Router : Jurnal Teknik Informatika dan Terapan
Volume. 3 Nomor. 4 Desember 2025
e-ISSN : 3032-3312; p-ISSN : 3026-3611, Hal. 87-95
DOI: https://doi.org/10.62951/router.v3i4.743
Tersedia: https://journal.aptii.or.id/index.php/Router

Analisi Perbandingan Kinerja Algoritma AES-128 GCM dan AES-256 GCM Pada Protokol TLS 1.2 Di Server Ngnix

Muhammad Arifin Ilham[1*], Dody Herdiana[2], M.Agreindra Helmiawan[3], Asep Saeppani[4]
[1,2,3,4] Informatika, Universitas Sebelas April Sumedang, Indonesia
Email: muhamadarifinilham718@gmail.com [1], dody@unsap.ac.id [2], agreindra@unsap.ac.id [3], saepani@unsap.ac.id [4]*
**Penulis Korespondensi: muhamadarifinilham718@gmail.com*

Abstract *While TLS 1.3 is the latest standard, TLS 1.2 remains widely implemented in many cloud infrastructures. The selection of cipher suites in TLS 1.2, particularly between AES-128-GCM and AES-256-GCM, presents a trade-off between cryptographic strength and system performance. This research aims to analyze the performance comparison of these two algorithms on an Nginx server to determine the optimal configuration for cloud storage services. The study uses a quantitative experimental method by benchmarking two scenarios: (A) Strict (AES-256-GCM), and (B) Balanced (AES-128-GCM). Performance metrics measured include Requests Per Second (RPS), Latency, and Throughput. The results show that handshake performance (RPS and Latency) is nearly identical across all scenarios. However, in large file transfer tests, the AES-128-GCM algorithm (Scenario B) achieved a throughput of 32.4 MB/s, which is 12.5% faster than AES-256-GCM (28.8 MB/s). This study concludes that AES-128-GCM provides the best balance of security and efficiency for data-intensive environments.*

Keywords*: AES; Cloud Storage; Nginx; Throughput; TLS 1.2*

Abstrak. Meskipun TLS 1.3 adalah standar terbaru, TLS 1.2 masih banyak diimplementasikan pada infrastruktur cloud. Pemilihan cipher suite dalam TLS 1.2, khususnya antara AES-128-GCM dan AES-256-GCM, menghadirkan trade-off antara kekuatan kriptografi dan kinerja sistem. Penelitian ini bertujuan untuk menganalisis perbandingan kinerja kedua algoritma tersebut pada server Nginx guna menentukan konfigurasi optimal untuk layanan cloud storage. Penelitian menggunakan metode eksperimental kuantitatif dengan melakukan benchmark pada tiga skenario: (A) Ketat (AES-256-GCM), dan (B) Seimbang (AES-128-GCM). Metrik kinerja yang diukur meliputi Requests Per Second (RPS), Latency, dan Throughput. Hasil penelitian menunjukkan kinerja handshake (RPS dan Latency) hampir identik pada semua skenario. Namun, pada pengujian transfer file besar, algoritma AES-128-GCM (Skenario B) mencapai throughput 32.4 MB/s, atau 12.5% lebih cepat dibandingkan AES-256-GCM (28.8 MB/s). Penelitian ini menyimpulkan bahwa AES-128-GCM memberikan keseimbangan terbaik antara keamanan dan efisiensi untuk lingkungan padat data.

Kata kunci: AES-256; Cloud Storage; Nginx; Throughput; TLS 1.2

1. LATAR BELAKANG

Keamanan data pada layanan cloud storage merupakan aspek krusial, terutama saat data sedang ditransmisikan melalui jaringan publik. Ancaman seperti penyadapan (sniffing) dan serangan Man-in-the-Middle (MITM) menuntut penggunaan protokol enkripsi yang kuat seperti Transport Layer Security (TLS). Meskipun TLS 1.3 telah dirilis(Bhargavan et al., n.d.), protokol TLS 1.2 masih menjadi standar yang dominan digunakan di berbagai infrastruktur server karena kompatibilitasnya yang luas(Ali & Singh, 2023).

