An analysis of Structured Encryption (StE) compared to other computation techniques on encrypted data

1. Motivation & Background

Outsourcing data and computation is very common nowadays with cloud environments. However, sensitive data with high privacy requirements, such as medical records, cannot be stored as-is on a remote server. It must be encrypted. Classical encryption techniques, such as AES, do not allow the server to make any kind of operation on the data, essentially making the computation power of the cloud useless.

Structured Encryption is a set of techniques that stores structured data encrypted on an untrusted server, while still allowing the client to outsource operations securely.

2. Research Question

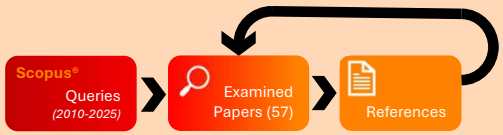
What are the capabilities and limitations of modern Structured Encryption technologies?

How does Structured Encryption perform compared to Oblivious RAM (**ORAM**), Fully Homomorphic Encryption (**FHE**), Trusted Execution Environments (**TEEs**) and Secure Multiparty Computation (**SMC**) ?

- State of the art technologies
- Threat Models / Limitations
- Characteristics Comparison

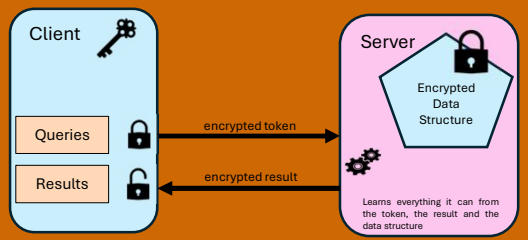
3. Methodology

Snowball Sampling through Backward Reference Searching



4. Technologies & Characteristics

- **SSE (Symmetric Searchable Encryption) :** symmetric-key cryptographic technique allowing a user to search over encrypted labeled data stored on an untrusted server, without revealing the data or the search terms. [1]
- **StE :** generalization of SSE to any data structure. [2]
- **Semi-honest model (honest-but-curious) :** the server performs the operations honestly but learns as much info as possible in the process



4.1 Functionality and Use Cases

- **Labelings**: boolean queries [6] (AND, OR and NOT), add / delete operations [5], substring & wildcard queries
- **Numerical data**: range queries [7]

⇒ Encrypted Databases (SQL queries) [3]

- **Graphs**: adjacency, shortest path and focused subgraph queries [2]
  - Concept and Knowledge Graphs (for AI models) [8]

⇒ **Usages:** MongoDB's Queryable Encryption, Cipherstash, Crypteron

4.2 Efficiency

- **I/O Efficiency Tradeoff** [15]

Storage:	$O(N)$	⟷	$\omega(N)$	in practice → $O(N)$
Locality:	$\omega(1)$		$O(1)$	$o(\log N)$

N is the number of keyword-document pairs, and the locality is the amount of memory accesses for a search

- **Parallelism** : improvement with sufficient cores: $O(r) \rightarrow O(\log N)$ [4]
  r is the size of the query result
- **Concurrency** : lock-free concurrent update operations

4.3 Security

- **CKA1 (Chosen Keyword Attack)** : non-adaptative security [9]
- **CKA2:** adaptative security, attacker can choose keyword after seeing the encryption. This is the standard for StE schemes.
- **UC (Universally Composable)** : a cryptographic protocol remains secure even when composed with other arbitrary protocols. Very strong security but often too heavyweight.
- **Leakage**: information leaked to the server in the setup or query process
  - **Volume Leakage:** the size of the result of a query
  - **Access pattern leakage**: which data corresponds to the query

Attacks	Countermeasures
File-injection attacks on dynamic multimaps: 1% of file injected provides 40% efficiency (2016)	⇒ backward and forward private schemes (addition/deletion do not reveal information about previous and future queries)
Inference attacks on keywords: exponential in the amount of keyword and requires 70% of dataset knowledge to become slightly successful. (2012 & 2015) [10]	⇒ access-pattern hiding schemes (ORAM – based schemes, from 2016) e.g. **TWORAM, EMMavl**
Inference attacks on range queries: only $O(N \log N)$ queries needed to reconstruct the entire data (2017 & 2018)	⇒ volume-hiding schemes (from 2018) e.g. **d-DSE, XorMM** [16] (polylog efficiency overhead)
Encrypted databases attacks on column-equality leakage (2024)	

5. Comparison

ORAM = technique that hides memory access patterns. [12]

Similarities with StE	Differences with StE
Threat model: Semi-honest	Generic data access instead of structured data
Sublinear (logarithmic) efficiency	The client must keep track of the data structure
Client-Server architecture	Does not leak the access pattern in queries

- ORAM can be used as a component in StE schemes to hide the access pattern and provide protection against inference attacks, by adding a logarithmic overhead to the query complexity. e.g. TWORAM

MPC = method that allow multiple parties to jointly compute a function over their inputs without revealing them. [13]

Similarities with StE	Differences with StE
Threat Model: Semi-honest (sometimes malicious)	Can perform any computation
Many different protocols for many use cases	Does not leak any information
Both can perform Private Set Intersection (PSI)	Client-Server and Distributed architecture

- StE provides the most efficient solution for Updatable PSI, which is a typical MPC problem. MPC remains the best solution for static PSI.

