Measuring the energy consumption and carbon footprint of encrypted databases using CodeCarbon

MIHAI POP, University of Twente, The Netherlands

Privacy and climate change are two crucial subjects that have concerned people recently. Encryption is vital in preventing malicious actors from interfering and tampering with sensitive data. Nonetheless, encrypting and decrypting data impacts the energy consumption of the systems that run those algorithms. This research will measure the carbon footprint of database queries with and without encryption. After gathering the measurements, some statistical analysis is carried out to verify the significant difference between the two and their magnitude. The findings of this paper are pertinent for sustainable data privacy and the cybersecurity domain because they provide additional information in the comparison between searchable encryption and plain, unsecured search queries.

Additional Key Words and Phrases: Databases, Encryption, Privacy, Energy, Carbon Footprint, CodeCarbon

1 INTRODUCTION

Privacy is a topic that has become more prevalent in recent years in our lives. Legislation like GDPR (General Data Protection Regulation) in the European Union or CCPA (California Consumer Privacy Act) in California defines how privacy can be safeguarded and what companies that store personal data need to do to ensure that the rights of their users are not violated.

Encrypted databases protect users' privacy by offering a secure way to query data and reducing the chances of tampering with sensitive information by a malicious actor. However, encrypting and decrypting information takes extra computational time and materializes in more energy consumed, unlike the same task without cryptographic steps. Furthermore, reducing the carbon footprint is a problem companies are trying to solve by evaluating and optimizing electricity assets to combat climate change. As a result, companies need to increase their privacy protection to fight against malevolent actors and simultaneously reduce their carbon emissions. This study aims to investigate these points of interest to understand better the environmental consequences of using a tool like encrypted databases.

Based on the information mentioned above, the study will focus on the following research question: *What is the difference in energy usage and carbon footprint of queries in an encrypted database from that of a non-encrypted database?*

To address these research questions, an experiment was carried out to measure the possible differences between the energy consumption and carbon footprint. The setup of the experiment involved the utilisation of tools like SWiSSSE[5] (a framework that enables encrypted queries to a Redis database) and CodeCarbon[2] (a Python framework to track the carbon emissions of a computer program) to enable the possibility to perform encrypted search queries and to be able to measure the energy consumption and carbon emissions of unencrypted queries and encrypted ones. The analysis carried out on the results of the experiments uncovers noticeable differences in energy consumption and, subsequently, in carbon footprint, encrypted databases requiring more power to perform the same amount of queries.

The remainder of this paper is structured as follows. Section 2 offers an overview of privacy, encrypted databases, and confidentiality. Section 3 outlines the methodology for designing and conducting the experiment. Section 4 showcases the experiment's findings. Section 5 discusses the results and considers potential threats to their validity. Finally, Section 6 offers concluding remarks.

2 RELATED WORK

In this section, we compare existing papers that measure energy consumption and the performance of systems, as well as their techniques to achieve the respective measurements.

Energy consumption. Studies like Warade et al.[12] or Cabrera et al.[1] discuss and analyse the possibilities of measuring energy consumption using EML (energy measurement library) or using hardware like smart energy plugs, that transmit the recorded data. Khan[6] analyses the possibility of measuring the energy consumption of High Performance Computing using Intel RAPL (Running Average Power Limit), a tool that gathers the data directly from the CPU to estimate the energy consumption used. The relationship between energy consumption and encryption measures has been previously investigated to identify if there is a correlation in Machine Learning[8]. Another area where this was researched was in the area of advertisement blockers on mobile devices[10].

Encrypted databases. In Gui et al.[5], the issue of measuring the performance of encrypted databases in comparison with non-encrypted databases was analysed, and it was concluded that it is a performance difference that can be in the range of two times slower for solutions like SWiSSSE in comparison with unencrypted databases.

Carbon footprint. Research has been conducted to determine the carbon footprint of data transmission on a backbone network[4]. However, no research has been done on the impact caused by encrypted databases.