Dalam implementasi TLS 1.2 pada web server seperti Nginx, administrator dihadapkan pada pilihan konfigurasi cipher suite. Algoritma enkripsi Advanced Encryption Standard (AES) dengan mode Galois/Counter Mode (GCM) adalah standar industri saat ini. Namun, terdapat variasi panjang kunci(Engineering, 2025), (Subha et al., n.d.), yaitu 128-bit dan 256-bit. Penggunaan AES-256 sering dianggap lebih aman, namun secara teoretis membutuhkan

Naskah Masuk: 18 Oktober 2025; Revisi: 17 November 2025; Diterima: 29 Desember 2025; Terbit: 31 Desember 2025

overhead komputasi yang lebih besar dibandingkan AES-128. Pada layanan cloud storage yang mentransfer data dalam jumlah besar, penurunan kinerja akibat enkripsi yang berat dapat mempengaruhi pengalaman pengguna (QoS)(Utomo & Yasirandi, 2024).

Penelitian terdahulu membahas konfigurasi SSL aman pada Nginx, dampak throughput antara varian AES. membandingkan TLS 1.2 dan 1.3, namun fokus pada protokol MQTT(Nova et al., 2025), bukan transfer file HTTP. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah tersebut dengan melakukan analisis perbandingan kinerja (benchmarking) antara algoritma AES-128-GCM dan AES-256-GCM secara spesifik pada server Nginx. Tujuannya adalah menemukan konfigurasi yang paling optimal yang dapat menyeimbangkan standar keamanan tinggi dengan kecepatan transfer data (throughput) yang maksimal(Erkamim et al., 2024).

2. KAJIAN TEORITIS

Bagian Transport Layer Security (TLS) 1.2 merupakan protokol enkripsi standar yang berfungsi mengamankan transmisi data melalui jaringan publik dengan tujuan mencegah ancaman keamanan seperti penyadapan (sniffing) dan serangan Man-in-the-Middle (MITM). Meskipun versi TLS 1.3 telah diperkenalkan sebagai standar terbaru, faktanya TLS 1.2 masih diimplementasikan secara luas pada berbagai infrastruktur server karena keunggulan kompatibilitasnya yang universal dengan berbagai sistem. Keamanan dalam protokol ini sangat bergantung pada pemilihan cipher suite, di mana Advanced Encryption Standard (AES) dengan mode Galois/Counter Mode (GCM) menjadi standar industri yang paling efisien saat ini.

Dalam implementasinya, algoritma AES memiliki dua varian panjang kunci yang umum digunakan, yaitu AES-128 dan AES-256 (Irwanto, 2024). AES-128 beroperasi dengan panjang kunci 128-bit melalui 10 putaran (rounds) enkripsi, yang secara teoretis memberikan beban komputasi CPU yang lebih ringan. Di sisi lain, AES-256 menggunakan panjang kunci 256-bit dengan 14 putaran enkripsi yang menawarkan tingkat keamanan lebih tinggi, namun konsekuensinya memberikan overhead komputasi yang lebih besar pada server (Eromosele, 2025). Pengaturan prioritas algoritma ini dikendalikan melalui konfigurasi cipher suite pada file nginx.conf di dalam web server Nginx, yang bertanggung jawab atas proses terminasi SSL/TLS serta pengiriman data (Engineering, 2025).

Untuk mengukur efisiensi dari penggunaan protokol keamanan tersebut, penelitian ini menitikberatkan pada tiga metrik kinerja jaringan utama. Metrik pertama adalah Requests Per Second (RPS), yang digunakan untuk mengukur kapasitas server dalam menangani jumlah permintaan per detik selama fase handshake (Herman et al., 2025). Metrik kedua adalah

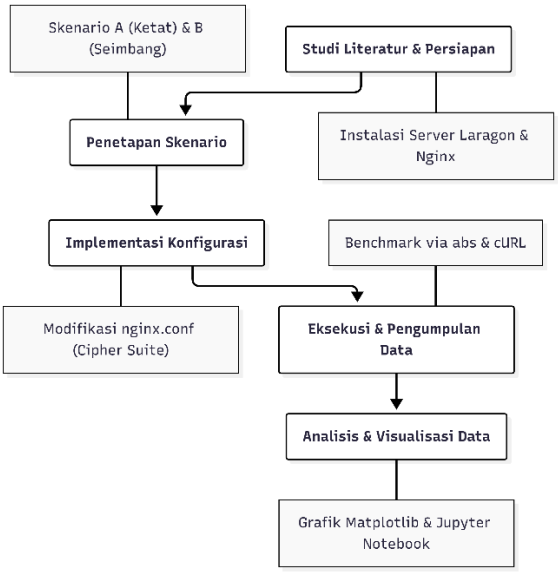
Latency, yang merepresentasikan waktu total yang diperlukan untuk menyelesaikan satu permintaan koneksi(Maharani et al., 2025). Metrik terakhir adalah Throughput, yaitu parameter yang mencatat kecepatan rata-rata transfer data dalam satuan MB/s selama proses enkripsi massal (bulk encryption) berlangsung. Ketiga metrik ini menjadi indikator fundamental dalam menentukan konfigurasi yang paling optimal untuk layanan cloud storage (Erkamim et al., 2024).

