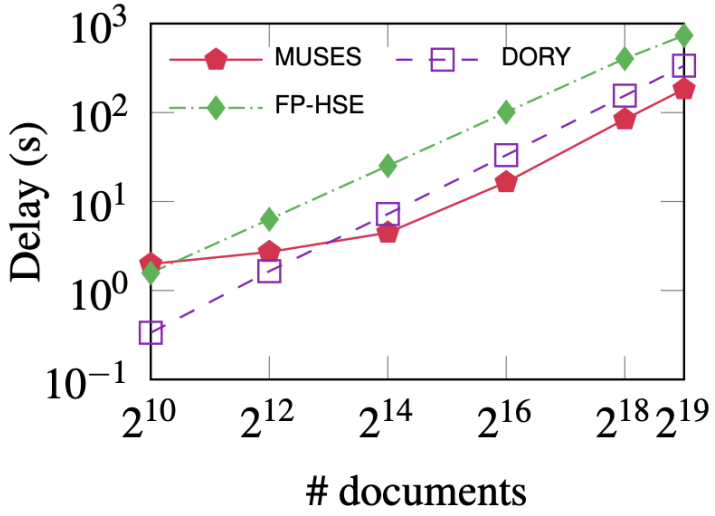


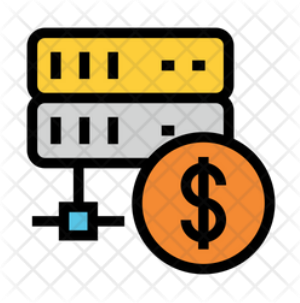
Figure 8: E2E permission revocation delay (log scale on y-axis).

Ongoing Work

- Our prior work relies on distributed computation for secure search
  - Expensive deployment and maintenance cost

- PKSE [EUROCRYPT'04, USENIX'22] can support multi-user more naturally in practical settings (e.g., email, messaging)

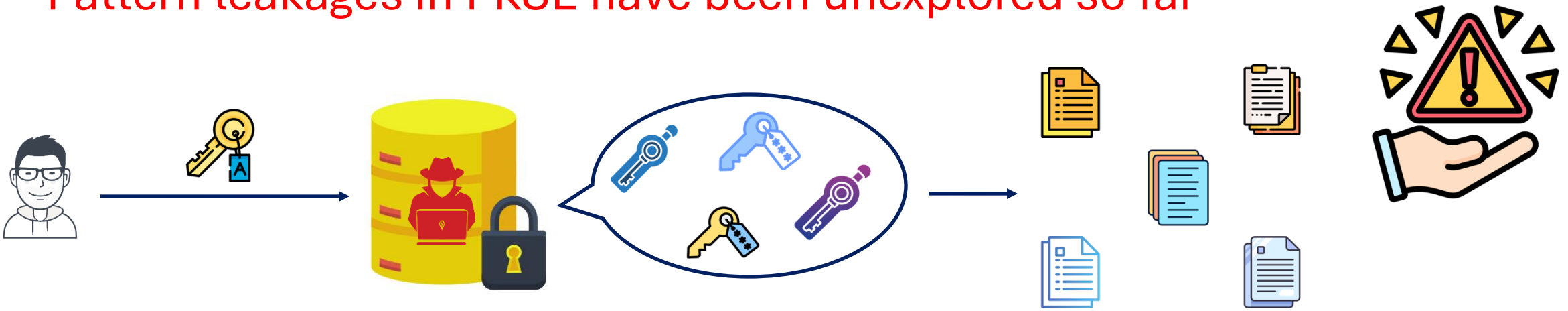
- Many open problems:
  - Keyword-guessing attacks
  - Inefficient forward privacy
  - High server computation cost for search



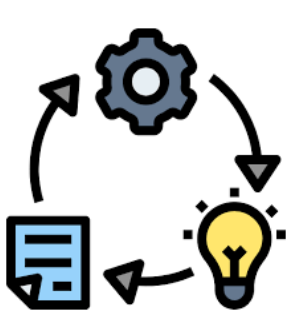
- This work addresses the above fundamental security and performance issues

Future Work

- Pattern leakages in PKSE have been unexplored so far



- We aim to resolve pattern-leakage attacks in public-key settings while maintaining/improving efficiency




Conclusion

- My dissertation aims to: 
  - Design an authenticated and retrievable data storage system
  - Address user/data privacy and utilization dilemma: provide efficient search functionality while preventing information leakage

 All are essential to build practical encrypted data outsourcing systems providing high performance and security guarantees






THANK YOU FOR YOUR ATTENTION

Q&A

VIRGINIA TECH®

References

[S&P'00] Dawn Xiaoding Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000*, pages 44–55, 2000.

[EUROCRYPT'04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. Cryptology ePrint Archive, Paper 2003/195, 2003. https: //eprint.iacr.org/2003/195.

[CCS'06] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, page 79–88, New York, NY, USA, 2006. Association for Computing Machinery.

[CCS'07] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. Provable data possession at untrusted stores. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 598–609, 2007.

[CCS'07] Ari Juels and Burton S Kaliski Jr. Pors: Proofs of retrievability for large files. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 584–597, 2007.

[CCSW'09] Kevin D Bowers, Ari Juels, and Alina Oprea. Proofs of retrievability: Theory and implementation. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 43–54, 2009.

