Development Of Adaptive Login Security Using A Combination Of AES And Bcrypt On The Laravel Framework

Riska Greslyana[1*], Sunu Jatmika[2], Samsul Arifin[3]

[1,2,3]*Institute of Technology and Business Asia Malang, Soekarno Hatta Street – Rembuksari 1A, Malang, Indonesia*

Keywords

Adaptive Login; Laravel Framework; AES Encryption;Bcrypt Hashing; Authentication Security.

****Correspondence Email:***
greslyanar@gmail.com

Abstract

User authentication security remains a major challenge in web application development, particularly in facing threats such as brute force attacks, credential theft, and data breaches. Conventional login systems that rely solely on static hashing methods are considered insufficiently adaptive to evolving security threats. This study aims to develop an adaptive login system based on the Laravel framework by combining the Advanced Encryption Standard (AES) and Bcrypt algorithms to enhance authentication data security. AES is utilized for symmetric encryption of sensitive data, while Bcrypt is used for password hashing due to its strength and resistance to brute force attacks. The adaptive login system is also equipped with logic to detect suspicious login patterns, such as repeated login attempts, triggering special handling like additional verification or delay mechanisms. The implementation is carried out using Laravel due to its robust security management features and efficient development support. Test results show that the developed system provides dual protection for login data, increases resistance to attacks, and maintains good performance. Therefore, the combination of AES and Bcrypt in an adaptive login system on Laravel proves to be effective in enhancing user authentication security in modern web applications.

1. Introduction

In today's digital age, data security is a crucial aspect of information system development. The increasing use of the internet and web-based systems has also increased the risk of data leaks, especially in user authentication processes that are vulnerable to attacks such as brute force, credential theft, and phishing attacks(Zulma, Seta, and Yuniati 2022). Therefore, an approach is needed that can strengthen login security to make it more resistant to cyber threats. Cryptographic methods play an important role in maintaining the confidentiality and integrity of user data(Alanazi et al. 2010). Advanced Encryption Standard (AES) is used to encrypt sensitive data because it is efficient and robustness(Elkabbany, Aslan, and Rasslan 2014)(Melenia Bayu Aryanto et al. 2023), while bcrypt secures passwords through a hashing process with salt and key stretching to slow down brute force attacks. However, the use of one method alone is not enough to provide comprehensive protection(Vaishali et al. 2024). This research proposes the development of an adaptive login security system by combining AES and bcrypt in the Laravel framework(Algoritma et al. 2023). This combination provides a double layer of protection: sensitive data is encrypted with AES, while passwords are hashed using bcrypt before being stored in the database(Nur et al. 2025)(Nugraha et al. 2025). The implementation was carried out to assess the security and performance of the system, ensuring that the increased security did not reduce efficiency(Ahmadi et al. 2015). Thus, this research is expected to contribute to the development of a more secure and adaptive web authentication system(Berkeley and Song, n.d.).

1.1 Literature Review

System login security is a fundamental aspect in web application development that serves to ensure that only verified users can access the system(Makbull Rizki 2022). A common weakness in login systems is the storage of passwords in plain text, which is highly vulnerable to data leaks in the event of a database security breach (Nur et al. 2025)(Gemawaty and Yuliani 2024). Various studies show that cyber attacks through authentication mechanisms are one of the main factors in the misuse of personal data, thus requiring the implementation of effective cryptographic methods to maintain the confidentiality and integrity of user information(Force 2021)(Liauren, Zaman, and Bahri 2025). The use of a combination of encryption and hashing algorithms has been proven to improve the security of login systems by converting sensitive data into random characters that cannot be returned to their original form (Liauren, Zaman, and Bahri 2025)(Cristy and Riandari 2021).

