Field-Level AES-128 Encryption in Laravel-Based E-Commerce for MSME Data Protection

[1*]Luthfia Afifah, [2]Ir. Ali Nurdin, [3]Ade Silvia Handayani
[1,2,3]Applied Telecommunication Engineering Program,
Department of Electrical Engineering, Politeknik Negeri Sriwijaya, Indonesia
Email: [1]062140352376@student.polsri.ac.id, [2]ali_viking_kps@yahoo.com, [3]ade_silvia@polsri.ac.id

Article Info

Article history:
Received Jun 02nd, 2025
Revised Jul 14th, 2025
Accepted Jul 30th, 2025

Keyword:
AES-128
E-Commerce
Encryption
Laravel
MSMEs

ABSTRACT

The increasing digitization of Micro, Small, and Medium Enterprises (MSMEs) in e-commerce brings critical challenges in protecting customer data. Despite the widespread use of encrypted communication protocols such as HTTPS and TLS for secure data transmission, many MSMEs still fail to implement encryption at the data storage level. This means that once the data reaches the server, it is often stored in unencrypted form within the database. This study implemented AES-128 encryption at the field level in a Laravel-based e-commerce system to protect MSME customer data. The encryption was applied to sensitive data fields and tested through black-box testing and benchmark analysis. A dataset of 10,000 records was used to compare performance between plaintext and encrypted operations. Results showed an average encryption overhead of 0.0409 seconds, indicating minimal impact on performance. The encryption-decryption process consistently returned correct outputs across all trials. This solution offers an affordable and scalable encryption model for MSMEs, enhancing customer data security without relying on external tools or infrastructure.

Corresponding Author:
Luthfia Afifah,
Department of Electrical Engineering, Politeknik Negeri Sriwijaya
Jl. Srijaya Negara, Bukit Lama, Bukit Besar, Palembang City, South Sumatra 30139, Indonesia
Email: 062140352376@student.polsri.ac.id

DOI: http://dx.doi.org/10.24014/ijaidm.v8i2.37675

1. INTRODUCTION

The role of Micro, Small, and Medium Enterprises (MSMEs) is crucial in driving a country's economy. MSMEs play a crucial role in national economic development, significantly contributing to the Gross Domestic Product (GDP) and employment. In Indonesia, MSMEs account for more than 60% of the national GDP and employ approximately 97% of the workforce [1], [2]. To maintain competitiveness, these enterprises have increasingly adopted digital platforms [3]. E-commerce has opened up opportunities for MSMEs to enhance and improve their advertising by providing wider operational distribution for their products or services [4]. The comfort and accessibility of e-commerce allow consumers to shop from anywhere at any time using internet-connected devices [5]. In 2021, e-commerce sales at the state level experienced a significant increase. For example, the UK spent $16,902 million, while the US spent $843.5 million and $27,931 million. This indicates that e-commerce has become a highly influential economic factor worldwide [6]. However, this shift to digital platforms also presents new cybersecurity challenges, particularly in protecting sensitive customer data, such as phone numbers, addresses, and transaction records. Previous studies indicate that most MSMEs still lack adequate data security mechanisms.

Cybersecurity is the most common problem for e-commerce websites. Without proper system protection, reliability, specific requirements, and several communication protocols, customers can suffer losses. According to a report, the e-commerce sector experiences approximately 32.4% of all cyberattacks globally, making it one of the most targeted industries [7]. E-commerce security risks are a significant concern in the competitive online market. Businesses must integrate enhanced security technologies and e-commerce applications to build a safe online shopping platform [8]. With the increasing threat of cyber attacks, it is important to implement comprehensive security measures to protect user data. Many digital payment systems are still vulnerable to cyber attacks due to a lack of strong encryption in the payment process.

Encryption serves as a primary tool in protecting data, ensuring confidentiality, integrity, and privacy. By encrypting data at rest and in transit, organizations can reduce the risk of unauthorized access and eavesdropping by malicious individuals. Encryption strategies also help to meet regulatory requirements by providing mechanisms for secure data storage and transmission [9]. Encryption helps ensure that only authorized parties (such as customers, merchants, and payment gateways) can access the encrypted information. This helps maintain the confidentiality of transaction data [10].