TScIT 42, January 31, 2025, Enschede, The Netherlands

3 MEASUREMENT METHODOLOGY

This section will focus on the experiment setup: the tools used to run the experiment and the environment in which the tools were run to measure the possible differences. The code created for the experiment is open-source and can be found Github[3], so that the results can be reproduced and analysed.

3.1 Measuring the carbon footprint

To record the energy consumption and carbon footprint of the experiment, we used a tool called CodeCarbon. This tool was created by researchers with the goal of registering the aforementioned data. Unlike alternatives, such as measuring the energy drawn by the computer during the experiment, CodeCarbon measures the energy consumed by the CPU, GPU, and memory of the computer separately, helping to understand if there is a specific part that the program is using more. Consequently, it identifies which component is using more power than the rest to run the program. Furthermore, after the energy tracking is stopped, it calculates the carbon emissions for the interval that it ran, relevant for the present study. It does this by utilising the energy consumption value and the regional carbon intensity of electricity, which is based on the ratio of fossil-based and renewable-generated energy in each country.

For the experiment, a wide data range was used to better understand the differences in carbon emissions between different numbers of search queries and to check how the increase in the number of queries affects energy usage and, subsequently, the carbon footprint.

Like any tool used to measure anything, CodeCarbon has some limitations. Since it is a software solution for measuring energy consumption, it relies on the data provided by the computer's components. That data is estimated using different tools, such as Intel Power Gadget for Intel CPUs or an estimation of memory usage. However, the value can differ for each module and manufacturer, so inaccuracies may occur. Furthermore, the tracking is not continuos because CodeCarbon measures the energy usage at a set interval.

All things considered, CodeCarbon is a good tool to track carbon emissions, despite the negative points mentioned above. It is a software-based solution that makes emission monitoring simpler and more convenient without needing specialised equipment; these points are crucial in using it as the monitoring tool for this research.

3.2 Encrypted database

There are multiple DBMSs (Database management systems) like Microsoft SQL[9] or MongoDB[7] that support encrypted queries, but there are limitations. Microsoft SQL supports encrypted queries but only allows the encryption of specific columns, not the entire database. Mongodb has a limited set of features available for the Community Edition, like Automatic Encryption not being available, and there are some limitations to the possible queries that can be used. Furthermore, both options require the database managers to be locked in their respective ecosystems. Either of the aforementioned options is also not open-source and not available for free. For these reasons, neither of these options was selected for the experiment.

For the experiment, a Redis-based implementation called SWiSSSE was used because it provides encryption by using encrypted queries. It is a solution created by security researchers, and a reference benchmark exists between SWiSSSE and a non-encrypted Redis database[5]. Furthermore, even though the implementation is using Redis, the principles can be ported to other DBMSs like PostgreSQL or MariaDB in the future, for example, since the research behind talks about the technique of encrypting the queries, not about providing a dedicated tool only in Redis. There is extensive documentation on how to reproduce the benchmarks, with detailed steps, which was ideal for this research's experiment [11]. On the other hand, SWiSSSE is not a mature solution, and it is a more complex solution to work with than a built-in product like MongoDB's Queryable Encryption. Also, it is a form of Searchable Systematic Encryption, so it is more optimised in keyword-based search scenarios, but in some other scenarios, it might not be applicable altogether.

These points, together with the fact that both SWiSSSE algorithms and the implementation are open source, so other researchers in the cybersecurity domain could reproduce the process, motivated the decision to choose SWiSSSE for this research.

3.3 Materials

Python and libraries like matplotlib and scipy are used to run the CodeCarbon energy tracking and data analysis to identify the statistical significance of the different values. To populate the database, the Enron mail dataset was used since it was linked in the SWiSSSE's Github repository, and using it made the setup of the experiment way easier than reconfiguring the SWiSSSE tool to use another dataset.

3.4 Environment

To make the experiment reproducible and easy to maintain the experiment was run using Docker. A Redis container with the SWiSSSE configuration files was created to run the DBMS.

The experiment ran on the ARM64 architecture on the following machine:

- 16-inch Macbook Pro with M1 Max (10 cores CPU + 24 cores GPU) and 32GB of RAM.

