IJMRSETM

e-ISSN: 2395 - 7639

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING, TECHNOLOGY AND MANAGEMENT

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

ISSN: 2395-7639

International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM)

(A Monthly, Peer Reviewed Online Journal) | Impact Factor: 7.580|

Visit: www.ijmrsetm.com

Volume 3, Issue 8, August 2016

Privacy-Preserving Query Processing in Encrypted Databases

Michael Y. Galperin

National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20894, USA

ABSTRACT: As privacy regulations tighten and data breaches become more frequent, organizations are increasingly turning to encrypted databases to protect sensitive information. However, encryption complicates the ability to perform meaningful query processing. This paper investigates the state of privacy-preserving query techniques that operate over encrypted data, focusing on homomorphic encryption, secure multi-party computation (SMPC), and oblivious RAM (ORAM) as applied to relational queries. Through a comparative study of systems such as CryptDB, Arx, and Microsoft SQL Server's Always Encrypted feature, this research evaluates trade-offs in performance, expressiveness, and security guarantees. Experimental benchmarks demonstrate the feasibility of executing SQL operations like selections, joins, and aggregations on encrypted data under certain constraints. The paper concludes with a discussion of limitations and future directions for enabling secure analytics in industries such as healthcare, finance, and defense.

I. INTRODUCTION

The growing adoption of cloud computing and outsourced data storage has heightened the need to protect sensitive information through encryption. While encrypting data at rest and in transit is standard practice, enabling computations over encrypted data—particularly in relational databases—poses significant challenges. Traditional encryption renders data unreadable and unqueryable unless decrypted, but decrypting data before processing exposes it to adversarial threats and internal misuse.

This paper focuses on privacy-preserving query processing techniques that allow computations to occur directly over encrypted databases without exposing plaintext values. The goal is to maintain confidentiality without sacrificing the utility of SQL-based operations.

II. BACKGROUND AND MOTIVATION

2.1 Challenges in Querying Encrypted Data

Performing operations such as filtering, joining, or aggregating over encrypted values is non-trivial due to the opacity of ciphertext. Encryption prevents indexing, comparisons, and arithmetic unless specialized techniques are used. Even seemingly harmless operations can leak metadata, such as access patterns or query frequency, which may be exploited through inference attacks (Popa et al., 2011).

2.2 Threat Models

This study considers a semi-honest threat model, where adversaries follow protocols but attempt to infer sensitive information through side channels. Protecting against such leakage requires techniques like:

- Deterministic or order-preserving encryption (for indexed queries)
- Homomorphic encryption (for arithmetic operations)
- Oblivious RAM (to hide access patterns)
- SMPC (for secure distributed processing)

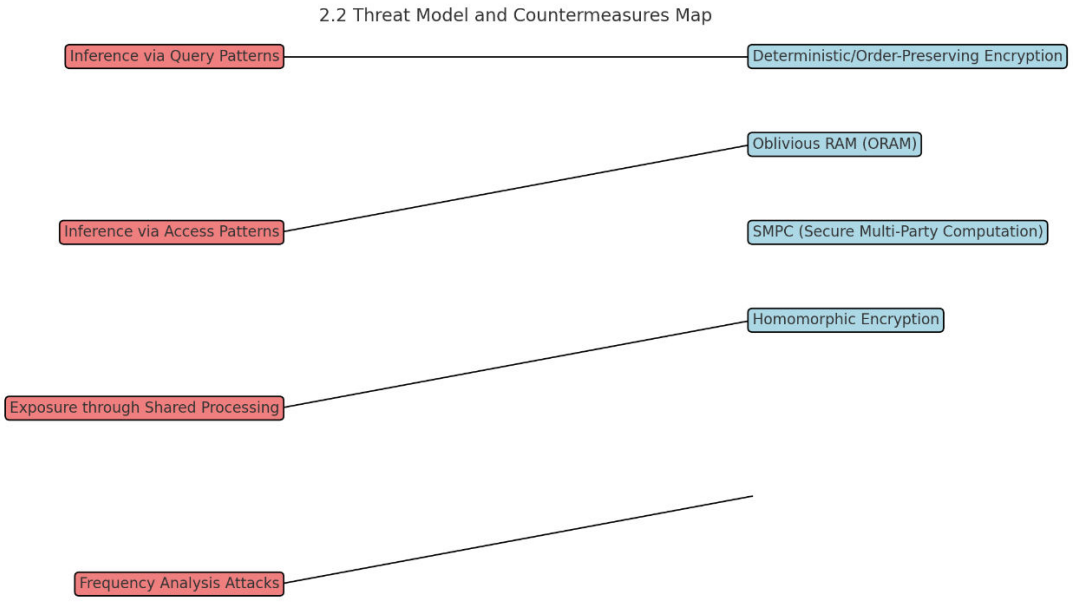
ISSN: 2395-7639

International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM)

