IMPACT OF DATABASE ENCRYPTION ON WEB APPLICATION PERFORMANCE

Aleksa Vidaković*,
[0009-0005-3527-011X]
Teodor Petrović,
[0009-0008-7186-2552]
Petar Kresoja,
[0009-0008-3343-1540]
Mladen Veinović
[0000-0001-6136-1895]

Singidunum University,
Belgrade, Serbia

Abstract:

This study explores the performance impact of integrating AES-256-GCM encryption into web application operations, examining both read and write processes involving encrypted and non-encrypted user data. After a series of tests aimed to assess the response times for creating and fetching user records, our research reveals that implementation of database encryption introduces modest overhead for read requests, and minimal overhead on write operations. Even under conditions simulating high concurrency and mixed operation loads, the difference in performance remains relatively small. These findings show that encryption can be implemented within web applications with only a slight compromise in performance. This research provides important insights into the practical balance between security and performance in the development of secure web applications.

Keywords:

Database Encryption, Web Application Performance, Application Security, Load Testing, Data Protection.

INTRODUCTION

The integration of encryption into database systems can introduce both solutions and problems in keeping adequate performance while at the same time ensuring data security. As data protection becomes more and more important to organizations, understanding the impact of encryption on performance becomes essential. This paper builds upon research that has explored different encryption strategies and how they affect application performance [1].

The proper implementation of database encryption is deemed essential for protecting sensitive data from unauthorized access when we take into consideration different weaknesses present in many database systems. However, encryption can also degrade application performance [2]. Therefore, the encryption needs to be managed carefully in order to maintain the efficiency of the application. The goal of this paper is to measure the performance impact of integrating AES-256-GCM encryption at the column level and assess whether the security benefits it provides are worth the potential decline in performance [3].

Correspondence:
Aleksa Vidaković

e-mail:
avidakovic@singidunum.ac.rs

DOI: 10.15308/Sinteza-2024-82-87

Recent papers have indicated that while database encryption is critical for securing data at rest, it needs to be carefully planned and implemented to avoid significant drops in performance and to ensure that it complements the existing security measures without becoming a bottleneck [4]. This study aims to analyze the trade-offs between performance and security further, particularly in web applications, in scenarios where throughput and integrity of the data is essential.

Furthermore, the role of encryption in protecting the data against both internal and external threats has been extensively documented, underscoring the importance of robust encryption mechanisms that can be implemented efficiently into the existing database architectures without degrading the efficiency of the applications [5] [6].

By building on the previous work that aimed to analyze different encryption models and their effect on database performance [7], this study looks to provide a comprehensive analysis of the performance of AES-256-GCM algorithm in a simulated web application environment. The goal of this paper is to contribute to the ongoing dialogue on how to balance the aspects of performance and security when working with database systems [8]. Specifically, this study explores how the integration of the aforementioned algorithm affects the throughput of the application, with the aim to provide insights into the operational effects of encryption in high-demand scenarios.

2. METHODOLOGY

This study aims to assess how column-level database encryption affects web application performance, specifically focusing on the response time difference between fetching/creating non-encrypted user records and encrypted user records. For understanding methodology, it is important to address the architecture of the application used for testing, the implementation of the encryption mechanism, the configuration of the testing environment, and the testing procedures that were used.

2.1. APPLICATION ARCHITECTURE

The web application was built using Node.js (v18.16.0) with the Express framework (v4.18.3), which is one of the most popular solutions for building API-s today. The API interacted with a MariaDB database (v10.4.28). The database was hosted on the same physical server with the aim to minimize network latency impact on the application performance.

2.2. ENCRYPTION IMPLEMENTATION

Since MariaDB doesn't support the GCM mode for AES, the encryption was implemented on the application level. This approach allowed the encryption and decryption of potentially sensitive user data to be done before database interaction, using "crypto" Node.js library to handle the AES-256-GCM operations. This setup provided a controlled environment to directly assess the overhead introduced by encryption and decryption processes.