E-commerce needs to utilize AES encryption to protect sensitive customer data, build trust, prevent fraud, and secure online transactions. Previous research has extensively utilized cryptographic encryption techniques to protect online transactions. AES encrypts payment data, such as credit card numbers, bank account information, and other personal details [11]. Advanced Encryption Standard (AES) is a widely trusted symmetric encryption method that balances security and efficiency. According to Zhang et al. Symmetric encryption, like AES, has a very fast computational speed [12]. Implementing strong encryption, such as AES, demonstrates a commitment to protecting customer data security, which can enhance customer trust and brand reputation in a competitive e-commerce market.

The AES is very important for protecting data in web applications. Before AES, the commonly used encryption standard was the Data Encryption Standard (DES). DES used a 56-bit key, which was considered insecure due to the growth in computational power [13]. Previous research has shown that the implementation of AES in e-commerce websites can provide protection against cyber attacks, especially in payment transactions. However, there has been no testing of the AES algorithm implementation in real cases [14]. Lightweight cryptographic techniques such as AES are suitable for web-based applications due to their balance of security and performance This research aims to enhance the security of user data in web applications and to ensure that the data is strongly secured and can only be accessed by parties with the decryption key, in order to help MSMEs to protect their data from cyber threats.

Despite the widespread adoption of transport-layer security mechanisms such as HTTPS and TLS, most MSME e-commerce systems still store sensitive customer data in plaintext once it reaches the database. This practice exposes MSMEs to severe risks in the event of database breaches or insider threats. Furthermore, existing enterprise-grade encryption solutions often rely on external cloud services or costly infrastructure, making them unsuitable for MSMEs. Therefore, this research addresses the practical need for an affordable, application-level encryption mechanism that can be directly integrated into Laravel-based e-commerce systems without compromising performance.

Table 1. Literature Review

	Author (Year)	Issues Raised	Method Used	Research Findings
[9]	Mohammad (2022)	Data encryption in SaaS	Encryption at rest & in transit (AES, TLS/SSL)	Comprehensive overview of strategies but lacks empirical evaluation
[13]	Fadlil et al. (2020)	Secure school top-up transaction	AES-256, XSS testing	AES effectively protects nominal data even under XSS injection
[15]	Goyal et al. (2023)	Importance of encryption algorithms	Comparison of AES, DES, RSA, Blowfish, Twofish	AES plays a key role in modern data security but lacks real-world case studies
[16]	Joshi (2023)	Payment gateway performance	AES-256, TLS, tokenization	AES & serverless Azure reduce transaction latency by 50%
[17]	Komandla (2023)	Fintech cybersecurity	AES-256, RSA, TLS, Zero Trust	AES & Zero Trust enhance protection by 90%, reduce insider threat

In Table 1, Most of the existing studies have emphasized the effectiveness of AES encryption in securing digital systems [Goyal et al., 2023; Mohammad, 2022]. Goyal et al. (2023) highlight the theoretical importance of AES, while Mohammad (2022) elaborates on encryption at various data states (rest/in transit), both lacking practical implementation in MSMEs E-Commerce or Laravel systems.

Asha et al. (2024) present AES-128 in secure messaging with access controls, while Fadlil et al. (2021) validate AES under XSS threats in top-up systems. However, these do not address the performance. Joshi (2023) and Komandla (2023) indicate AES effectiveness in fintech with high protection scores but focus more on cloud environments.

From the literature review, it is evident that AES is not only theoretically good but also practically efficient when implemented properly. This research integrates AES directly into the Laravel application layer and quantitatively measures its impact through structured benchmarking involving 10,000 data entries. Furthermore, it targets MSMEs, a group often overlooked in security research, by proposing a lightweight and affordable encryption mechanism that focuses on its feasibility, security strength, and real-world performance impact. The objective is to ensure customer data confidentiality while maintaining system responsiveness.

This study bridges a critical gap in secure software engineering by integrating AES-128-CBC encryption directly into the Laravel application layer, offering a distinct alternative to database-level encryption. Through validative benchmarking involving 10,000 data points, the research moves beyond theoretical discourse to provide actionable performance metrics in a real-world e-commerce context. Ultimately, the results confirm that this approach minimizes computational overhead while maximizing affordability, establishing it as a viable security standard for MSMEs operating with finite technical resources.

