Jurnal Informatika Universitas Pamulang **ISSN: 2541-1004**
Penerbit: Fakultas Ilmu Komputer Universitas Pamulang **e-ISSN: 2622-4615**
Vol. 10, No. 1, Maret 2025 (1-6) **https://doi.org/10.32493/informatika.v10i1.44216**

Optimization of Classical Cryptography Security: An Experimental Study on Affine and Vigenere Algorithms in Modern Computing Environments

Edy Prayitno[1*], Basuki Heri Winarno[2], Ani Lestari[3]

[1,2,3]Fakulty of Informatioan Technology, Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia, 55198
e-mail: [1]edyprayitno@utdi.ac.id, [2]bheriw@utdi.ac.id, [3]ani.lestari@students.utdi.ac.id
*Corresponding author

Submitted Date: October 7th, 2024 Reviewed Date: Desember 17th, 2024
Revised Date: Desember 25th, 2024 Accepted Date: February 12th, 2025

Abstract

Data security in modern computing, particularly in IoT and edge environments, demanded lightweight cryptographic solutions due to resource constraints. Classical cryptographic algorithms, such as the Affine Cipher and Vigenere Cipher, were attractive for their simplicity and low computational overhead but suffer from vulnerabilities to modern attacks, including brute force and frequency analysis. This study aimed to optimize these algorithms by increasing key length and applying polyalphabetic variations, enhancing their resilience to attacks while maintaining efficiency. Experimental resulted demonstrate significant improvements in resistance to decryption attempts, with optimized algorithms proving highly resilient even under constrained environments. Despite minor increased in computational resources, the algorithms remain efficient and practical for IoT applications. These findings underscored the potential of optimized classical cryptography as a secure yet resource-efficient alternative for modern computing systems, bridging the gap between robust security and operational feasibility. Future research could explore hybrid algorithms combining classical and modern techniques.

Keywords: Cryptography Optimization, Affine Cipher, Vigenere Cipher, Cryptography Security, Modern Computing

1. Introduction

Data security is a critical issue in modern computing environments, driven by advancements in technologies such as cloud computing, the Internet of Things (IoT), and edge computing. These interconnected systems frequently handle sensitive information, where breaches in security can result in significant economic and operational consequences (Shen, 2020). Ensuring robust cryptographic solutions in such environments is vital, yet challenging, due to the resource constraints of IoT and edge devices.

While modern cryptographic algorithms offer high levels of security, their computational complexity makes them unsuitable for resource-constrained environments (Nugraha, Purnama, & Rizky, 2022). In contrast, classical cryptographic algorithms, such as the Affine Cipher and Vigenere Cipher, are attractive for their simplicity and efficiency. However, these algorithms are highly susceptible to modern attacks, including brute force and frequency analysis, limiting their practical use (Ramadhani & Triani, 2022;Winarno, Prayitno, & Samudra, 2019).

This study addresses these challenges by optimizing the security of the Affine Cipher and Vigenere Cipher through two primary strategies: expanding key lengths to resist brute force attacks and introducing polyalphabetic variations to counter frequency analysis. By evaluating these optimizations in modern computing environments, this research aims to bridge the gap between security and efficiency, offering a viable cryptographic solution for IoT and edge systems.

2. Research Methodology

This study adopted an experimental approach to optimize the security of classical cryptographic algorithms, namely the Affine Cipher and Vigenere Cipher, for modern computing environments. The methodology was structured into four main stages:

ISSN: 2541-1004
e-ISSN: 2622-4615
https://doi.org/10.32493/informatika.v10i1.44216

algorithm implementation, optimization, experimental setup, and performance evaluation.

2.1. Algorithm Implementation

The Affine Cipher and Vigenere Cipher were implemented using Python 3.8, with supplementary libraries to facilitate performance measurement: 1) time library: To record execution time during encryption and decryption. 2) memory_profiler library: To measure memory usage in different stages of the encryption process.

The implementation followed the modular arithmetic principles commonly applied in classical cryptography (TOPALOĞLU, CALP, & TÜRK, 2016). For the Vigenere Cipher, the polyalphabetic substitution was implemented as described by (Hananto, Solehudin, Irawan, & Priyatna, 2019), ensuring compatibility with modern computational frameworks. These implementations served as baselines for comparison with their optimized versions.

2.2. Algorithm Optimization

The optimization process focused on addressing the known weaknesses of these algorithms—vulnerability to brute force and frequency analysis attacks—through the following strategies:

1. Key Length Expansion

For the Affine Cipher, the key space was expanded from 26 to 52 symbols, effectively doubling the possible key combinations. This adjustment increases resistance to brute force attacks, as supported by (Hanafi, Hibban, Zulfikar, & Adhinata, 2021), who demonstrated the impact of increased key space on attack complexity.