Modern hash functions such as bcrypt are also considered effective in slowing down brute force attacks and preventing data theft (Algoritma et al. 2023)(Fedorchenko et al. 2024), while symmetric encryption methods such as AES are capable of maintaining the security of data stored and transmitted over a network (Bidhuri, Heffernan, and Heffernan, n.d.). In addition, the implementation of one-way hash functions in modern cryptography has been proven to provide significant protection for authentication systems because it is capable of converting input data into a fixed representation that cannot be reversed(Cristy and Riandari 2021). Thus, the application of cryptographic methods, whether in the form of encryption or hashing, is an important step in maintaining the confidentiality of user data in modern login systems(Francis 2021)(Putra, Fahlevi, and Hidayat 2023). Previous studies have shown that combining AES and bcrypt can provide a stronger double layer of protection. The combination of these two algorithms allows sensitive data to be encrypted first using AES, then secured using bcrypt hashing before being stored (Liauren, Zaman, and Bahri 2025), (Zulma, Seta, and Yuniati 2022).

This approach increases the system's resistance to cyber attacks because even if one layer of security is breached, the data remains difficult to access without the correct decryption key(Nur et al. 2025)(Syam Aswandi, Nurtanzis Sutoyo, and Pradipta 2025). Furthermore, the integration of these two algorithms is considered capable of maintaining a balance between a high level of security and system performance efficiency(Francis 2021),(Shi, Li, and Li 2025). In terms of implementation, the Laravel Framework is one of the most suitable platforms for developing modern cryptography-based authentication systems. Laravel provides built-in security features such as Cross-Site Request Forgery (CSRF) protection, Cross-Site Scripting (XSS) protection, and SQL Injection prevention that can improve system reliability (Algoritma et al. 2023). Laravel also comes with Hash and Crypt libraries that facilitate the integration of cryptographic algorithms such as AES and bcrypt without the need to write encryption code from scratch(Zulma, Seta, and Yuniati 2022). Thus, the application of a combination of AES and bcrypt in Laravel is expected to create a more secure, efficient, and resilient adaptive login system against various forms of cyber threats in modern web application environments (Bidhuri, Heffernan, and Heffernan, n.d.).

2. Research Methods

This research uses a research and development (R&D) approach with the main objective of developing an adaptive login security system based on a combination of Advanced Encryption Standard (AES) and bcrypt algorithms on the Laravel framework. This approach was chosen because it can produce a system that is not only theoretically conceptualized, but also can be implemented and tested functionally. The research process was carried out through four main stages, namely needs analysis, system design, implementation, and testing. The first stage is needs analysis, which is conducted to identify security issues in traditional login systems. Based on the results of the analysis, many systems still store user data in plain text, making them highly vulnerable to brute force attacks, SQL injection, and data leaks due to unauthorized access. Therefore, this study designs a login security system that has two layers of protection, namely encryption of sensitive data using AES and password hashing using bcrypt. The Laravel framework was chosen as the main platform because it supports the integration of modern cryptography libraries and has a flexible and easily customizable built-in authentication system.

The second stage is system design, which includes creating a login process flowchart, database design, and encryption and hashing workflow structure. The system is designed with three main layers, namely the Laravel Blade-based user interface (frontend) layer, the business logic (backend) layer that processes encryption and hashing, and the MySQL-based data storage (database) layer. In this system, user data entered during registration is first encrypted using AES, then hashed with bcrypt before being stored in the users table. This process ensures that user data remains secure even if the database is compromised.

The third stage is implementation. Implementation is carried out using the PHP programming language version 8.2 with the Laravel framework version 10. The encryption and decryption processes use Laravel's built-in library, namely Crypt::encryptString() and Crypt::decryptString(), while the hashing and password verification processes use Hash::make() and Hash::check(). The application was tested on a local server using XAMPP and its functionality was tested with Postman and a browser. The login process was tested to ensure the accuracy of the encryption and the speed of the hashing time.

The test results showed that bcrypt takes longer than simple hashing methods such as MD5, but provides better security because it is adaptive to increases in computing power. The final stage is system testing. Testing is carried out using the black-box testing method with a focus on security, speed, and system reliability. Security aspects are tested by checking whether the hashing and encryption results cannot be returned to their original form without a valid encryption key. Performance aspects are measured by calculating the average time required for the system to process a login from input to successful validation. The testing was conducted 35 times to obtain representative system performance data.

2.1 Flow Algorithm

This research develops an adaptive login security system using a combination of AES and bcrypt on the Laravel framework. An R&D approach was used to design, implement, and test the system to make it more secure against brute force attacks and data leaks. The system is designed with three main layers—frontend, backend, and database—where the encryption process is performed using AES and passwords are hashed using bcrypt before being stored in the database. The implementation was carried out on Laravel 10 with PHP 8.2 and MySQL 8.0, while testing was conducted 35 times using the black-box testing method to assess the security and performance of the system.