[CRYPTO'11] Siavosh Benabbas, Rosario Gennaro, and Yevgeniy Vahlis. Verifiable delegation of computation over large datasets. In *Annual Cryptology Conference*, pages 111–131. Springer, 2011.

[ACSAC'12] Emil Stefanov, Marten van Dijk, Ari Juels, and Alina Oprea. Iris: A scalable cloud file system with efficient integrity checks. In *ACSAC'12*, pages 229–238, 2012.

[NDSS'12] Mohammad Saiful Islam, Mehmet Kuzu, and Murat Kantarcioglu. "Access pattern disclosure on searchable encryption: ramification, attack and mitigation". In *NDSS*, 2012.

[CCS'13] Elaine Shi, Emil Stefanov, and Charalampos Papamanthou. Practical dynamic proofs of retrievability. In *ACM CCS'13*, pages 325–336, 2013.

[CRYPTO'13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic prfs and their applications. In *Annual International Cryptology Conference*, 2013.

References

[NDSS'14] David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit S. Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, and Michael Steiner. Dynamic searchable encryption in very-large databases: Data structures and implementation. *IACR Cryptol. ePrint Arch.*, 2014:853, 2014.

[CCS'15] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-abuse attacks against searchable encryption. In *CCS*, 2015.

[CCS'16] David Pouliot and Charles V Wright. The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption. In *CCS*, 2016.

[CCS'16] Raphael Bost. $\sum o\phi o\varsigma$: Forward secure searchable encryption. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 1143–1154, New York, NY, USA, 2016. Association for Computing Machinery.

[JoC'17] David Cash, Alptekin Kupcu, and Daniel Wichs. Dynamic proofs of retrievability via oblivious ram. *Journal of Cryptology*, 2017.

[PET'19] Thang Hoang, Muslum Ozgur Ozmen, Yeongjin Jang, and Attila A Yavuz. Hardware-supported oram in effect: Practical oblivious search and update on very large dataset. *Proceedings on Privacy Enhancing Technologies*, 2019(1), 2019.

[OSDI'20] Emma Dauterman, Eric Feng, Ellen Luo, Raluca Ada Popa, and Ion Stoica. Dory: An encrypted search system with distrust. In *Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation*, OSDI'20, USA, 2020. USENIX Association.

[NDSS'20] Laura Blackstone, Seny Kamara, and Tarik Moataz. Revisiting leakage abuse attacks. In *NDSS*, 2020.

[CCS'21] Jianting Ning, Xinyi Huang, Geong Sen Poh, Jiaming Yuan, Yingjiu Li, Jian Weng, and Robert H Deng. Leap: Leakage-abuse attack on efficiently deployable, efficiently searchable encryption with partially known dataset. In *CCS*, 2021.

[USENIX'21] Gaspard Anthoine, Jean-Guillaume Dumas, Mélanie de Jonghe, Aude Maignan, and Clément Pernet, Michael Hanling, and Daniel S Roche. Dynamic proofs of retrievability with low server storage. In *30th USENIX Security Symposium*, pages 537–554, 2021.

[USENIX'21] Marc Damie, Florian Hahn, and Andreas Peter. A highly accurate Query-Recovery attack against searchable encryption using Non-Indexed documents. In *USENIX Security*, 2021.

[USENIX'21] Simon Oya and Florian Kerschbaum. Hiding the access pattern is not enough: Exploiting search pattern leakage in searchable encryption. In *USENIX Security*, 2021.

References

[USENIX'22] Simon Oya and Florian Kerschbaum. IHOP: Improved statistical query recovery against searchable symmetric encryption through quadratic optimization. In *USENIX Security*, 2022.

[USENIX'22] Jiafan Wang and Sherman S. M. Chow. Omnes pro uno: Practical Multi-Writer encrypted database. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2371–2388, Boston, MA, August 2022. USENIX Association.

[CCS'23] Lei Xu, Leqian Zheng, Chengzhi Xu, Xingliang Yuan, and Cong Wang. 2023. Leakage-Abuse Attacks Against Forward and Backward Private Searchable Symmetric Encryption. In CCS '23.

[NDSS'23] Tung Le, Pengzhi Huang, Attila A. Yavuz, Elaine Shi, and Thang Hoang. "Efficient Dynamic Proof of Retrievability for Cold Storage." In *2023 Annual Network and Distributed System Security Symposium (ISOC NDSS 2023)*, San Diego, CA, March 2023.

[PETS'23] Tung Le, and Thang Hoang. "MAPLE: A Metadata-Hiding Policy-Controllable Encrypted Search Platform with Minimal Trust." In 2023 Privacy Enhancing Technologies Symposium *(PETS 2023)*, Lausanne, Switzerland, July 2023.

[USENIX'24] Tung Le, Rouzbeh Behnia, Jorge Guajardo, and Thang Hoang. "MUSES: Efficient Multi-User Searchable Encrypted Database." In USENIX Security Symposium *(USENIX Security 2024)*, Philadelphia, PA, August 2024.

[USENIX'24] Hao Nie and Wei Wang and Peng Xu and Xianglong Zhang and Laurence T. Yang and Kaitai Liang. "Query Recovery from Easy to Hard: Jigsaw Attack against SSE." In USENIX Security Symposium *(USENIX Security 2024)*, Philadelphia, PA, August 2024.