FHE = encryption scheme allowing computation to be directly performed on encrypted data. [11]

Similarities with StE	Differences with StE
Client-Server architecture	Can perform any computation
Non-interactive scheme	Does not leak any information
	Polynomial efficiency overhead

- FHE can be used to perform the same task as StE schemes with no leakage, but with impractical efficiency

TEEs = secure area within a processor ensuring that data and code are protected. [14]

Similarities with StE	Differences with StE
Client-Server architecture (cloud environment)	Can perform any computation
Practical efficiency (near native)	Threat Model: malicious server software
	Hardware-based

6. Conclusion

- StE schemes provide practical efficiency and functionality and is used in real-world DBMS like MongoDB
- StE leaks information to the server, and schemes not providing the most recent security features like forward/backward privacy, volume-hiding and access pattern hiding, can be attacked easily.
- Future research should focus on new attacks and study of the complex leakage profile of StE schemes.

- In contrast with the other techniques, StE schemes do leak some information, to achieve practical efficiency.
- Future research could focus on the creation of a precise benchmark to compare these techniques on practical scenarios.

References

[1] 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, May 2000. ISSN: 1081-6011.
[2] Melissa Chase and Seny Kamara. Structured Encryption and Controlled Disclosure. In Masayuki Abe, editor, Advances in Cryptology - ASIACRYPT 2010, pages577–594, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
[3] Seny Kamara and Tarik Moataz. Sql on structurally-encrypted databases. In Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 26,2018, Proceedings, Part I 24, pages 149–180. Springer,2018.
[4] Seny Kamara, Charalampos Papamanthou, and Tom Roeder. Dynamic searchable symmetric encryption. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12, pages 965–976, New York, NY, USA, 2012. Association for Computing Machinery. event place: Raleigh, North Carolina, USA.
[5] David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, and Michael Steiner. Dynamic Searchable Encryption in Very-Large Databases: Data Structures and Implementation, 2014. Published: Cryptology ePrint Archive, Paper 2014/853.
[6] Seny Kamara and Tarik Moataz. Boolean searchable symmetric encryption with worstcase sublinear complexity. In Advances in Cryptology–EUROCRYPT2017: 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques,Paris, France, April 30 May 4, 2017, Proceedings, Part III 36, pages 94–124. Springer, 2017.
[7] Juntao Gao, Xinxiang Liu, Xuelian Li, and Baocang Wang. Leakage-Suppressed Range Query Scheme for Structured Data in IoT. IEEE Systems Journal,16(3):3531–3542, September 2022.
[8] Geong Sen Poh, Moesfa Soeheila Mohamad, and Muhammad Reza Z'aba. Structured encryption for conceptual graphs. In International Workshop on Security, pages 105–122. Springer, 2012.
[9] 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, pages 79–88, New York,NY, USA, 2006. Association for Computing Machinery. event-place: Alexandria, Virginia, USA.
[10] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-Abuse Attacks Against Searchable Encryption. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15, pages 668–679, New York, NY, USA,2015. Association for Computing Machinery. event-place: Denver, Colorado, USA.
[11] Craig Gentry. A fully homomorphic encryption scheme. Stanford university, 2009
[12] Ling Ren, Xiangyao Yu, Christopher W Fletcher, Marten Van Dijk, and Srinivas Devadas. Design space exploration and optimization of path oblivious ram in secure processors. In Proceedings of the 40th Annual International Symposium on Computer Architecture, pages 571–582, 2013.
[13] Andrew C Yao. Protocols for secure computations. In 23rd annual symposium on foundations of computer science (sfcs 1982), pages 160–164. IEEE, 1982.
[14] Victor Costan and Srinivas Devadas. Intel sgx explained. Cryptology ePrint Archive, 2016.
[15] David Cash and Stefano Tessaro. The locality of searchable symmetric encryption. In Advances in Cryptology–EUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings 33, pages 351–368. Springer, 2014.
[16] Sarvar Patel, Giuseppe Persiano, Kevin Yeo, and MotiYung. Mitigating leakage in secure cloud-hosted datastructures: Volume-hiding for multi-maps via hashing.In Proceedings of the 2019 ACM SIGSAC conferenceon computer and communications security, pages 79–93, 2019.

TU Delft
Author: Dorian Herbiet (d.n.g.herbiet@student.tudelft.nl)
Supervisor: Lilika Markatou (e.a.markatou@tudelft.nl)