4 RESULTS

This section will address the experiment's results, splitting them based on the research questions. Data was collected using the method described in Section 3. The complete measurements can be found in Appendix A.

4.1 Measuring the energy usage and carbon footprint when the database size is constant and the number of queries is variable

Procedure. We measured the energy consumption and carbon emissions by running 500, 1000, 2000, 3000, 4000 and 5000 random search queries to check for a difference between running these queries using SWiSSSE or in a classic database like Redis. Furthermore, we wanted to identify emissions and energy usage tendencies when scaling the number of queries in both scenarios. The database size was set to 400 000, and the Enron email dataset was used. We used natural logarithm for the figures so that the difference between the classic database and SWiSSSE is more noticeable.

Classic database. As shown in Figure 1, the energy consumption of unencrypted queries is at a maximum ($4.00 \cdot 10^{-5}$ kWh) for running 5000 search queries. Figure 2 shows that the value of the CO_2 emissions is at ($14.98 \cdot 10^{-5}$ kg) for the same amount of search queries. Both figures show that the energy usage and carbon footprint increase slightly but are relatively stable relative to the resource usage of SWiSSSE as a database solution. The high values for both energy usage and carbon emissions at 4000 queries could be due to the fact that a background task was running at the same time with the test, a limitation discussed in Subsection 5.3.

SWiSSSE database. In contrast to the values for a classic Redis database, the energy usage and carbon values for SWiSSSE are consistently higher, reaching a higher peak ($212.89 \cdot 10^{-5}$ kWh) in energy consumption when running 5000 random search queries, showed in Figure 1. Carbon emissions reach a higher point too ($795.52 \cdot 10^{-5}$ kg), 53 times higher than the value recorded for 5000 random search queries for a database that uses unencrypted queries. This difference is shown in Figure 2. Furthermore, in both figures it can be noticed a significant increase as the number of search queries is increased, scaling proportionally, the graph for SWiSSSE being almost parallel to the graph of the classic database.

Statistical analysis. To further analyse the entire data gathered, statistical analysis was used to identify if there is a significant difference between the distributions. For that, the Mann-Whitney U test was performed, with the alpha set at 5% and the following hypotheses used:

- H_0: The distributions of both samples are equal.
- H_1: The distributions of both samples are unequal.

The statistical test results were the following: the p-value of the test was 0.0010, which was lower than the acceptance level of 0.05, thus rejecting the null hypothesis and proving that the distributions of the two scenarios are not the same.