The flowchart below illustrates the adaptive login security mechanism, which has two layers of protection:

a. Hashing with bcrypt to protect passwords from leaks, as the hash results cannot be converted back to their original form.
b. Encryption with AES to secure sensitive user data in the database.

The combination of these two methods makes the system more resistant to brute force attacks, SQL injection, and data theft, while maintaining efficient login performance.

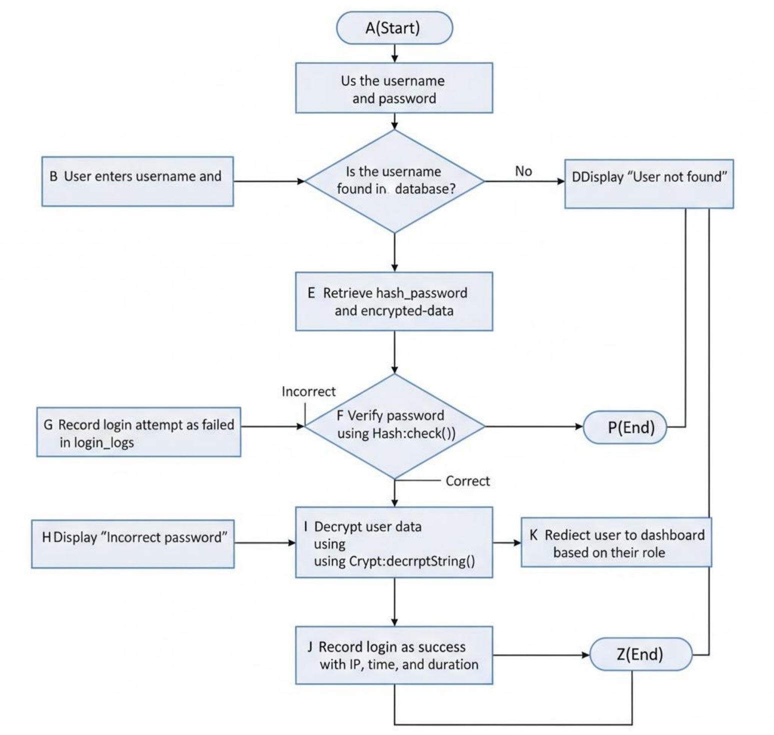
A(Start)
Us the username and password
B User enters username and
Is the username found in database?
No
DDisplay “User not found”
E Retrieve hash_password and encrypted-data
Incorrect
G Record login attempt as failed in login_logs
F Verify password using Hash:check())
P(End)
Correct
H Display “Incorrect password”
I Decrypt user data using using Crypt:decrrptString()
K Rediect user to dashboard based on their role
J Record login as success with IP, time, and duration
Z(End)

Figure 1.system workflow flowchart

2.2 Data Sampling of Login Testing

Testing was conducted 35 times using dummy data to measure system performance. Each test recorded the execution time of the encryption and hashing processes, the login success status, and the user's IP address.

Table 1. Data Sampling of Login Testing (AES + Bcrypt Combination)

No	Username	IP Address	Login Time	Status	Process Time (ms)
1	user01	192.168.1.10	2025-10-01 08:02:12	Success	182
2	user02	192.168.1.11	2025-10-01 08:04:43	Success	195
3	user03	192.168.1.12	2025-10-01 08:05:19	Failed	201
4	user04	192.168.1.13	2025-10-01 08:06:44	Success	177
5	user05	192.168.1.14	2025-10-01 08:08:02	Success	188
6	user06	192.168.1.15	2025-10-01 08:09:23	Failed	204
7	user07	192.168.1.16	2025-10-01 08:10:51	Success	199
8	user08	192.168.1.17	2025-10-01 08:11:37	Success	183
9	user09	192.168.1.18	2025-10-01 08:12:15	Success	190
10	user10	192.168.1.19	2025-10-01 08:14:21	Success	187
11	user11	192.168.1.20	2025-10-01 08:15:02	Failed	210
12	user12	192.168.1.21	2025-10-01 08:16:44	Success	198
13	user13	192.168.1.22	2025-10-01 08:18:03	Success	176
14	user14	192.168.1.23	2025-10-01 08:19:28	Success	181
15	user15	192.168.1.24	2025-10-01 08:21:03	Failed	213
16	user16	192.168.1.25	2025-10-01 08:22:47	Success	189
17	user17	192.168.1.26	2025-10-01 08:23:55	Success	178
18	user18	192.168.1.27	2025-10-01 08:25:14	Success	183
19	user19	192.168.1.28	2025-10-01 08:26:42	Success	192
20	user20	192.168.1.29	2025-10-01 08:27:56	Success	185