2. Polyalphabetic Variation

For the Vigenere Cipher, a polyalphabetic variation was introduced to disrupt predictable patterns in ciphertext. This technique, based on recommendations by (Sabonchi & Akay, 2020b), increased randomness in character distribution, thereby reducing susceptibility to frequency analysis and Kasiski attacks. The effectiveness of such variations has been highlighted in previous studies focusing on enhancing classical cryptographic algorithms (Prihanto & Pakereng, 2020).

2.3. Experimental Setup

The experiments were conducted on a Raspberry Pi 4 Model B, selected as a representative low-powered IoT device, with the following specifications:

- Processor: Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- Memory: 4GB LPDDR4-3200 SDRAM
- Operating System: Raspbian OS

The data used for testing included plain text and text files of varying sizes (1KB, 5KB, and 10KB) to represent typical IoT data loads. These testing scenarios aligned with prior studies that utilize small-scale text files for cryptographic performance evaluation (Mulyani, Kurniadi, & Akbar Musadad, 2021).

2.4. Performance and Security Evaluation

The optimized algorithms were evaluated based on the following metrics:

1. Resilience to Attacks

The algorithms were subjected to three types of cryptographic attacks:

- Brute Force Attack: The time required to exhaustively search all possible keys was recorded before and after optimization. This aligns with findings on the inefficiency of brute force methods for Vigenere Ciphers when the key length was extended (Sabonchi & Akay, 2020a; Hasan et al., 2024).
- Frequency Analysis Attack: The vulnerability of ciphertext patterns to frequency distribution analysis was tested, focusing on the time needed to identify patterns. The significance of polyalphabetic variations in reducing frequency analysis effectiveness was well-documented (Hananto et al., 2019; Setyaningrum, Wijanarto, & Rohmani, 2019).
- Kasiski Attack: The Kasiski method was used to evaluate repetitive cryptograms in the ciphertext and determine key length. Findings on the role of key length in mitigating these attacks support the study's optimization approach (Hananto et al., 2019).

2. Execution Time and Memory Usage

Memory consumption during encryption and decryption was measured using the memory_profiler library to assess the practicality of implementing the optimized algorithms in resource-constrained environments. Similar studies evaluating lightweight cryptographic algorithms in IoT devices provide benchmarks for memory and performance efficiency (Silva, Cunha, Barraca, & Aguiar, 2024; Latif, 2020).

ISSN: 2541-1004
e-ISSN: 2622-4615
https://doi.org/10.32493/informatika.v10i1.44216

3. Result and Discussion

This study evaluated the impact of security optimization on Affine Cipher and Vigenere Cipher in modern computing environments, focusing on three main aspects: security, execution time, and memory usage. The results are presented below, supported by tables and figures for clarity.

3.1. Security Testing Results

The security of the optimized algorithms was evaluated using three types of attacks: brute force, frequency analysis, and the Kasiski method.

Brute Force Attacks:

Expanding the key length from 26 to 52 characters doubled the possible key combinations, significantly increasing resistance to brute force attacks. Before optimization, the Affine Cipher required 15 seconds to crack the ciphertext on plain text with a key length of 26 symbols. After optimization, the time increased to 30 seconds for the same ciphertext with a key length of 52 symbols. Similarly, the Vigenere Cipher showed an increase in cracking time from 60 seconds to over 200 seconds. Table 1 below summarized the results of brute force testing, showing a significant increase in resilience for both algorithms.

Table 1. Results of Brute Force Testing

Algorithm	Data Type	Data Size	Key Length	Cracking Time (seconds)	
				Before Optimization	After Optimization
Affine Cipher	Plain text	100 characters	26 symbols	15	0
Affine Cipher	Text file	1 KB	26 symbols	30	60
Vigenere Cipher	Plain text	100 characters	6 characters	50	150
Vigenere Cipher	Text file	1 KB	6 characters	120	300

Figure 1 illustrates the comparison of brute force cracking times before and after optimization for both algorithms.