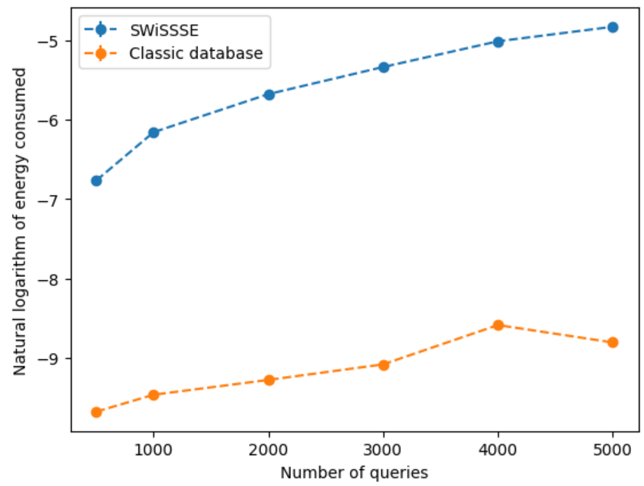

Fig. 1. Energy consumption for variable amount of queries

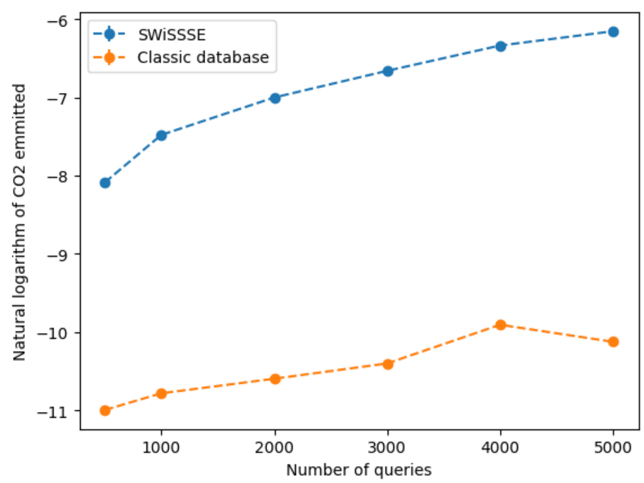

Fig. 2. CO_2 emissions for variable amount of queries

4.2 Measuring energy and carbon costs when the number of queries is constant and database size is variable

Procedure. In contrast to Subsection 4.1, in this scenario, the number of queries used remained at 1000. Still, the database's size changed to 1000, 2000, 3000, 4000, 5000 entries from the Enron email dataset. This measurement aims to identify the differences between the costs per query and how those change depending on the database size. Like the previous subsection, the figures include the standard error for the measurements, but it is not noticeable due to the small changes. Logarithm scale was used also for the figures included in this subsection since they emphasize better the differences.

Classic database. In Figure 3 and Figure 4, it can be observed that the energy usage and carbon emissions are relatively stable, reaching a maximum ($2.90 \cdot 10^{-8}$ kWh in energy usage and $0.78 \cdot 10^{-8}$ kg in carbon emissions) for a database size of 5000 entries.

SWiSSSE database. This type of database had a way more significant and noticeable increase in both energy consumption and CO_2, both of them presented in Figure 3 and Figure 4. However, a similar pattern to the classical database is described by the plots: the values for both energy consumption and for carbon emissions peak at 5000 entries ($37.57 \cdot 10^{-8}$ kWh and $10.05 \cdot 10^{-8}$ kg), using up to 13 times more resources per query than the classical database.

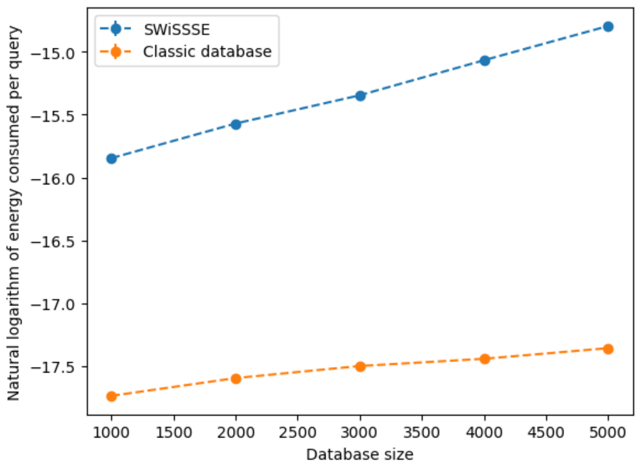

Fig. 3. Energy consumption per query for variable database size

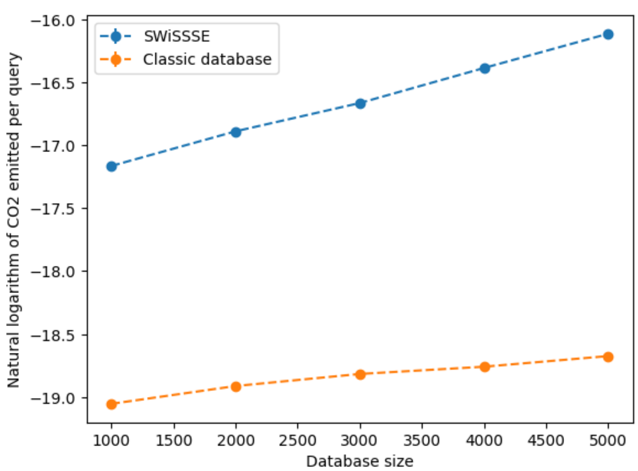

Fig. 4. CO_2 emissions per query for variable database size

5 DISCUSSION

In this section, the experiment's results will be interpreted, compared with previous related data, and discussed, along with the implications of the findings and possible limitations of the experiment.