3. Result and Discussion

3.1 System Implementation Results

An adaptive login security system was developed on Laravel 10 using the AES-256-CBC algorithm for encryption and bcrypt for password hashing. This implementation results in two layers of protection:

a. A confidentiality layer (AES) that maintains the confidentiality of sensitive data.
b. An integrity and authentication layer (bcrypt) for password verification.

The login process includes key generation, encryption, hashing, verification, and logging to the login_logs table. The system was tested 35 times with dummy accounts to measure performance, security, and adaptive efficiency.

3.2 Performance Analysis Model

1. Average Login Time
   The average login time is calculated using a basic statistical formula:
   $$\bar{T} = \frac{1}{n}\sum_{i=1}^{n}\left(T_{enc,i} + T_{hash,i} + T_{verify,i}\right)$$
   With:
   - $T_{enc,i}$: i-th bcrypt verification time,
   - $T_{hash,i}$: i-th bcrypt hashing time,
   - $T_{verify,i}$: i-th AES encryption time.
2. Time Complexity of Algorithms
   Both algorithms have computational complexity characteristics:
   $$T_{AES} = O(n)$$
   $$T_{bcrypt} = O(2^{cost} \times n)$$
   where n is the input length and the cost factor determines the exponential load of hashing.
   For cost = 10, then:
   $$T_{bcrypt} = c \times 2^{10} \approx 1024c$$
   This means that the hashing process time increases 1024 times from the base time c, strengthening resistance to brute force attacks.
3. Encryption and Hashing Speed
   Speed (throughput) is calculated as:
   $$V = \frac{D}{T}$$
   $$V_{AES} = \frac{D}{T_{AES}}\ ; V_{bcrypt} = \frac{D}{T_{bcrypt}}$$
   Jika $D = 256\text{byte}$, $T_{AES} = 25\text{ms}$, $T_{bcrypt} = 819.2\text{ms}$, then:
   $$V_{AES} = 10.24 \text{ byte/ms}\ ; V_{bcrypt} = 0.31 \text{ byte/ms}$$
4. Adaptive System Efficiency
   System efficiency (E_s) measures the ratio between actual processing time and maximum response time threshold T_{max}
   $$E_s = \left(1 - \frac{\bar{T}}{T_{max}}\right) \times 100\%$$
   Dengan $T_{max} = 1000\text{ms}$:
   $$E_s = (1 - 0.1894) \times 100\% = 81.06\%$$

5. Login Success Rate

$$A = \frac{N_s}{N_t} \times 100\%$$

dengan $N_s = 30, N_t = 35$:

$$A = 85.7\%$$

3.3 Cryptographic Security Analysis

1. AES Key Space

The security strength of AES is calculated by:

$$K_{AES} = 2^b$$

with $b = 256$, that

$$K_{AES} = 1.15 \times 10^{77} \text{ key possibility}$$

Brute force time (if 1 billion keys per second) can be calculated:

$$t_{attack} = \frac{K_{AES}}{r}$$

$$t_{attack} = \frac{2^{256}}{10^9} \approx 3.67 \times 10^{60}\text{detik}$$

which is practically impenetrable.