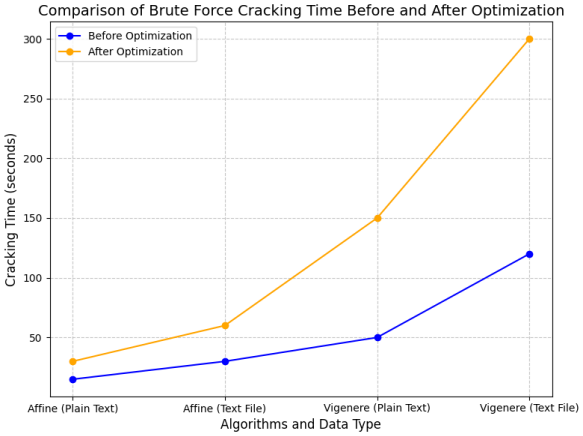

Figure 1. Comparison of Brute Force Cracking Time

Frequency Analysis Attacks:

The introduction of polyalphabetic variations disrupted patterns typically exploited in frequency analysis, increasing decryption time. Before optimization, frequency analysis cracked the Vigenere Cipher in under 180 seconds. After optimization, this time increased to over 300 seconds. Table 2 presented the results of frequency analysis testing, highlighting the improvements in resistance to this type of attack.

Table 2. Results of frequency analysis testing

Algorithm	Data Type	Data Size	Cracking Time (seconds)	
			Before Optimization	After Optimization
Affine Cipher	Plain text	100 characters	5	10
Affine Cipher	Text file	1 KB	25	45
Vigenere Cipher	Plain text	100 characters	40	100
Vigenere Cipher	Text file	1 KB	120	250

3.2. Execution Time

The execution time of the optimized algorithms was evaluated to ensure their practicality in resource-constrained environments. The results indicate a manageable increase in processing time, which remains suitable for IoT devices. For a 5KB text, the encryption time for the Affine Cipher increased from 120ms to 145ms, while decryption time rose from 125ms to 152ms.

ISSN: 2541-1004
e-ISSN: 2622-4615
https://doi.org/10.32493/informatika.v10i1.44216

Similar trends were observed for the Vigenere Cipher.

Table 3 provided a detailed summary of execution time measurements before and after optimization.

Table 3. Execution time measurement results

Algorithm	Data Type	Data Size	Execution Time (milliseconds)	
			Before Optimization	After Optimization
Affine Cipher	Plain text	100 characters	20	25
Affine Cipher	Text file	1 KB	35	40
Vigenere Cipher	Plain text	100 characters	30	40
Vigenere Cipher	Text file	1 KB	70	85

Figure 2 below compared the execution times of encryption and decryption processes for varying text sizes.

Figure 2. Comparison of Execution Time

3.3. Memory Usage

Memory usage was analyzed to determine the feasibility of implementing the optimized algorithms in IoT devices. The results indicated a marginal increase in memory consumption, which remains within acceptable limits.

Before optimization, memory consumption during encryption was 2.4MB, increasing to 2.8MB after optimization. Similarly, memory usage during decryption increased from 2.5MB to 2.9MB.

Table 4 below summarizes memory usage measurements for encryption and decryption processes.

Table 4. Memory usage measurement results

Algorithm	Data Type	Data Size	Memory Usage (KB)	
			Before Optimization	After Optimization
Affine Cipher	Plain text	100 characters	200	220
Affine Cipher	Text file	1 KB	250	280
Vigenere Cipher	Plain text	100 characters	300	350
Vigenere Cipher	Text file	1 KB	350	400

3.4. Implications for IoT and Edge Computing

The study demonstrated the effectiveness of optimized Affine and Vigenere Ciphers for IoT systems, which required both security and resource efficiency. While edge computing systems were not directly tested, the findings suggest potential applications in similar constrained environments.

4. Conclusion

This study aimed to optimize the security of classical cryptographic algorithms, namely the Affine Cipher and Vigenere Cipher, to enhance their applicability in modern computing environments, particularly IoT systems. Through key length expansion and polyalphabetic variations, the algorithms demonstrated significant improvements in resistance to cryptographic attacks, such as brute force, frequency analysis, and the Kasiski method.

The results of this study highlight three key contributions:

1. Improved Security: The optimized algorithms exhibited increased resilience to attacks. For example, the Vigenere Cipher's resistance to brute force attacks increased from 60 seconds

to over 200 seconds, demonstrating the effectiveness of key length expansion and polyalphabetic variation.
2. Sustained Efficiency: Despite the enhancements, the algorithms maintained acceptable levels of execution time and memory usage, making them practical for implementation on resource-constrained IoT devices.
3. Relevance to IoT Systems: The optimized algorithms address the dual challenge of security and resource efficiency, positioning them as viable solutions for lightweight cryptography in IoT environments. While edge computing systems were not directly tested, the findings suggest potential applicability in similar contexts.

This research contributes to the broader effort of adapting classical cryptographic methods to meet the demands of modern, resource-constrained environments. Future studies could explore hybrid approaches, combining classical and modern cryptographic techniques, or evaluate the performance of the optimized algorithms in diverse real-world applications such as edge computing and smart city systems.