5.1 Interpretation of the results

Given the data gathered in the experiment and the results of the aforementioned experiment described with plots in Section 4 we can conclude there is a significant difference in energy usage and subsequently, in the value of carbon emissions when encrypted databases like SWiSSSE are used. The resources increase could be attributed to the number of cryptographic operations needed to decrypt and encrypt the data for the queries.

5.2 Implications of the findings

It can be observed in all of the plots that the difference between classic database and SWiSSSE is significant, SWiSSSE being 54 times more energy consuming than a classical database. However, if the respective data is small and not queried often, encrypted databases could be a solution in scenarios that require the data to be encrypted like storing sensitive information. The benefits of better protecting the respective data could outweigh the negative implications of utilising more energy per query and increasing carbon emissions.

However, for large scale deployments, the economical disadvantages of increasing the energy consumption by orders of magnitudes could be a factor that would make the use of encrypted query database like SWiSSSE unpractical. This bottleneck could be why most of the solutions that offer encrypted database solutions rely on encrypting a specific column in a table since that is way less computational intensive and thus, more power efficient and environmentally friendly.

5.3 Limitations of the experiment

SWiSSSE. The experiment is based on the implementation available on GitHub of the SWiSSSE database[11]. This implementation is based on the algorithms on the original paper, but it is not a released or mature solution, so optimisations could be done to reduce the overhead.

CodeCarbon. Since it is a software-based solution for measuring energy consumption and calculating carbon emissions, data-gathering errors could occur way more than hardware-based solutions. Furthermore, this tool monitors the energy consumption of the entire machine, not only the Docker and Redis container processes, in the case of this experiment, which could result in some differences between each iteration due to the background tasks of the operating system or other programs running. Additionally, data is gathered at a set interval, the energy measurements are not continuous, and CodeCarbon doesn't consider any network overhead caused by increasing the sizes of the queries. Moreover, encrypted queries may be larger than classic queries, thus inducing a communication overload. The experiment doesn't measure and consider this potential overload, which is still an important problem in sustainable ICT.

ARM64. A computer with an ARM64 processor was used and the results could be different due to the performance and optimisation differences between the ARM architecture and the X86_64 architecture.

5.4 Future work

Further research in this field, such as generalising this experiment and testing more DBMSs, is needed to identify if the trend is similar across the solutions. Furthermore, experimenting with different datasets could help verify if the results remain consistent for databases that are structured differently: the ones that rely heavily on foreign keys or the ones in which the tables have a minimum amount of foreign keys. Another point of interest could be checking if the energy usage and the carbon footprint are influenced by different architectures. On top of that, more and more companies use Kubernetes to deploy databases and systems since it is easier to manage and allows load balance the requests, so measuring the energy usage and carbon emissions of encrypted databases in that scenario could be valuable information for a broad spectrum of companies and institutions.

6 CONCLUSION

Encrypted databases are a solution that responds to the reality of today, where we need more and more protection for our privacy. However, it is not a perfect solution because it uses significantly more energy resources, and thus, it could heavily influence the carbon footprint of the systems that use them. This difference could be minimized in the future by technological advancements or software optimization, but at the moment there is a significant difference in using a classical database and using an encrypted one.

This paper managed to identify the differences in energy usage and carbon emissions by using a database solution that encrypts queries like SWiSSSE and a solution that doesn't do that. The differences between the two presented in the Figures in Section 4, small but still significant when the dataset or the number of queries is small, increase considerably to up to 53 times and, thus, become substantial for more substantial data and queries. This difference makes the usage of encrypted databases in systems that work with big datasets or are very data-driven very impractical for companies that try to minimise their energy usage and carbon footprint.