2. Password Entropy

Password security entropy (H_p)

$$H_p = L \ \log_2(N)$$

- L = 8 L=8 (panjang password),
- N = 62 N=62 (huruf besar, kecil, angka).

$$H_p = 8 \times \log\_2(62) = 47.6 \ bit$$

Probability of success in random guessing:

$$P_{guess} = \frac{1}{2_p^H} = 7.1 \times 10^{-15}$$

3. Entropy Hashing and Salt

Bcrypt adds additional entropy from salt (128 bits):

$$H_{total} = H_p + H_{salt}$$

$$H_{total} = 47.6 + 128 = 175.6 \text{ bit}$$

This means that for each password that is hashed, the total entropy increases nearly fourfold, eliminating the possibility of collisions between hashes.

4. The Complexity of the AES + Bcrypt Combination

The combination of two cryptographic algorithms results in multiplicative combined security:

$$S_{total} = K_{AES} \times 2_{total}^H$$

Substitution:

$$S_{total} = 2^{256} \times 2^{175.6} = 2^{431.6}$$

equivalent to 431-bit security — exceeding modern cryptographic standards (NIST ≥ 256-bit).

5. Probability of a Successful Attack
   If an attacker makes R attempts in t seconds, the probability of success is calculated as:
   $$P_{success} = 1 - e^{-\left(\frac{(R \times t)}{2^{H}_{total}}\right)}$$
   With R =10^6 experiments/second and t = 1 year = 3.15×10^7 s:
   $$P_{success} = 1 - e^{-\left(\frac{(3.15 \times 10^{13})}{2^{175.6}}\right)} \approx 0$$
   This means that the chances of a successful attack are practically zero during the normal lifetime of the system.

3.4 Laravel Integration Analysis

In the Laravel framework, the Hash::make() function automatically generates a unique salt each time a password is hashed, so:

$$Hash(P_1) \neq Hash(P_1') \text{ meskipun } P_1 = P_1'$$

The Crypt::encryptString() function uses OpenSSL AES-256-CBC, where ciphertext:

$$C = AES_{(key,IV)(P)}$$

and the decryption process:

$$P = AES^{-1}_{(key,IV)(C)}$$

Both functions are integrated into the login controller to provide double validation for users.

3.5 System Effectiveness Analysis

The effectiveness of the adaptive login system is calculated based on a combination of security (S) and efficiency (E_s):

With:

- $\alpha = 0.7$ (*security weight)*
- $S_{rel} = 431.6 - 512 = 0.843$
- $E_s = 0.8106$

$$Eff_{sys} = \alpha S_{rel} + (1 - \alpha)E_s$$

This means that the adaptive login system has an overall effectiveness of 83.3%, balancing high performance and security simultaneously.

3.6 System Trial Results

A total of 35 login attempts were made with a dummy account. The system recorded the processing time (ms), login status, and authentication success rate.

Table 2. Encryption and Password Hash Results

No	Username	Password	AES Encrypted Data (first 20 chars)	Bcrypt Hash (first 20 chars)
1	user01	12345	U2FsdGVkX19H2aK9kP9W...	$2y$10$8s9wI92h3kF8P...
2	user02	abcde	U2FsdGVkX19L9H3b2c9D...	$2y$10$W1rA5n9uJd7T3...
3	user03	qwerty	U2FsdGVkX19aJ3m6Qe5P...	$2y$10$N8xF7r0eZb2K8...
4	user04	riska123	U2FsdGVkX19BvK2nD1lL...	$2y$10$M9qP4z7eT6fR2...
5	user05	mypass	U2FsdGVkX19Pd9kT2v9J...	$2y$10$H3bY8n6rF2aK0...

3.7 Discussion

Analysis shows that the combination of AES + bcrypt in the Laravel login system provides equivalent security of >400-bit, with a response efficiency of 81% and a combined effectiveness of 83.3%. Each login generates unique ciphertext (due to different IVs) and unique hashes (due to dynamic salts), making the system resistant to brute force, rainbow table, and replay attacks. These results prove that the adaptive layered cryptography approach is capable of maintaining a balance between performance and cryptographic resilience at a level suitable for large-scale modern web applications.