2. RESEARCH METHOD

This research employed an experimental approach to implement and evaluate AES-128-CBC encryption for protecting customer data in a Laravel-based web application system. The design focuses on data security and performance benchmarking using usability testing methods. The methodology includes the following stages of conducting the research:

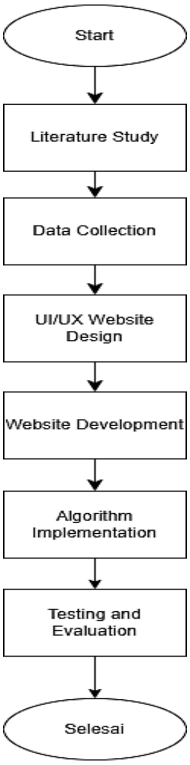


Figure 1. Research Process Design Flowchart

The research methodology followed a structured experimental workflow as illustrated in Figure 1. The flowchart presents a systematic sequence of stages designed to ensure that the implementation of AES-128-CBC encryption and its evaluation are conducted in a controlled and measurable manner. Each stage is interconnected, ranging from theoretical analysis to system validation, to ensure both functional correctness and performance reliability of the proposed encryption approach.

The initial stages focus on preparation and system development. A literature study was conducted to examine existing research on data encryption, e-commerce security, and the application of AES algorithms in web-based systems. This was followed by data collection and UI/UX website design to prepare customer data structures and system interfaces that support secure data input. Website development was then carried out using the Laravel framework, which included database schema design, model–controller implementation, and the integration of encryption logic at the application layer.

The final stages concentrate on algorithm implementation and system evaluation. AES-128-CBC encryption was embedded directly into the Laravel controller to secure sensitive data before it was stored in the database. Testing and evaluation were performed using black-box testing to verify encryption–decryption accuracy and benchmarking techniques to measure performance differences between plaintext and encrypted data operations. This structured methodology ensures that the proposed system is not only secure but also practical for real-world MSME e-commerce applications.

1. Literature Study: The literature study method aims to understand the development of research and the application of the AES algorithm to ensure the security of user data.
2. Data Collection: Collecting dummy customer data for simulation and testing purposes.
3. UI/UX Website Design: Designing the Laravel system interface to support data input and encryption.
4. Website Development: Building Laravel features including models, controllers, and databases.
5. Algorithm Implementation: Implementing the AES-128-CBC algorithm for the encryption and decryption process.
6. Testing and Evaluation: Conducting benchmarking and black-box testing to measure system performance and accuracy.

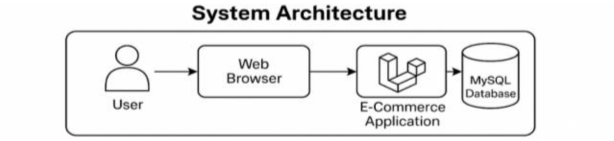


Figure 2. System Architecture

The research employs a comparative benchmark test to assess the performance impact of handling plaintext and encrypted data. The encryption processes were implemented in a Laravel environment to simulate a real-world MSME transaction system, utilizing black-box testing methods to evaluate the functional success of AES-128. The process involved the stages of Figure 3.

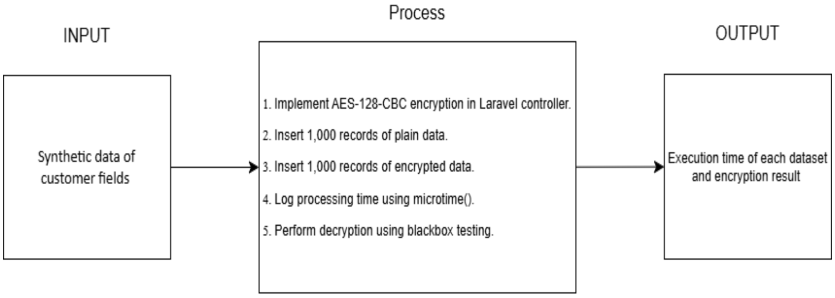

INPUT
Process
OUTPUT
Synthetic data of customer fields
1. Implement AES-128-CBC encryption in Laravel controller.
2. Insert 1,000 records of plain data.
3. Insert 1,000 records of encrypted data.
4. Log processing time using microtime().
5. Perform decryption using blackbox testing.
Execution time of each dataset and encryption result

Figure 3. AES-128-CBC Encryption Workflow

The overall research process is visualized in Figure 1, while the system architecture is depicted in Figure 2. The detailed encryption workflow is shown in Figure 3. AES is a symmetric key block cipher that operates on fixed-size blocks of 128 bits using key sizes of 128, 192, or 256 bits. This research uses AES-128 with Cipher Block Chaining (CBC) mode.