2.3. TESTING ENVIRONMENT

The research was conducted on an on-premises server, equipped with AMD Ryzen 7 5800X processor with 8 cores and 16 threads, clocked at 3.8 GHz (up to 4.7 GHz boost). The server had 32 GB of DDR4 RAM at 3600MHz. The server was equipped with an SSD with a maximum read speed of 3500 MB/s and a maximum write speed of 3300 MB/s.

2.4. TESTING PROCEDURE

2.4.1. Baseline Performance Testing

In order to establish a foundational understanding of the impact of column encryption on web application performance, we started our study by employing baseline performance testing. This phase involved measuring execution times of sequential requests to both retrieve and create user data, using automated tests built with k6 testing tool. The test consisted of 100,000 requests for fetching user data through the "getUserById" and "getEncryptedUserById" functions, and 10,000 requests for creating new users with "addUser" and "addEncryptedUser" functions. This approach was designed to simulate a scenario where individual user interactions occur independently of one another, with the idea of getting a clear picture of the performance overhead introduced by encryption in a controlled environment.

2.4.2. Concurrent Testing

After the baseline testing, the next step was concurrent testing with the aim of evaluating the web application performance under more realistic conditions. Using k6, an open-source load testing tool for web applications, we simulated an environment where the API received up to 100 requests per second over the

course of ten minutes. With this setup, we aimed to simulate the concurrent access patterns that are often found in the production environments, with the idea of testing the system's scalability and efficiency of the encryption mechanism under heavier load. The testing scenarios that were chosen encompass different types of user activities, including both read and write operations, in order to provide insights into the API's responsiveness and throughput when working with encrypted data under high concurrency.

2.4.3. Mixed Load Testing

Following the baseline and concurrent testing phases, we implemented mixed load testing in order to further investigate performance impact under combined operations. Using the k6 testing tool, we conducted tests that performed both read and write operations in order to reflect more dynamic and varied user interactions that are typical for production environments. This phase consisted of two scenarios:

1. Non-Encrypted Scenario: This involved the simultaneous execution of 100 "getUserById" requests and 50 "addUser" requests per second, over the course of ten minutes. The goal was to assess the application's handling of mixed operation types in the absence of encryption.
2. Encrypted Scenario: The test involved 100 "getEncryptedUserById" and 50 "addEncryptedUser" requests per second, also over a ten-minute interval. The aim was to analyze the performance implications introduced by AES-256-GCM encryption when the application processes a mixed load of operations.

3. RESULTS

The examination of the impact of database encryption on web application performance, conducted through baseline, concurrent, and mixed load testing, provided insightful results. These findings outline the effects of encryption on response times for key operations within web applications.

3.1. BASELINE PERFORMANCE TESTING RESULTS

In the baseline performance testing, the goal was to establish a performance benchmark under controlled conditions. We observed a minimal increase in response times for operations involving encrypted data compared to the ones involving non-encrypted data. Specifically, the process for fetching user data ("getUserById" and "getEncryptedUserById" functions), gave the following results: an average response time of 0.29 milliseconds for non-encrypted data, and 0.31 milliseconds for encrypted data. The median response times were closely aligned at approximately 0.50 milliseconds for both types of data. The range of response times, from the minimum to the maximum response time, shows a slightly broader distribution for encrypted data, peaking at 2.80 milliseconds, while for non-encrypted data it peaked at 2.06 milliseconds. This small difference highlights the negligible overhead introduced by decryption in read operations. This suggests a small increase in variability under decryption, that is also visible in the upper percentiles (95^{th}, 99^{th}, and 99.9^{th}), which were marginally higher for encrypted data.

As for the write operations, the results showed a negligible difference in performance, with the average response time for non-encrypted user creation of 48.48 milliseconds, and 48.99 milliseconds for the encrypted user creation. Both scenarios showed very close median and high percentile values.

Table 1. Results of the Baseline Performance Testing.