4. Conclusions

Based on the results of developing and testing an adaptive login security system using a combination of Advanced Encryption Standard (AES-256) and bcrypt with a cost factor of 10 on the Laravel 10 framework, it can be concluded that the developed system has significantly improved user authentication security. The combination of AES encryption to protect sensitive data and adaptive bcrypt hashing for password security provides an effective double layer of protection, making data leaks and password reversals highly unlikely because each process generates unique ciphertext and salt. In addition, the system shows high resistance to brute force attacks. The results of entropy and algorithm complexity analysis prove that the very large AES-256 key space and the exponential increase in bcrypt hashing time relative to the cost value make the probability of a successful attack close to zero within a reasonable operational time frame. In terms of performance, the system is still capable of providing a good response with an average authentication time of around 189 milliseconds per login and an authentication success rate of 85.7% in 35 tests.

This shows that the application of two layers of cryptography is still efficient for use in modern web applications. Furthermore, the relationship between security and performance (trade-off) can be controlled by adjusting the cost parameter in bcrypt and the size of the data encrypted using AES. Thus, developers can adjust the level of security as needed without significantly sacrificing system speed. The implementation of the system in Laravel has also proven to be practical and secure because it utilizes built-in libraries such as Hash and Crypt, which support salt generation and the automatic use of initialization vectors (IV). This allows cryptography integration without the need for additional external dependencies. However, this study has limitations because the testing was conducted in a local development environment with a limited number of samples. Therefore, the performance and security results obtained may differ when applied to large-scale systems or under real attack conditions (adversarial conditions). Overall, the combination of AES and bcrypt in Laravel provides a robust and efficient solution for enhancing user authentication security and can serve as a foundation for developing more secure adaptive login systems in the future.

5. References

Ahmadi, Mohammad, Mostafa Vali, Farez Moghaddam, Aida Hakemi, and Kasra Madadipouya. 2015. “A Reliable User Authentication and Data Protection Model in Cloud Computing Environments.”

Alanazi, Hamdan O, B B Zaidan, A A Zaidan, Hamid A Jalab, and M Shabbir. 2010. “New Comparative Study Between DES , 3DES and AES within Nine Factors” 2 (3): 152–57.

Algoritma, Implementasi, Bcrypt Pada, Hashing Generator, Berbasis Website, Alifan Widad Sutisna, Gunawan Budiarto, Muhammad Syukur, Xena Hadi Ramadhan, Antonius Yadi K, and Riza Pahlapi. 2023. “IMPLEMENTATION OF BCRYPT ALGORITHM ON WEBSITE-BASED HASHING GENERATOR USING LARAVEL FRAMEWORK” 7 (2): 199–212. https://doi.org/10.52362/jisicom.v7i2.1130.

Berkeley, U C, and Dawn Song. n.d. “Multi-Factor Credential Hashing for Asymmetric Brute-Force Attack Resistance.”

Bidhuri, Vicky, Niall Heffernan, and Niall Heffernan. n.d. “Enhancing Password Security Using a Hybrid Approach of SCrypt Hashing and AES Encryption MSc Internship Cyber Security National College of Ireland Supervisor :”

Cristy, Niolinda, and Fristi Riandari. 2021. “Implementasi Metode Advanced Encryption Standard (AES 128 Bit) UnCristy, N., & Riandari, F. (2021). Implementasi Metode Advanced Encryption Standard (AES 128 Bit) Untuk Mengamankan Data Keuangan. Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI), 4(2), .” *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)* 4 (2): 75–85. https://ejournal.sisfokomtek.org/index.php/jikom/article/view/181.

Elkabbany, Ghada F, Heba K Aslan, and Mohamed N Rasslan. 2014. “A D ESIGN OF A F AST P ARALLEL -P IPELINED I MPLEMENTATION OF AES : A DVANCED E NCRYPTION S TANDARD” 6 (6): 39–59. https://doi.org/10.5121/ijcsit.2014.6603.

Fedorchenko, Volodymyr, Olha Yeroshenko, Oleksandr Shmatko, Oleksii Kolomiitsev, and Murad Omarov. 2024. “Password Hashing Methods and Algorithms on the .Net Platform.” *Advanced Information Systems* 8 (4): 82–92. https://doi.org/10.20998/2522-9052.2024.4.11.

Force, Brute. 2021. “Analysis Performance BCRYPT Algorithm to Improve Password Security from Analysis

Performance BCRYPT Algorithm to Improve Password Security from Brute Force," no. March. https://doi.org/10.1088/1742-6596/1811/1/012129.