References

Hanafi, A. A., Hibban, N., Zulfikar, F. M., & Adhinata, F. D. (2021). Penyelesaian Permainan Sudoku Menggunakan Algoritma Backtracking Berbasis Artificial Intelligence. *Journal ICTEE*, *2*(2), 50. https://doi.org/10.33365/jictee.v2i2.1288

Hananto, A. L., Solehudin, A., Irawan, A. S. Y., & Priyatna, B. (2019). Analyzing the kasiski method against vigenere cipher. *ArXiv*, *6*(6), 1–8.

Hasan, K. F., Simpson, L., Baee, M. A. R., Islam, C., Rahman, Z., Armstrong, W., … McKague, M. (2024). A Framework for Migrating to Post-Quantum Cryptography: Security Dependency Analysis and Case Studies. *IEEE Access*, *12*(December 2023), 23427–23450. https://doi.org/10.1109/ACCESS.2024.3360412

Latif, I. H. (2020). Time Evaluation of Different Cryptography Algorithms Using Labview. *IOP Conference Series: Materials Science and Engineering*, *745*(1). https://doi.org/10.1088/1757-899X/745/1/012039

Mulyani, A., Kurniadi, D., & Akbar Musadad, M. (2021). Rancang Bangun Aplikasi Pengenalan Rukun Islam Sebagai Media Pembelajaran Menggunakan Teknologi Augmented Reality. *Jurnal Algoritma*, *18*(1), 50–61. https://doi.org/10.33364/algoritma/v.18-1.936

Nugraha, G., Purnama, T. A., & Rizky, A. A. (2022). Rancang Bangun Alat Handrub Otomatis Dan Cek Suhu Tubuh Terhubung Ke Telegram Di Puskesmas Sawahlega. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *7*(1), 10–21. https://doi.org/10.29100/jipi.v7i1.2167

Prihanto, D. J. E., & Pakereng, M. I. (2020). Perancangan Teknik Kriptografi Block Cipher Berbasis Pola Tarian Sajojo Papua. *Ultima Computing : Jurnal Sistem Komputer*, *11*(2), 71–80. https://doi.org/10.31937/sk.v11i2.1454

Ramadhani, N. F., & Triani, N. N. A. (2022). Penerapan Teknologi Berbasis Iot (Internet of Things) Dalam Pengumpulan Bukti Audit Di Masa Pandemi Covid-19. *Accounting Global Journal*, *6*(2), 154–169. https://doi.org/10.24176/agj.v6i2.7572

Sabonchi, A. K. S., & Akay, B. (2020a). A binomial crossover based artificial bee colony algorithm for cryptanalysis of polyalphabetic cipher. *Tehnicki Vjesnik*, *27*(6), 1825–1835. https://doi.org/10.17559/TV-20190422225110

Sabonchi, A. K. S., & Akay, B. (2020b). Cryptanalysis of polyalphabetic cipher using differential evolution algorithm. *Tehnicki Vjesnik*, *27*(4), 1101–1107. https://doi.org/10.17559/TV-20190314095054

Setyaningrum, Y. D., Wijanarto, W., & Rohmani, A. (2019). Penerapan Algoritma AES pada Dokumen Penting Yang Disisipkan Dalam Citra Berbasis Algoritma LSB Dan Sobel. *JOINS (Journal of Information System)*, *4*(2), 178–189. https://doi.org/10.33633/joins.v4i2.3099

Shen, Y. (2020). Research on Internet Information Security in the Big Data Era. *E3S Web of Conferences*, *218*, 3–6. https://doi.org/10.1051/e3sconf/202021804008

Silva, C., Cunha, V. A., Barraca, J. P., & Aguiar, R. L. (2024). Analysis of the Cryptographic

Algorithms in IoT Communications. *Information Systems Frontiers*, *26*(4), 1243–1260. https://doi.org/10.1007/s10796-023-10383-9

TOPALOĞLU, N., CALP, M. H., & TÜRK, B. (2016). Bilgi Güvenliği Kapsamında Yeni Bir Veri Şifreleme Algoritması Tasarımı ve Gerçekleştirilmesi. *Bilişim Teknolojileri Dergisi*, *9*(3), 291–301. https://doi.org/10.17671/btd.36875

Winarno, B. H., Prayitno, E., & Samudra, S. T. (2019). Analysis of Easy Perception of Use of Information System Using Technology Acceptance Model Method. *Journal of International Conference Proceedings*, *2*(2), 46–49. https://doi.org/10.32535/jicp.v2i2.601