2.1. System Setup and Tools

1. Framework: Laravel 10
2. Language: PHP 8.2
3. Database: MySQL 8.0
4. Encryption Method: AES-128-CBC using PHP’s openssl_encrypt()
5. Key Management: 16-byte encryption key and IV, stored securely via Laravel’s .env configuration
6. Encoding: Encrypted strings were base64-encoded before storage

2.2. Data Collection Techniques

Data collection using Laravel microtime and recording during the testing process. Two sets of data were prepared: one entered in plaintext format, and the other encrypted using AES-128-CBC. Both sets consisted of 1,000 records and were imported into the MySQL 8.0 database. The insertion duration was recorded and averaged over repeated trials to eliminate variance. Customer data was created manually and through Laravel seeder scripts to ensure control over the data content. Encryption was applied during data insertion, while decryption was tested during retrieval.

2.3. Algorithm Structure

The AES encryption process consists of several key stages that enhance the security and complexity of the encrypted data.

1. Key Expansion: The 128-bit key is expanded into 11 round keys.
2. Initial Round:
   a. AddRoundKey: XOR plaintext block with first round key.

3. Rounds (9x):
   a. SubBytes: Substitute bytes using an S-box.
   b. ShiftRows: Shift rows of the state array.
   c. MixColumns: Mix data within columns.
   d. AddRoundKey: XOR with the round key.

4. Final Round (1x):
   a. SubBytes → ShiftRows → AddRoundKey

CBC Mode: Each plaintext block is XOR-ed with the previous ciphertext block before encryption. The first block uses an Initialization Vector (IV). These stages include:

2.3.1. AES Algorithm

At this stage, the AES-128 encryption algorithm with CBC mode is implemented within the Laravel controller (see Figure 4 for the encryption process). This process involves configuring the secret key from the ENV file, generating the IV (Initialization Vector), and merging the encryption results with the IV in base64 format. Functions are used as the core mechanism for encrypting and decrypting data, namely openssl_encrypt() and openssl_decrypt().

Plain Text
AES-128
Chiper Text

Figure 4. Encryption Process

After implementing AES on the database, a comparison test was conducted with 1,000 customer data entries without any prior encryption process. Each storage operation recorded its processing time using the function to allow for comparison with the encrypted version, microtime(). After that, the execution time of the entire storage process (both plain data and encrypted data) was measured using functions before and after the data process. The results of the time differences were recorded and analyzed as part of the performance evaluation of the algorithm, microtime(true). Once the performance test data has been obtained from the data processing, a black box test is conducted to ensure that the data that has been encrypted can be decrypted correctly using the same key.

The AES-128-CBC encryption and decryption processes were implemented at the application layer using PHP and the OpenSSL cryptographic library. A helper class was developed to encapsulate cryptographic operations, including key generation, encryption, and decryption. The encryption key is derived from a user-specific identifier combined with an application-wide secret, ensuring data isolation between users while maintaining consistent cryptographic strength.

```
class EncryptionHelper
{
    private const CIPHER_METHOD = 'AES-128-CBC';

    public static function generateUserKey(int $userId): string
    {
        $secret = env('GLOBAL_SECRET_KEY');
        $rawKey = $userId . '|' . $secret;
        return substr(hash('sha256', $rawKey, true), 0, 16);
    }

    public static function encrypt(string $plaintext, string $key): string
    {
        $ivLength = openssl_cipher_iv_length(self::CIPHER_METHOD);
        $iv = random_bytes($ivLength);

        $ciphertext = openssl_encrypt(
            $plaintext,
            self::CIPHER_METHOD,
            $key,
            OPENSSL_RAW_DATA,
            $iv
        );

        return base64_encode($iv . $ciphertext);
    }

    public static function decrypt(string $encryptedBase64, string $key): string
    {
        $data = base64_decode($encryptedBase64);
        $ivLength = openssl_cipher_iv_length(self::CIPHER_METHOD);

        $iv = substr($data, 0, $ivLength);
        $ciphertext = substr($data, $ivLength);

        return openssl_decrypt(
            $ciphertext,
            self::CIPHER_METHOD,
            $key,
            OPENSSL_RAW_DATA,
            $iv
        );
    }
}
```

During the encryption process, a random Initialization Vector (IV) is generated for each operation to ensure ciphertext uniqueness. The IV is concatenated with the ciphertext and encoded using Base64 before storage in the database. For decryption, the stored data is decoded, separated into IV and ciphertext components, and decrypted using the same key. This approach ensures data confidentiality, integrity, and reproducibility across encryption–decryption cycles.