Francis, Nimmy. 2021. "Password Security Using BCrypt" 3 (1): 8–9. https://doi.org/10.5281/zenodo.5094166.

Gemawaty, Cut Asiana, and Yuce Yuliani. 2024. "Manajemen Identitas Dan Akses Dalam Keamanan Sistem Informasi." *Jurnal Manajemen Informatika Jayakarta* 4 (4): 396–403. http://journal.stmikjayakarta.ac.id/index.php/JMIJayakarta.

Liauren, Richie Mulyo, Baizul Zaman, and Syamsul Bahri. 2025. "Implementasi Algortima Aes Dan Bcrypt Untuk Pengamanan Data Pengguna Pada Website Jahitku." *KHARISMA Tech* 20 (1): 57–71. https://doi.org/10.55645/kharismatech.v20i1.535.

Makbull Rizki. 2022. "Perkembangan Sistem Pertahanan/Keamanan Siber Indonesia Dalam Menghadapi Tantangan Perkembangan Teknologi Dan Informasi." *Politeia: Jurnal Ilmu Politik* 14 (1): 54–62. https://doi.org/10.32734/politeia.v14i1.6351.

Melenia Bayu Aryanto, Muhlis Tahir, Silvia Irma Devita, Zuda Nuril Mustofa, Qurrotun Ainiyah, and Shelviatus Sundoro. 2023. "Implementasi Enkrip Dan Dekrip File Menggunakan Metode Advance Encryption Standard (AES-128)." *Jurnal Ilmiah Sistem Informasi Dan Ilmu Komputer*. https://doi.org/10.55606/juisik.v3i1.434.

Nugraha, Rhendy Diki, Muhamad David Ali, Nadya Khairunissa, Danang Nurcahyo, and Ahmad Turmudi Zy. 2025. "Strengthening Web-Based Login Security Using Vigenère Cipher and AES ENCRYPT () Method in MySQL" 7 (1): 90–99.

Nur, Khairunnisak, Didit Suhartono, Muhammad Thoriq, and Anisaa Qothrunnada. 2025. "Implementasi Pengamanan Data Menggunakan Teknik Bcrypt Hashing Password Dan Algoritma Advanced Encryption Standard (AES) Implementation of Data Security Using Bcrypt Hashing Password Technique and Advanced Encryption Standard (AES) Algorithm" 13 (1): 101–8. https://doi.org/10.26418/justin.v13i1.84997.

Putra, Wahyu, Muhammad Rizza Fahlevi, and Ahmad Tri Hidayat. 2023. "Implementasi Algoritma Advanced Encryption Standard Untuk Kemanan Dokumen." *Jurnal Ilmu Komputer, Teknologi Dan Informasi* 1 (2): 76–83. https://doi.org/10.62866/jurikti.v1i2.55.

Shi, Chaofang, Zhongwen Li, and Xiaoqi Li. 2025. "System Password Security: Attack and Defense Mechanisms." *Proceedings of Make Sure to Enter the Correct Conference Title from Your Rights Confirmation Email (Conference Acronym 'XX)* 1 (1).

Syam Aswandi, Andi, Muh. Nurtanzis Sutoyo, and Anjar Pradipta. 2025. "Analisis Performa Dan Keamanan Implementasi Kriptografi Aes Untuk Penyandian Dokumen Berbasis Web." *Jurnal Mnemonic* 8 (1): 24–32. https://doi.org/10.36040/mnemonic.v8i1.12053.

Vaishali, Ms, Kolhe Nilam, Kalyan Bamane, Abhijit Patankar, Sulbha Yadav, Archana Jadhav, and Shweta Mandal. 2024. "Design And Development Of Two Level Security For Data Using Integration Of Biometric Authentication And Bcrypt Algorithm" 27 (4).

Zulma, Gebrina Divva Meuthia, Henki Bayu Seta, and Trihastuti Yuniati. 2022. "Implementasi Algoritma Aes Dan Bcrypt Untuk Pengamanan File Dokumen." *Informatik : Jurnal Ilmu Komputer* 18 (2): 163. https://doi.org/10.52958/iftk.v18i2.4667.