3. METODE PENELITIAN

Penelitian ini menggunakan rancangan eksperimental kuantitatif dengan desain pengujian komparatif (benchmarking) untuk mengevaluasi kinerja protokol keamanan TLS 1.2. Populasi dan sampel data dalam penelitian ini adalah transmisi paket data pada layanan penyimpanan awan (cloud storage), yang disimulasikan menggunakan berkas berukuran 100MB untuk mengukur performa enkripsi secara mendalam (Nova et al., 2025). Lingkungan pengujian dikonfigurasi menggunakan prosesor Intel Core i5-6200U dengan RAM 12GB, serta menjalankan web server Nginx v1.27.3 pada platform Laragon (Engineering, 2025) .

Teknik pengumpulan data dilakukan melalui observasi langsung terhadap metrik kinerja jaringan saat proses transmisi data berlangsung. Instrumen yang digunakan meliputi ApacheBench untuk mengukur kapasitas jabat tangan (handshake) server dan alat bantu cURL untuk mengukur kecepatan aliran data(Informasi, 2020). Alat analisis data yang digunakan dalam penelitian ini adalah perangkat lunak berbasis Python melalui Jupyter Notebook yang memanfaatkan pustaka Matplotlib untuk visualisasi data dan NumPy untuk perhitungan statistik metrik kinerja(Petrović, 2024).

Model penelitian ini membandingkan dua skenario konfigurasi cipher suite sebagai variabel utama . Skenario pertama menerapkan tingkat keamanan ketat dengan algoritma AES-256-GCM, sedangkan skenario kedua menggunakan pendekatan seimbang dengan AES-128-GCM(Agustiara et al., 2022). Pengujian validitas instrumen dilakukan dengan melakukan proses running berulang sebanyak sepuluh kali untuk setiap skenario guna memastikan reliabilitas hasil data yang diperoleh, di mana hasil pengujian menunjukkan konsistensi nilai yang stabil dengan deviasi standar yang minimal (Erkamim et al., 2024), Tahapan penelitian secara sistematis dimulai dari persiapan lingkungan hingga visualisasi data sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alur Proses Penelitian.

4. HASIL DAN PEMBAHASAN

Proses pengumpulan data dilakukan melalui serangkaian pengujian eksperimental terkontrol di laboratorium menggunakan server Nginx yang dijalankan di atas platform Laragon. Penelitian dilaksanakan dengan melakukan pengujian kinerja terhadap dua skenario konfigurasi enkripsi untuk mengukur dampak penggunaan panjang kunci AES terhadap performa jaringan. Hasil analisis data disajikan dalam bentuk tabel dan grafik komparatif guna memberikan gambaran yang jelas mengenai kinerja sistem.

Skenario Konfigurasi

Dua skenario konfigurasi cipher suite diterapkan pada file nginx.conf.

a. Skenario (Ketat): Memaksa penggunaan AES-256-GCM. ssl_ciphers 'EECDH+AESGCM:EDH+AESGCM';
b. Skenario (Seimbang): Memprioritaskan penggunaan AES-128-GCM.ssl_ciphers'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:...';

Teknik Pengumpulan Data

Pengumpulan data dilakukan menggunakan dua alat ukur berbasis command line (CLI) yang dijalankan melalui terminal laragon.

a. ApacheBench (abs.exe): Digunakan untuk mengukur kinerja handshake SSL Perintah Parameter yang dicatat untuk Requests Per Second (RPS) dan Time per request (Latency).

   abs -n 100 -c 10 https://NAMA FILE INTERNAL.test:PORT YANG DIPAKAI

Contoh :

abs -n 100 -c 10 https://mycloud.test:8443

b. cURL: Digunakan untuk mengukur kinerja transfer data (bulk encryption). Perintah Parameter yang dicatat adalah kecepatan rata-rata download (Throughput).