REFERENCES

[1] Alberto Cabrera, Francisco Almeida, Javier Arteaga, and Vicente Blanco. 2015. Measuring energy consumption using EML (energy measurement library). *Comput Sci Res Dev* 30, 2 (May 2015), 135–143. https://doi.org/10.1007/s00450-014-0269-5

[2] Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stęchły, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. 2024. *mlco2/codecarbon: v2.4.1.* https://doi.org/10.5281/zenodo.11171501

[3] Marc Damie, Merijn Posthuma, and Mihai Pop. 2025. MarcT0K/privacy-carbon-experiments. https://github.com/MarcT0K/privacy-carbon-experiments

[4] Marion Ficher, Françoise Berthoud, Anne-Laure Ligozat, Patrick Sigonneau, Maxime Wisslé, and Badis Tebbani. 2021. Assessing the Carbon Footprint of the Data Transmission on a Backbone Network. In *24th Conference on Innovation in Clouds, Internet and Networks* (Paris, France). https://hal.science/hal-03196527

[5] Zichen Gui, Kenneth G. Paterson, Sikhar Patranabis, and Bogdan Warinschi. 2020. SWiSSSE: System-Wide Security for Searchable Symmetric Encryption. https://eprint.iacr.org/2020/1328 Publication info: Published elsewhere. Minor revision. PoPETs 2024.

[6] Nizam Kashif Khan. 2018. *Energy Measurement and Modeling in High Performance Computing with Intel's RAPL.* Aalto University. https://aaltodoc.aalto.fi/handle/123456789/30613 ISSN: 1799-4942 (electronic).

[7] MongoDB. 2025. Queryable Encryption - MongoDB Manual v8.0. https://www.mongodb.com/docs/manual/core/queryable-encryption/

[8] Rakshit Naidu, Harshita Diddee, Ajinkya Mulay, Aleti Vardhan, Krithika Ramesh, and Ahmed Zamzam. 2021. Towards Quantifying the Carbon Emissions of Differentially Private Machine Learning. *arXiv* (July 14 2021). http://arxiv.org/abs/2107.06946

[9] Pietervanhove. 2024. Always Encrypted - SQL Server. https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine?view=sql-server-ver16

[10] Kent Rasmussen, Alex Wilson, and Abram Hindle. 2014. Green Mining: Energy Consumption of Advertisement Blocking Methods. In *Proceedings of the 3rd International Workshop on Green and Sustainable Software* (Hyderabad, India). ACM, 38–45. https://doi.org/10.1145/2593743.2593749

[11] SWiSSSE-crypto. 2024. SWiSSSE-crypto/SWiSSSE. https://github.com/SWiSSSE-crypto/SWiSSSE

[12] Mehul Warade, Kevin Lee, Chathurika Ranaweera, and Jean-Guy Schneider. 2023. Monitoring the Energy Consumption of Docker Containers. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, Torino, Italy, 1703–1710. https://doi.org/10.1109/COMPSAC57700.2023.00263

A EXPERIMENT DATA

Country where the carbon emissions were measured: the Netherlands[1]

A.1 Measurements when the number of queries is variable and database size is constant

No. of queries	Energy (10^{-5} kWh)	CO_2 (10^{-5} kg)
500	1.67	6.26
1000	2.07	7.76
2000	2.50	9.34
3000	3.04	11.36
4000	4.99	18.64
5000	4.00	14.98

Table 1. Classic database measurements

No. of queries	Energy (10^{-5} kWh)	CO_2 (10^{-5} kg)
500	30.77	114.99
1000	56.52	211.22
2000	91.25	340.96
3000	128.32	479.51
4000	177.39	662.85
5000	212.89	795.52

Table 2. SWiSSSE database measurements

[1] Carbon emissions values differ per country so the results might be different depending on the geographical location

A.2 Measurements when the database size is variable and number of queries is constant

Database size	Energy (10^{-8} kWh)	CO_2 (10^{-8} kg)
1000	1.98	0.53
2000	2.28	0.61
3000	2.51	0.67
4000	2.66	0.71
5000	2.90	0.78

Table 3. Classic database measurements

Database size	Energy (10^{-8} kWh)	CO_2 (10^{-8} kg)
1000	13.13	3.51
2000	17.30	4.63
3000	21.65	5.79
4000	28.60	7.66
5000	37.57	10.05

Table 4. SWiSSSE database measurements