Operation	Data Type	Average Response Time (ms)	Min (ms)	Median (ms)	Max (ms)	95^{th} Percentile (ms)	99^{th} Percentile (ms)	99.9^{th} Percentile (ms)
Fetch User Data	Non-Encrypted	0.29	0	0.50	2.06	0.66	0.75	1.35
Fetch User Data	Encrypted	0.31	0	0.50	2.80	0.80	0.85	1.55
Create New User	Non-Encrypted	48.82	47.51	48.85	53.42	49.37	49.85	52.39
Create New User	Encrypted	48.99	47.51	49.02	53.21	49.55	50.03	52.40

3.2. CONCURRENT TESTING RESULTS

With the concurrent testing phase, we aimed to assess the application's performance under more realistic, high-concurrency conditions in order to reinforce the findings of the baseline testing and further assess the impact of AES-256-GCM encryption on the response times for both read and write operations. During this stage, the application handled 100 requests per second for read operations, over the course of ten minutes. The average response time for non-encrypted data was approximately 0.16 milliseconds, with a maximum response peaking at 2.18 milliseconds. The 95th, 99th, and 99.9th percentile responses were recorded at 0.63, 0.75, and 1.19 milliseconds, respectively. As for the encrypted data, the average response time was 0.54 milliseconds, with the maximum response time reaching up to 2.56 milliseconds. The 95th, 99th, and 99.9th percentile response times for encrypted data were 0.74, 0.89, and 1.75 milliseconds, respectively.

The write operation concurrent testing involved handling 50 requests per second, over the ten-minute interval. The average response time for non-encrypted data was 50.04 milliseconds, with the maximum response time being 60.59 milliseconds. The 95th, 99th, and 99.9th percentile response times were 50.76, 51.66, and 53.90 milliseconds. Encrypted write operation saw a negligible increase in average response times to 5.13 milliseconds, with a maximum response time of 55.94 milliseconds. The 95th, 99th, and 99.9th percentile response times were 50.88, 51.80, and 53.68 milliseconds.

3.3. MIXED LOAD TESTING RESULTS

The mixed load testing was conducted in order to the performance impact of encryption when the application is subjected to simultaneous read and write operations, simulating an even more realistic scenario. For non-encrypted data, the application was processing 100 requests per second for fetching user data alongside 50 requests per second for user creation over a ten-minute interval. The average response time was 16.91 milliseconds. The median response was 0.51 milliseconds, while the maximum response time recorded was 74.01 milliseconds. The 95th, 99th, and 99.9th percentile response times were 50.73 milliseconds, 51.31 milliseconds, and 52.87 milliseconds, respectively.

For encrypted data, the application handled the same set of operations. The result was an average response time of 17.20 milliseconds. The median response time was 0.64 milliseconds, and the maximum response time was 58.49 milliseconds. The response times at the 95th, 99th, and 99.9th percentiles were 51.13, 51.74, and 53.17 milliseconds.

Table 2. Results of the Concurrent Testing.

Operation	Data Type	Average Response Time (ms)	Min (ms)	Median (ms)	Max (ms)	95th Percentile (ms)	99th Percentile (ms)	99.9th Percentile (ms)
Fetch User Data	Non-Encrypted	0.16	0	0	2.18	0.63	0.75	1.19
Fetch User Data	Encrypted	0.54	0	0.56	2.56	0.74	0.89	1.75
Create New User	Non-Encrypted	50.04	48.44	50.05	60.59	50.76	51.66	53.90
Create New User	Encrypted	50.13	48.28	50.13	55.94	50.88	51.80	53.68

Table 3. Results of the Mixed Load Testing.

Parameter	Non-Encrypted Data	Encrypted Data
Average Response Time (ms)	16.91	17.20
Median Response Time (ms)	0.51	0.64
Maximum Response Time (ms)	74.01	58.49
95th Percentile (ms)	50.73	51.13
99th Percentile (ms)	51.39	51.74
99.9th Percentile (ms)	52.87	53.17

4. ANALYSIS

Through testing conducted across baseline, concurrent, and mixed load scenarios, we collected a lot of data regarding the impact of AES-256-GCM encryption on web application performance. The goal of this analysis is to assess these findings and discuss their imlications for practical and technical aspects of web application development.

4.1. ANALYSIS OF THE BASELINE PERFORMANCE TESTING RESULTS

The results of the baseline performance testing showed that encryption adds minimal overhead to both read and write requests. Encrypted operations had only slightly longer response times compared to non-encrypted operations. This would suggest that the computational cost of implementing AES-256-GCM encryption and decryption, although measurable, is not substantial enough to cause a significant reduction in user experience under controlled conditions.