curl -k -o NUL https:// NAMA FILE INTERNAL.test: PORT YANG DIPAKAI /NAMA FILE BESERTA FORMAT

Contoh :

abs -n 100 -c 10 https:/mycloud.test:8443I/file-uji-100mb.zip

Hasil Pengujian Data

Perbandingan utama terlihat antara Skenario A dan Skenario B. Pada metrik handshake (RPS dan Latency), tidak terdapat perbedaan signifikan antara AES-256 dan AES-128 (selisih < 1%). Hal ini disebabkan karena beban komputasi terbesar saat handshake terletak pada pertukaran kunci asimetris (RSA/ECDHE), bukan pada enkripsi simetrisnya Namun, perbedaan signifikan ditemukan pada metrik Throughput. Skenario B (AES-128-GCM) mencatatkan kecepatan 32.5 MB/s, sedangkan Skenario A (AES-256-GCM) hanya 28.8 MB/s. Ini menunjukkan peningkatan kinerja sebesar 12.5% saat menggunakan AES-128. Peningkatan ini terjadi karena AES-128 memiliki jumlah putaran (rounds) enkripsi yang lebih sedikit (10 putaran) dibandingkan AES-256 (14 putaran)(Irwanto, 2024),(Eromosele, 2025), sehingga beban CPU lebih ringan saat memproses file besar(Teknologi et al., 2022). Temuan ini sejalan dengan penelitian yang menyatakan efisiensi AES-128 pada perangkat dengan sumber daya terbatas yang dapat kita lihat pada Gambar 2.

Skenario Konfigurasi	Algoritma Enkripsi	RPS (req/sec)	Latency (ms)	Throughput (MB/s)
A (Ketat)	AES-256-GCM	33.05	302.61	28.8
B (Seimbang)	AES-128-GCM	32.69	305.91	32.5

Gambar 2. Skenario A merepresentasikan enkripsi terberat (256-bit) dan Skenario B merepresentasikan enkripsi optimal (128-bit).

Kemudian Hasilnya di Olah menggunakan Jupiter Notebook dengan kode Berfungsi sebagai skrip pemrograman menggunakan bahasa Python dengan pustaka Matplotlib dan NumPy yang digunakan untuk memvisualisasikan hasil pengujian kinerja server. Fungsi utama kode ini adalah mengonversi data mentah (angka hasil benchmark) menjadi Grafik Batang (Bar Chart) yang komparatif antara Skenario A (AES-256) dan Skenario B (AES-128).Skrip ini bekerja dengan mendefinisikan data metrikkinerja (Throughput dan RPS) ke dalam array

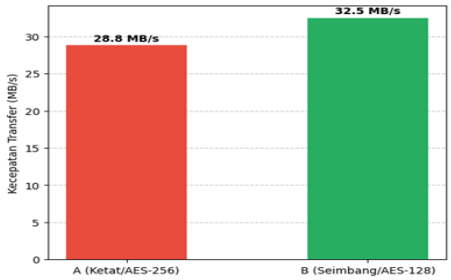
kemudian memetakan data tersebut ke dalam sumbu grafik. Kode ini juga dilengkapi fitur otomatisasi untuk menambahkan label nilai presisi di atas setiap batang grafik dan mengekspor hasil visualisasi menjadi file gambar berkualitas tinggi (300 DPI) agar layak cetak untuk publikasi ilmiah, kode Tertulis Seperti.

```
import matplotlib.pyplot as plt
import numpy as np

scenarios = ['A (Ketat/AES-256)', 'B (Seimbang/AES-128)']
colors = ['#e74c3c', '#27ae60']

throughput_data = [28.8, 32.5]
rps_data        = [33.05, 32.69]
latency_data    = [302.61, 305.91]

def buat_grafik(data, title, ylabel, filename, unit):
plt.figure(figsize=(6, 5))

bars = plt.bar(scenarios, data, color=colors, width=0.5, zorder=3)

plt.title(title, fontsize=12, fontweight='bold', pad=15)
plt.ylabel(ylabel, fontsize=10)

plt.grid(axis='y', linestyle='--', alpha=0.7, zorder=0)

   for bar in bars:
      height = bar.get_height()
      plt.text(bar.get_x() + bar.get_width()/2., height + (0.01 * height),
            f'{height} {unit}',
             ha='center', va='bottom', fontsize=10, fontweight='bold')

   plt.savefig(filename, dpi=300, bbox_inches='tight')
   plt.show()

buat_grafik(throughput_data,
```