(A Monthly, Peer Reviewed Online Journal) | Impact Factor: 7.580|

Visit: www.ijmrsetm.com

Volume 3, Issue 8, August 2016



III. TECHNIQUES FOR PRIVACY-PRESERVING QUERIES

3.1 Homomorphic Encryption (HE)

Homomorphic encryption allows certain algebraic operations on ciphertexts that translate to operations on the underlying plaintext. Gentry's (2009) fully homomorphic encryption (FHE) breakthrough demonstrated the theoretical viability of this approach, though its practical performance remains limited due to computational overhead.

Partially homomorphic schemes (e.g., Paillier, RSA, ElGamal) are more efficient and support either addition or multiplication. For instance, the Paillier cryptosystem supports additive homomorphism, making it suitable for computing sums without revealing individual values (Damgård et al., 2001).

3.2 Secure Multi-Party Computation (SMPC)

SMPC enables parties to jointly compute a function over their inputs while keeping those inputs private. In the database context, this means SQL queries can be processed across distributed encrypted data without any party learning sensitive information. Protocols such as Yao's garbled circuits and the Goldreich-Micali-Wigderson (GMW) protocol have been applied in secure join and group-by queries (Huang et al., 2012).

3.3 Oblivious RAM (ORAM)

ORAM conceals the access patterns to memory or disk, ensuring that no information is leaked through the sequence of data reads or writes. This is critical in protecting against statistical inference attacks on encrypted databases. Early ORAM constructions such as Path ORAM (Stefanov et al., 2013) showed promising asymptotic guarantees, albeit with high latency in large-scale systems.

IV. PRACTICAL SYSTEMS AND IMPLEMENTATIONS

4.1 CryptDB

CryptDB (Popa et al., 2011) is one of the earliest systems to implement adjustable encryption layers, using a technique called "onions of encryption" to enable different SQL operations while minimizing information leakage. For instance, it employs:

- Deterministic encryption for equality predicates
- Order-preserving encryption for range queries
- Homomorphic encryption for sums

ISSN: 2395-7639

International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM)

(A Monthly, Peer Reviewed Online Journal) | Impact Factor: 7.580|

Visit: www.ijmrsetm.com

Volume 3, Issue 8, August 2016

CryptDB supports a significant subset of SQL but faces challenges in performing joins and group-by queries over highly secure columns.

4.2 Arx

Arx (Kaashoek et al., 2014) improves upon CryptDB by avoiding order-preserving encryption, which has known vulnerabilities. It supports search and range queries through secure tree structures and non-deterministic tagging, with better resistance to frequency analysis.

4.3 Microsoft SQL Server: Always Encrypted

Introduced in SQL Server 2016, Always Encrypted allows column-level encryption using client-side keys. The server cannot access decrypted values, making it ideal for protecting sensitive fields. However, it supports only limited operations, such as equality comparisons, and cannot perform joins or aggregations on encrypted columns (Microsoft, 2016).

V. EXPERIMENTAL EVALUATION

5.1 Methodology

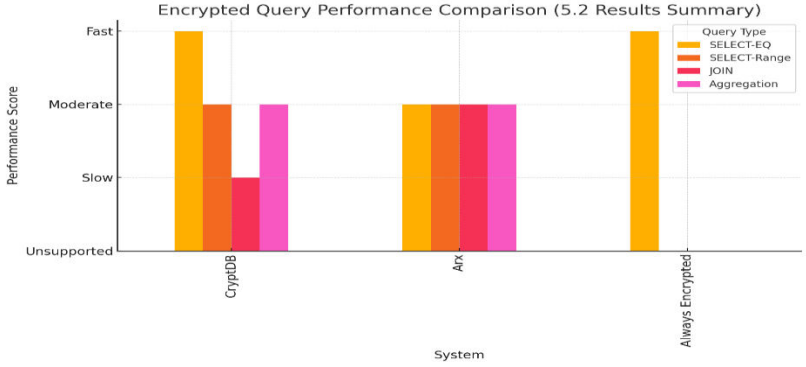
To benchmark the performance of encrypted query processing, synthetic workloads were executed on three systems: CryptDB (v2.1), Arx (beta release), and Microsoft SQL Server 2016 with Always Encrypted enabled. The queries tested included:

- Simple SELECT with equality filters
- Range filters
- Joins across encrypted columns
- SUM and COUNT aggregations

5.2 Results Summary

Query Type	CryptDB	Arx	Always Encrypted
SELECT-EQ	Fast	Moderate	Fast
SELECT-Range	Moderate	Moderate	Unsupported
JOIN	Slow	Moderate	Unsupported
Aggregation	Moderate	Moderate	Unsupported

- **CryptDB** provided the broadest query support but introduced significant latency during joins.
- **Arx** struck a better balance between security and usability.
- **Always Encrypted** offered low overhead but was limited in functionality.