2.4. Usability Testing (Black-box)

Encryption correctness was validated using a black-box testing method. The encrypted output was retrieved and encrypted before being sent to the database. This testing did not require internal code inspection but validated system behavior through external inputs and outputs.

2.5. Benchmarking Procedures

To measure performance, the system inserted 1,000 plaintext records and 1,000 encrypted records separately. Each test was repeated 10 times, and the average insert time was calculated using microtime logging within controller functions.

2.6. Data Analysis

The collected timing data were processed by calculating the average execution time. Performance overhead was then derived by comparing the results of the encrypted and non-encrypted scenarios. The timing experiments were conducted through repeated insertion trials to ensure measurement reliability. For each test scenario, multiple execution times were recorded under identical system conditions, and the average value was used to represent overall performance. This approach minimizes the influence of transient system fluctuations and background processes. Performance consistency was evaluated by observing time variation trends across repeated trials, which allowed for the identification of stable execution patterns for both plaintext and encrypted data processing.

2.7. Literature Review

Research on e-commerce security has largely focused on the theoretical application of cryptographic algorithms to secure sensitive transactions. Goyal et al. [15] and Mohammad [9] extensively discussed the importance of AES encryption in protecting data at rest and in transit. Their work establishes AES as a robust standard for digital security, emphasizing its role in maintaining data confidentiality. However, their studies are predominantly theoretical, providing comprehensive strategies for encryption but lacking empirical benchmarks regarding the performance overhead these security measures impose on small-scale application servers.

In terms of practical implementation, Fadlil et al. [13] successfully applied AES-256 combined with blockchain technology to secure school payment transactions. Their study demonstrated that AES is effective in protecting sensitive data against specific threats, such as Cross-Site Scripting (XSS). Similarly, Komandla [17] explored advanced cybersecurity strategies in Fintech, proposing a combination of AES encryption and Zero Trust architecture to mitigate insider threats. While these studies validate the security effectiveness of

AES, they focus on high-security environments (fintech and blockchain) and do not address the specific resource constraints faced by MSME platforms built on standard frameworks like Laravel.

Furthermore, recent research by Joshi [16] investigated the use of serverless architecture, specifically Azure Functions, to optimize payment gateways. Joshi's findings suggest that offloading encryption tasks to the cloud can reduce latency and improve scalability. While effective for large enterprises, such cloud-native solutions often introduce complexity and recurring costs that may not be feasible for MSMEs. A significant gap remains in the literature regarding lightweight, monolithic implementations of AES that balance robust security with the limited computational resources typically available to small business e-commerce sites.

Consequently, this research distinguishes itself by focusing on a field-level implementation of AES-128-CBC directly within the Laravel application layer. Unlike the cloud-centric or purely theoretical approaches discussed in prior works, this study aims to provide quantitative performance data (latency and processing time) to demonstrate that high-level security can be achieved in an MSME environment without relying on expensive third-party infrastructure.

3. RESULTS AND ANALYSIS

In this section, the research results are explained, and a comprehensive implementation is also provided. The implementation successfully encrypted sensitive fields using AES-128-CBC before saving to the MySQL database. Black-box testing showed that encrypted fields could be correctly decrypted and matched with the original input when the correct key and IV were supplied.

3.1. Performance testing of plaintext and encrypted

To evaluate the overhead introduced by AES-128-CBC encryption, a series of insert operations were performed using 10,000 records divided into 10 cycles of 1,000 records, both in plain form and encrypted with AES-128-CBC. Each test was conducted under identical conditions using the same hardware, database engine (MySQL 8.0), and Laravel controller functions with microtime(true) for precision timing. After the input process is completed, the time difference is calculated and stored as a benchmark log. This process is repeated ten times to obtain consistent results that can be compared between plain and encrypted data. The following is a table of the results of testing plaintext and encrypted text with a total of 10,000 data points conducted over 10 trials of 1,000 data points.