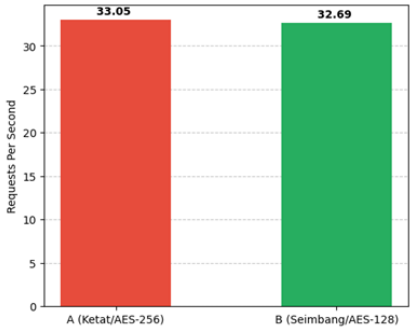
'Perbandingan Throughput (AES-128 Lebih Cepat)',
'Kecepatan Transfer (MB/s)',
'Grafik_Throughput_2Skenario.png', 'MB/s')

buat_grafik(rps_data,
'Perbandingan RPS (Hampir Setara)',
'Requests Per Second',
'Grafik_RPS_2Skenario.png', '')

Pada Gambar 3 dan Gambar 4, ketika kedua grafik dianalisis secara bersamaan, terlihat sebuah pola perilaku sistem yang sangat spesifik. Grafik Requests Per Second (RPS) menunjukkan batang yang tingginya hampir sejajar yaitu 33,05 berbanding 32,69, sedangkan grafik Throughput menunjukkan ketimpangan yang signifikan antara 28,8 MB/s berbanding 32,5 MB/s. Fenomena ini membuktikan bahwa kompleksitas algoritma enkripsi (AES-256 vs AES-128) hanya membebani satu fase komunikasi, yaitu fase transfer data, namun tidak berdampak signifikan pada fase inisiasi koneksi atau handshake. Pada grafik RPS, ketiadaan perbedaan signifikan menunjukkan bahwa proses negosiasi kunci awal (key exchange) membebani CPU dengan porsi yang sama, terlepas dari apakah server nantinya akan menggunakan enkripsi 256-bit atau 128-bit. Sebaliknya, lonjakan kinerja sebesar 12,8% pada grafik Throughput untuk Skenario B mengonfirmasi bahwa pengurangan jumlah putaran (rounds) enkripsi dari 14 putaran pada AES-256 menjadi 10 putaran pada AES-128 secara efektif meringankan beban komputasi server saat memproses aliran data besar. Secara keseluruhan (Eromosele, 2025), kedua grafik ini menyimpulkan bahwa Skenario B (AES-128) adalah konfigurasi yang paling efisien. Ia mampu mempertahankan responsivitas koneksi yang setara dengan konfigurasi ketat, namun menawarkan keunggulan kecepatan transfer data yang jauh lebih tinggi, menjadikannya pilihan ideal untuk layanan berbasis data intensif seperti cloud storage (No et al., 2023).



Gambar 3. Grafik Perbandingan Throughput (AES-128 Lebih Cepat).



Gambar 4. Grafik Perbandingan RPS (Hampir Setara).

5. KESIMPULAN DAN SARAN

KBerdasarkan hasil penelitian, dapat disimpulkan bahwa pemilihan algoritma enkripsi pada TLS 1.2 memiliki dampak langsung terhadap kinerja transfer data. Algoritma AES-128-GCM (Skenario B) terbukti lebih unggul dibandingkan AES-256-GCM (Skenario A) dengan peningkatan throughput sebesar 12.5%, tanpa perbedaan signifikan pada latensi koneksi awal.Oleh karena itu, konfigurasi yang memprioritaskan AES-128-GCM direkomendasikan sebagai solusi optimal untuk infrastruktur cloud storage, karena menawarkan keseimbangan terbaik antara standar keamanan modern yang kuat dan efisiensi kinerja transmisi data, serta keterjagannya karena seimbangnya kinerja dan stres hardware.

UCAPAN TERIMA KASIH

Puji syukur Saya panjatkan ke hadirat Allah SWT atas segala rahmat-Nya sehingga jurnal berjudul “Analisis Perbandingan AES-128 GCM vs AES-256-GCM pada TLS 1.2 di Server Nginx” ini dapat diselesaikan. Saya menyadari bahwa keberhasilan penelitian ini tidak lepas dari dukungan banyak pihak. Terima kasih yang mendalam Saya sampaikan kepada Dosen Pembimbing atas arahan dan kesabarannya dalam membimbing. Apresiasi juga ditujukan kepada seluruh Dosen yang telah membagikan ilmunya, serta kepada para Peneliti di bidang keamanan jaringan yang literatur-literaturnya terkait TLS 1.2 menjadi rujukan utama dalam studi ini. Terakhir, terima kasih kepada rekan-rekan mahasiswa atas kebersamaan, bantuan, dan semangat yang telah diberikan, baik yang terlibat langsung dalam teknis penelitian maupun yang memberikan dukungan moral.