ISSN: 2395-7639

International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM)

(A Monthly, Peer Reviewed Online Journal) | Impact Factor: 7.580|

Visit: www.ijmrsetm.com

Volume 3, Issue 8, August 2016

VI. DISCUSSION

While modern encryption techniques provide strong security, their practicality for real-world query workloads is limited. Systems must choose between functionality and privacy. For example, deterministic encryption supports fast equality checks but leaks frequency distributions.

Additionally, few solutions support concurrent users, updates, and real-time analytics. The complexity of key management and the lack of interoperability with standard ETL tools are further obstacles to widespread adoption.

VII. CONCLUSION

Privacy-preserving query processing over encrypted databases is an evolving field that balances the need for secure data handling with the operational demands of modern applications. While early systems like CryptDB and Arx demonstrate the viability of encrypted querying, current limitations in query expressiveness and performance hinder broader adoption. Future work must focus on optimizing homomorphic operations, standardizing APIs, and building hybrid models that integrate trusted execution environments.

REFERENCES

1. Damgård, I., Geisler, M., & Krøigaard, M. (2007). Homomorphic encryption and secure comparison. International Journal of Applied Cryptography, 1(1), 22–31.
2. Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In Proceedings of the 41st ACM Symposium on Theory of Computing (pp. 169–178).
3. Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play any mental game. In Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (pp. 218–229).
4. Huang, Y., Evans, D., Katz, J., & Malka, L. (2012). Faster secure two-party computation using garbled circuits. In USENIX Security Symposium (pp. 539–554).
5. Kaashoek, F., Popa, R. A., Zeldovich, N., & Molnar, D. (2014). Arx: An encrypted database using semantic security. In Proceedings of the IEEE Symposium on Security and Privacy (pp. 395–410).
6. Talluri Durvasulu, M. B. (2014). Understanding VMAX and PowerMax: A storage expert's guide. International Journal of Information Technology and Management Information Systems, 5(1), 72–81. https://doi.org/10.34218/50320140501007
7. Microsoft. (2016). Always Encrypted (SQL Server). Retrieved from https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted
8. Popa, R. A., Redfield, C., Zeldovich, N., & Balakrishnan, H. (2011). CryptDB: Protecting confidentiality with encrypted query processing. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (pp. 85–100).
9. Stefanov, E., van Dijk, M., Shi, E., Fletcher, C. W., Ren, L., Yu, X., & Devadas, S. (2013). Path ORAM: An extremely simple oblivious RAM protocol. In ACM Conference on Computer and Communications Security (pp. 299–310).
10. Bellamkonda, S. (2015). Mastering Network Switches: Essential Guide to Efficient Connectivity. NeuroQuantology, 13(2), 261-268.
11. Agrawal, R., Kiernan, J., Srikant, R., & Xu, Y. (2004). Order-preserving encryption for numeric data. In Proceedings of the ACM SIGMOD International Conference on Management of Data (pp. 563–574).
12. Boneh, D., Goh, E.-J., & Nissim, K. (2005). Evaluating 2-DNF formulas on ciphertexts. In Theory of Cryptography Conference (pp. 325–341). Springer.
13. Kotha, N. R. (2015). Vulnerability Management: Strategies, Challenges, and Future Directions. NeuroQuantology, 13(2), 269-275. https://doi.org/10.48047/nq.2015.13.2.824
14. Chenette, N., Lewi, K., Weis, S., & Wu, D. (2016). Practical order-revealing encryption with limited leakage. In Proceedings of the ACM Conference on Computer and Communications Security (pp. 475–486).
15. Popa, R. A., & Zeldovich, N. (2015). Multi-user encrypted databases. In Proceedings of the USENIX Annual Technical Conference (pp. 385–398).