Table 2. Benchmark Results of 10 Insert Tests (1,000 Records Each)

Test.	Plain Insert Time (s)	Encrypted Insert Time (s)
1	0.1030991077	0.1142511368
2	0.0549149513	0.1136770248
3	0.0703809261	0.1218979359
4	0.0726661682	0.1225900650
5	0.0879871845	0.1086120605
6	0.0736610889	0.0946280956
7	0.0806889534	0.1305539608
8	0.0785751343	0.1202721596
9	0.0673530102	0.1220319271
10	0.0610399246	0.1105461121

The results presented in Table 2 show a consistent pattern in plaintext data insertion time. As the number of inserted records increases, the execution time grows proportionally, indicating a linear performance trend. The relatively stable time increments across repeated tests demonstrate that the database insertion process operates consistently without significant fluctuations. This behavior confirms that, in the absence of cryptographic operations, system performance is primarily influenced by database write operations and input/output processing rather than computational overhead.

Table 3. Benchmark Results of Data Insertion

Record Type	Total Records	Average Time (s)
Plain Data	10,000	0.0750
Encrypted Data	1,000	0.1159

Table 3 illustrates the data insertion performance when AES-128-CBC encryption is applied. Compared to plaintext insertion, the encrypted scenario exhibits a slight increase in execution time due to the additional cryptographic processes involved, including key derivation, Initialization Vector (IV) generation, and encryption computation. However, the results remain consistent across multiple test iterations, with no extreme spikes or anomalies observed. The predictable and uniform delay suggests that the encryption overhead is computationally stable and does not introduce performance instability, making it suitable for small- to medium-scale transactional systems.

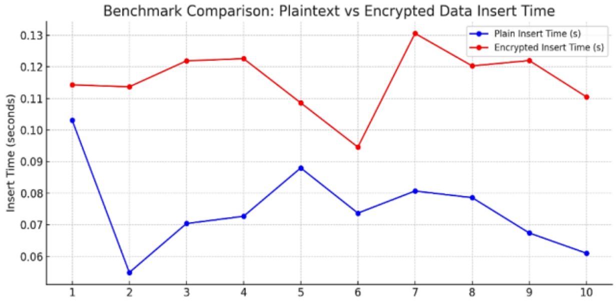


Figure 5. Comparison of Average Insert Times Between Plaintext and Encrypted Records

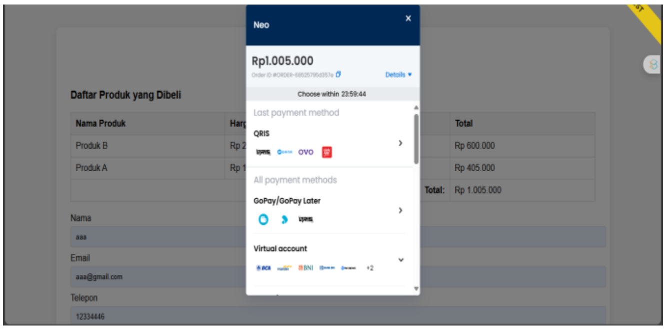
Figure 5 illustrates the comparative benchmark results between plaintext and encrypted data insertions over ten test iterations. Each test involved inserting 1,000 customer records into a MySQL database using Laravel controllers.

The blue line represents the time taken to insert plaintext data, while the red line indicates the time required for AES-128-CBC encrypted data. Across all test iterations, encrypted insertions consistently required more processing time compared to plaintext. However, the performance overhead remained minor, ranging between 0.03–0.06 seconds. The slight delay in processing encrypted data is primarily due to the computational load of AES-128 encryption combined with base64 encoding and random IV generation for each record. Unlike plain inserts, the encrypted records require dynamic encryption at runtime, which increases CPU utilization during the operation.

The variation in processing time is attributed to the encryption function and base64 encoding overhead, both executed on the application layer. Despite this, the graph demonstrates that encryption maintains a stable performance trend, indicating that AES-128-CBC integration does not introduce significant performance volatility or risk of bottleneck in medium-scale applications. These results confirm that field-level encryption is practical for MSMEs and can be adopted without sacrificing system responsiveness, especially in non-high-frequency transaction environments.

3.2. Testing the payment process

Testing on transaction processes involving sensitive customer data is conducted to ensure that the system works properly.