4.2. ANALYSIS OF THE CONCURRENT TESTING RESULTS

The concurrent testing results further validated our initial findings under conditions that aimed to mimic real-world usage more closely, where multiple users make requests simultaneously. Although the encryption led to somewhat higher response times, the differences, while measurable, were not big enough to raise significant concerns about the efficiency of the application. Write operations showed practically negligible differences in response times for encrypted and non-encrypted operations. However, it is important to note that read operations showed a measurable difference in average response time for encrypted data in contrast to non-encrypted data. To be precise, encrypted read operations took on average 0.54 milliseconds, while non-encrypted read operations took 0.16 milliseconds on average. This distinct increase, though still within a manageable range, illustrates the performance impact of encryption when the application is under a higher load. Despite these differences, the results affirm that the application can handle the added overhead of encryption/decryption, supporting its viability for systems that cannot compromise on data protection.

4.3. ANALYSIS OF THE MIXED LOAD TESTING RESULTS

Through mixed load testing, we aimed to gain insights into the application's behavior when subjected to simultaneous read and write operations. The idea was to reflect a more dynamic interaction typical for production environments. The results showed that the overall impact of encryption when the application is subjected to mixed load operations was in line with the separate read and write operation tests, further illustrating the efficiency of the implemented encryption approach. The minimal difference in performance between non-encrypted and encrypted operations highlights the capability of modern hardware and software to mitigate the performance penalties of encryption.

4.4. IMPLICATIONS FOR WEB APPLICATION DEVELOPMENT AND SECURITY

The findings of this study illustrate several important points for web application developers. Firstly, they confirm that the implementation of AES-256-GCM encryption does not significantly downgrade performance, supporting its use in applications where data security cannot be compromised. Furthermore, the results suggest that modern web applications can implement robust encryption without sacrificing user satisfaction with regard to performance, which is crucial for applications handling sensitive or personal data.

4.5. RECOMMENDATIONS FOR FUTURE RESEARCH

To continue building on the findings of this research, future studies could explore the impact of different encryption algorithms on performance. Exploring the impact of encryption on different database management systems could also provide important information. It could also be beneficial to compare the on-disk encryption at the database level with the application-level encryption.

5. CONCLUSION

In conclusion, this research has shown that AES-256-GCM encryption can be implemented in web applications with minimal performance overheads, even under scenarios involving higher concurrency and mixed operation loads. These findings can be encouraging for developers who aim to enhance their application security without significantly affecting the user experience. It is possible to achieve the balance between performance and security, and the findings of this study provide a foundation for further research and optimization in secure web application development.

6. REFERENCES

[1] E. S. M. Nigm, E.-S. M. El-Rabaie, O. S. Faragallah and A. Mousa, "The Effect of Cryptography Methods on Database Security," in *Annual Conference on Statistics, Computer Sciences and Operation*, Cairo, 2010.

[2] E. Shmueli, R. Vaisenberg, E. Gudes and Y. Elovici, "Implementing a database encryption solution, design and implementation issues," *Computers & Security*, vol. 44, pp. 33-50, 2014.

[3] U. Mattson, "Database Encryption - How to Balance Security with Performance," Protegrity Corp., 2005.

[4] E. Shmueli, R. Vaisenberg, Y. Elovici and C. Glezer, "Database encryption: an overview of contemporary challenges and design considerations," *ACM SIGMOD Record*, vol. 38, no. 3, pp. 29-34, 2010.

[5] U. Maurer, "The Role of Cryptography in Database Security," in *2004 ACM SIGMOD international conference on Management of data*, New York, 2004.

[6] M. Veinović and S. Adamović, Kriptologija I, Belgrade: Singidunum University, 2018.

[7] A. Gomes, C. Santos, C. Wanzeller and P. Martins, "Database Encryption for Balance Between Performance and Security," *Journal of Information Assurance & Cybersecurity*, vol. 2021, 2021.

[8] M. Veinović, G. Šimić, A. Jevremović and M. Tair, Baze podataka, Beograd: Singidunum University, 2022.