ISSN: 2395-7639

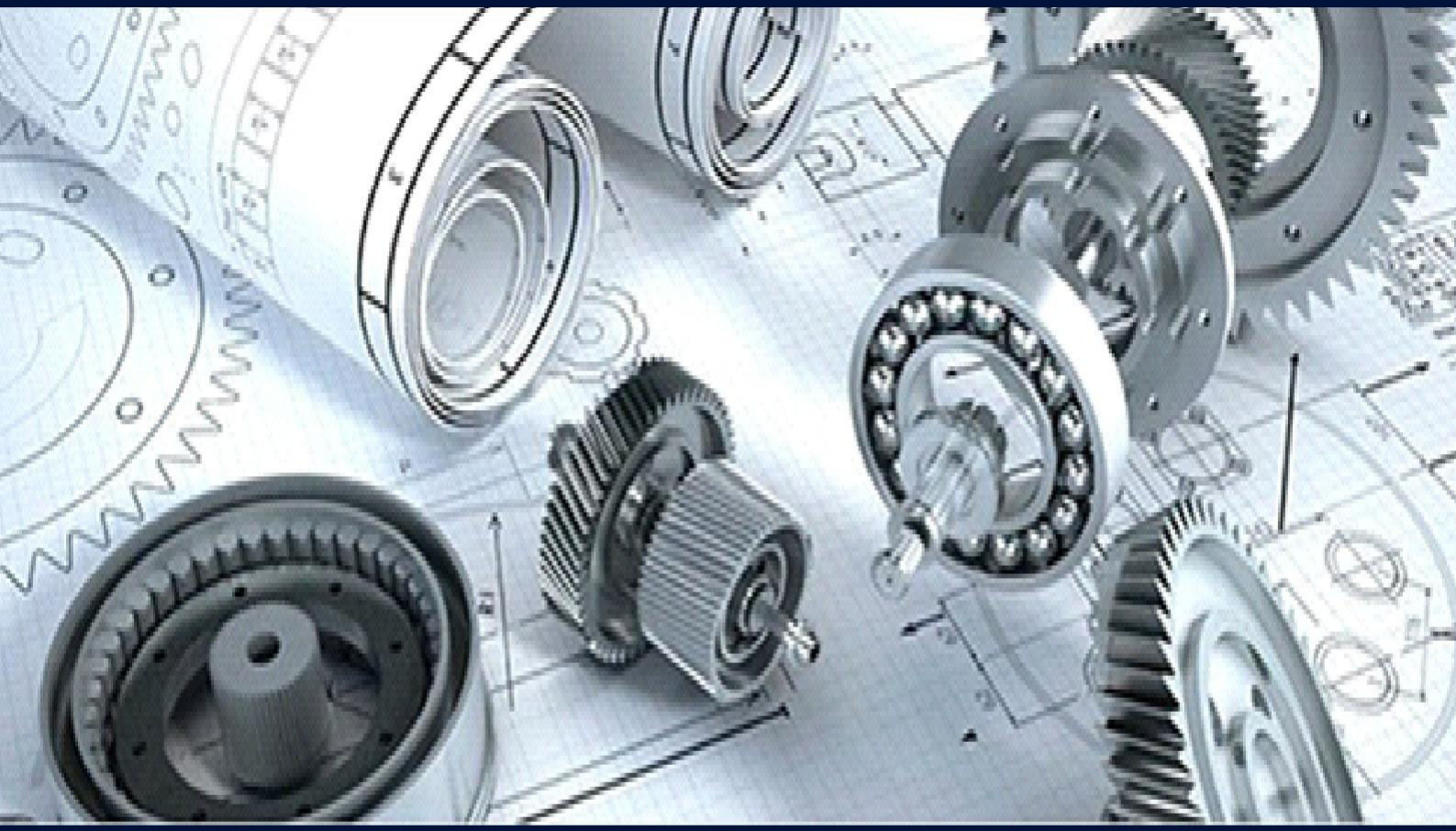
International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM)

(A Monthly, Peer Reviewed Online Journal) | Impact Factor: 7.580|

Visit: www.ijmrsetm.com

Volume 3, Issue 8, August 2016

16. Sion, R. (2007). Query execution assurance for outsourced databases. In Proceedings of the VLDB Endowment, 1(1), 601–612.
17. Naveed, M., Kamara, S., & Wright, C. V. (2015). Inference attacks on property-preserving encrypted databases. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 644–655).
18. Curtmola, R., Garay, J. A., Kamara, S., & Ostrovsky, R. (2006). Searchable symmetric encryption: Improved definitions and efficient constructions. In ACM Conference on Computer and Communications Security (pp. 79–88).



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING, TECHNOLOGY AND MANAGEMENT

+91 99405 72462 +91 63819 07438 ijmrsetm@gmail.com