Neo

Rp1.005.000

Order ID #ORDER-68525795d357e

Details

Choose within 23:59:44

Last payment method

QRIS

All payment methods

GoPay/GoPay Later

Virtual account

+2

Daftar Produk yang Dibeli

Nama Produk	Harg	Total
Produk B	Rp 2	Rp 600.000
Produk A	Rp 1	Rp 405.000
	Total:	Rp 1.005.000

Nama

aaa

Email

aaa@gmail.com

Telepon

12334446

Figure 6. Testing the Transaction Process in E-Commerce

The results of testing the system usage in Figure 6 indicate that the encrypted system successfully encrypts sensitive data using AES-128-CBC before storing it in the MySQL database in all trials without any failures, ensuring that customer data remains safe in case of any unwanted incidents such as unauthorized access breaches to the database.

Table 4. Encryption Result

Example	Data Field	Encrypted Value (Base64)
1	Phone Number	6L5UMIlknhFqTN9BCP27ehr102TtBxdj9i+SyjRGH4o=
2	Address	eqkgPQQPRJBLFv/GYCoC0hlJf4nJXnw7xWSiZERNkSTzSamFzzrzTqL7RDGecRt3

Table 4 presents two representative examples of encrypted customer data generated using the proposed AES-128-CBC implementation. The encrypted phone number and address appear as random Base64-encoded ciphertext, demonstrating that the original plaintext is fully obscured. This sample-based presentation provides a concise verification of the encryption output while avoiding unnecessary exposure of multiple data fields.

4. CONCLUSION

The research demonstrates that implementing AES-128 encryption in a Laravel-based system can enhance customer data security without significantly increasing storage overhead. Although the execution time for storing encrypted data is slightly higher than that for plain data, the difference still falls within an acceptable range for medium-scale applications. Therefore, the use of this encryption is suitable for information systems of MSMEs that manage sensitive data, such as customer identities, to prevent data leaks. Benchmarks conducted with 10 tests, each with 1,000 data points, show consistent results and reinforce the finding that the system remains efficient even when equipped with additional security mechanisms. Benchmark results across 10,000 records indicate an average overhead of only 0.0409 seconds per 1,000 encrypted entries, confirming that encryption can be applied efficiently even in resource-constrained MSME environments. Thus, it is hoped that encryption can be implemented at the MSME level. Through black-box validation, the encryption-decryption process was proven to be functionally correct and stable, preserving data integrity without errors across all test iterations. These findings suggest that field-level encryption using AES-128-CBC is both a practical and scalable approach for improving data protection in e-commerce systems operated by small businesses.

This research helps bridge the gap between theoretical encryption models and real-world applications by providing a replicable framework built with Laravel and PHP. It encourages MSMEs to adopt built-in data protection mechanisms that do not depend on third-party services or costly infrastructure. To maintain optimal system performance, further research can focus on comparisons with other encryption algorithms, as well as the integration of encryption with other security methods such as tokenization or file-level encryption, which also deserves further investigation to enhance the overall data protection layer.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Sriwijaya State Polytechnic for providing research facilities and institutional support. Special appreciation is extended to the Undergraduate Program in Applied Telecommunications Engineering for their academic guidance and support throughout the research process. We are also deeply thankful to the Office of Cooperatives and MSMEs of South Sumatra Province for their invaluable partnership and contributions, especially in facilitating access to relevant data, field insights, and local MSME stakeholders essential to the success of this study.

REFERENCES

[1] Eno, “Meningkat Pesat Tahun 2018, Pertumbuhan UMKM di Sumsel Tumbuh Tipis 2019,” Urban Id. [Online]. Available: https://kumparan.com/urbanid/meningkat-pesat-tahun-2018-pertumbuhan-umkm-di-sumsel-tumbuh-tipis-2019-1ssS5z0rxYI

[2] “Laporan Kinerja Kemenkop UKM,” Kementerian Koperasi dan Usaha Kecil dan Menengah Republik Indonesia. [Online]. Available: https://djpb.kemenkeu.go.id/kppn/lubuksikaping/id/data-publikasi/artikel/3134-kontribusi-umkm-dalam-perekonomian-indonesia.html

[3] N. Noerchoidah and N. Nurdina, “Media Website Sebagai Solusi Promosi Penjualan Pada Umkm,” J. Kreat. dan Inov. (Jurnal Kreanova), vol. 2, no. 1, pp. 1–6, 2022, doi: 10.24034/kreanova.v2i1.5212.