DAFTAR REFERENSI

Agustiara, W., Pratama, A., Junaidi, S., Padang, K., & Barat, S. (2022). Analisis keamanan protokol secure socket layer terhadap serangan packet sniffing pada website portal. *Jurnal Teknologi Informasi dan Komputer*, *6*(1). https://doi.org/10.59697/jtik.v6i1.313

Ali, A., & Singh, V. P. (2023). Comparative analysis of transport layer security (TLS) versions. *International Journal of Computer Applications*. https://doi.org/10.22214/ijraset.2023.57430

Bhargavan, K., Delignat-Lavaud, A., Protzenko, J., Zanella-Béguelin, S., & Zinzindohoué, J. K. (n.d.). *Implementing and proving the TLS 1.3 record layer*.

Engineering, S. (2025). Analysis of Nginx web server performance using IPv6 with load balancing method based on weighted round robin algorithm scheduling. *Journal of Computer Science and Engineering*, *8*(1), 138–147. https://doi.org/10.20895/inista.v7i2.1107

Erkamim, M., Prihatin, T., Saraswati, S. D., & Tonggiroh, M. (2024). Optimalisasi throughput pada penerapan load balancing dalam jaringan cloud menggunakan round robin dan least connection. *Jurnal Sistem Informasi*, *5*(1), 13–23. https://doi.org/10.61628/jsce.v5i1.1056

Eromosele, C. C. (2025). Evaluating the impact of AES-256 encryption on network performance: An analysis of transfer time, latency, and throughput. *Network Security Journal*, *4*(1), 49–58.

Herman, A. M., Wicoksono, W., Panchadria, P. A., Linda, N., & Laela, B. (2025). Performance comparison of NGINX, Apache, and Lighttpd using WRK on a Debian server. *Business and Technology*, *8*(1). https://doi.org/10.32877/bt.v8i1.2661

Informasi, J. T. (2020). Optimasi web server Nginx menggunakan metode reverse proxy di PT Beon Intermedia. *Jurnal Teknologi Informasi dan Teknik Komputer*, *11*(2), 68–73. https://doi.org/10.36382/jti-tki.v11i2.499

Irwanto, D. (2024). File encryption and decryption using algorithm AES-128 bit based website. *Journal of Applied Science*, *4*, 670–677. https://doi.org/10.57152/malcom.v4i2.1305

Maharani, A., Tahir, M., & Buana, D. S. (2025). Penerapan web proxy pada Mikrotik untuk optimalisasi keamanan jaringan wireless LAN di SMKN 1 Bangkalan. *Jurnal Aplikasi Teknologi Informasi*, *9*(4), 6029–6035. https://doi.org/10.36040/jati.v9i4.13950

No, V., Hal, J., Ujung, A. M., Irwan, M., & Nasution, P. (2023). Pentingnya sistem keamanan database untuk melindungi data pribadi. *Jurnal Keamanan Data*, *1*(2), 44–47. https://doi.org/10.47233/jiska.v1i2.929

Nova, M., Br, M., P, A. L., & Sipayung, S. P. (2025). Security analysis of data storage in cloud-based digital archive management systems. *International Journal of Digital Archives*, *2*(3), 80–83. https://doi.org/10.29103/jacka.v2i3.22436

Petrović, T. (2024). Impact of database encryption on web application. *Proceedings of SINTEZA*, 82–87. https://doi.org/10.15308/Sinteza-2024-82-87

Subha, R., Vaijayanth, S., J, S. M., Sanjay, R., & Santhosh, T. (n.d.). Improving data security in cloud computing. *International Journal of Computer Applications*, *256*, 23–27.

Utomo, R. G., & Yasirandi, R. (2024). Exploring trust, privacy, and security in cloud storage adoption among Generation Z: An extended TAM approach. *Journal of Information Security*, *4*(4). https://doi.org/10.22219/kinetik.v9i4.2009