[4] Ghada Taher, “E-Commerce: Advantages and Limitations,” Int. J. Acad. Res. Account. Financ. Manag. Sci., vol. 11, no. 2, pp. 202–221, 2021, doi: 10.6007/IJARAFMS.

[5] O. Challenges, “A_Review_of_Blockchains_Role_in_E-Commerce_Transa.pdf,” 2024.

[6] X. Liu et al., “Cyber security threats: A never-ending challenge for e-commerce,” Front. Psychol., vol. 13, no. October, pp. 1–15, 2022, doi: 10.3389/fpsyg.2022.927398.

[7] V. Jain, B. Malviya, and S. Arya, “An Overview of Electronic Commerce (e-Commerce),” J. Contemp. Issues Bus. Gov., vol. 27, no. 3, 2021, doi: 10.47750/cibg.2021.27.03.090.

[8] Z. Wu, S. Shen, H. Zhou, H. Li, C. Lu, and D. Zou, “An effective approach for the protection of user commodity viewing privacy in e-commerce website,” Knowledge-Based Syst., vol. 220, p. 106952, 2021, doi: 10.1016/j.knosys.2021.106952.

[9] N. Mohammad, “Encryption Strategies for Protecting Data in SaaS Applications,” no. March 2022, 2022.

[10] M. A. Hassan, Z. Shukur, and M. K. Hasan, “An efficient secure electronic payment system for e-commerce,” Computers, vol. 9, no. 3, pp. 1–13, 2020, doi: 10.3390/computers9030066.

[11] R. M. Mohammed, “Mitigating Man-in-the-middle Attack In Online Payment System Transaction Using Polymorphic AES Encryption Algorithm,” vol. 14, no. 3, pp. 102–112, 2023.

[12] Q. Zhang, “An Overview and Analysis of Hybrid Encryption: The Combination of Symmetric Encryption and Asymmetric Encryption,” IEEE, pp. 616–622, 2021, doi: 10.1109/CDS52072.2021.00111.

[13] A. Fadlil, I. Riadi, and A. Nugrahantoro, “Data Security for School Service Top-Up Transactions Based on AES Combination Blockchain Technology,” Lontar Komput. J. Ilm. Teknol. Inf., vol. 11, no. 3, p. 155, 2020, doi: 10.24843/lkjiti.2020.v11.i03.p04.

[14] H. Sulaimon, N. O.-F. of N. and Applied, and undefined 2024, “Design and implementation of secured e-commerce digital learning for the educational system in Nigeria,” Fnasjournals.Com, vol. 5, no. 4, pp. 23–32, 2024, [Online]. Available: https://fnasjournals.com/index.php/FNAS-JMSE/article/view/336

[15] P. Goyal, P. Sharma, M. Sharma, and A. Pareek, “The Importance of Data Encryption in Data Security,” J. Nonlinear Anal. Optim., vol. 13, no. 01, pp. 01–11, 2023, doi: 10.36893/jnao.2022.v13i02.001-011.

[16] P. K. Joshi, “Azure Functions in Payment Gateways : A Serverless Approach to Financial Journal of Artificial Intelligence & Cloud Computing Azure Functions in Payment Gateways : A Serverless Approach to Financial Systems,” vol. 2023, no. October, 2024, doi: 10.47363/JAICC/2023(2)390.

[17] V. Komandla, “Safeguarding Digital Finance : Advanced Cybersecurity Strategies for Protecting Customer Data in Fintech Safeguarding Digital Finance : Advanced Cybersecurity Strategies for Protecting Customer Data in Fintech,” no. October, 2024, doi: 10.5281/zenodo.13864693.

BIBLIOGRAPHY OF AUTHORS

Luthfia Afifah is a final-year undergraduate student in the Applied Telecommunications Engineering Program at Politeknik Negeri Sriwijaya. This research is part of her contribution to enhancing the security of digital transactions. Email: 062140352376@student.polsri.ac.id

Ali Nurdin is a senior lecturer at the Department of Electrical Engineering, Politeknik Negeri Sriwijaya. In this research, he provides in-depth guidance related to algorithm analysis and performance evaluation of security systems. Email: ali_viking_kps@yahoo.com

Dr. Ade Silvia Handayani is a lecturer and researcher at the Department of Electrical Engineering, Politeknik Negeri Sriwijaya. Her contribution to this research is invaluable in terms of methodology and technical validation. Email: ade_silvia@polsri.ac.id