84 DZULFIKAR KUDUS (translate EN) v02 (1) (1).docx

WORD COUNT 3472
TIME SUBMITTED 15-AUG-2026 02:06PM
PAPER ID 123687959

Journal of Computer Science and Technology
Volume X ; Number X ; Month Year ; Page 00-00
DOI : https://doi.org/10.59435/jocstec.xxxx.x
https://jurnal.padangtekno.com/index.php/jocstec
E-ISSN : 2985-4318
P-ISSN : 2985-5772

Implementation of the Advanced Encryption Standard-128 Algorithm and Huffman Compression to Protect Student Grade Data

Dzulfikar Ahmad Firdaus[1*], Fida Maisa Hana[2], Widya Cholid Wahyudin[3]
[1-3] Computer Science Study Program, Faculty of Science and Technology, Universitas Muhammadiyah Kudus
Ganesha I Street, Purwosari, Kudus, Central Java, Indonesia
[1*]42022110012@std.umku.ac.id, [2]fidamaisa@umkudus.ac.id, [3]widyacholidwahyudin@umkudus.ac.id

ABSTRACT

Student grade data is personal data that must be protected under Indonesian Law Number 27 of 2022 on Personal Data Protection, yet the rising number of data breaches in the education sector shows that such protection has not been optimally implemented. This study implements the 128-bit Advanced Encryption Standard (AES) algorithm combined with Huffman compression to protect the confidentiality and integrity of student grade files in Excel (.xlsx) format, and evaluates its effectiveness through functional, performance, and security testing. The system was developed as a Python desktop application with a CustomTkinter interface, in which the encryption process runs Huffman compression before AES-128, while integrity verification uses SHA-256 hashing. Testing was conducted on seven student grade files from MA Ma'ahid Kudus. The results show that all files were successfully encrypted and decrypted without failure or data alteration, with encryption times of 0.0142-0.0828 seconds and decryption times of 0.0046-0.1417 seconds. The SHA-256 hash values of the original and decrypted files were identical for all samples, yielding 100% recovery accuracy, while the file size increase from padding and header insertion ranged only from 1,092 to 1,101 bytes. The study concludes that the combination of AES-128 and Huffman compression is an effective and efficient solution for protecting student grade data in educational institutions.

Keyword : AES-128, Huffman Compression, SHA-256, Data Security, Student Grade Data

INTRODUCTION

Rapid developments in information technology have encouraged educational institutions to shift from conventional data management systems to digital-based systems. The implementation of efficient and structured systems not only improves user convenience but also plays an important role in ensuring regulatory compliance and overall operational efficiency [1]. Schools' need for digital data continues to increase along with the expanding use of information systems in academic activities, ranging from student data, attendance, grades, to financial information stored as digital assets of educational institutions. The proper, accurate, and secure use of information technology is an important factor in supporting the smooth running of the educational process [2].

On the other hand, digitalization has also created security vulnerabilities that are often exploited by *cybercrime* perpetrators to steal and misuse personal data [3]. Indonesia ranks eighth globally in the number of data breach cases, as seen in the theft of 15 million Bank Syariah Indonesia customer records, which exposed the fragility of the national data security infrastructure [4]. In the context of education, students' personal data stored in school information systems have become targets of increasing cybersecurity threats [5], while increasingly blurred privacy boundaries make student grade data easier to disseminate and potentially misuse [6]. Personal data refers to any information that can identify a person directly or indirectly, and the right to privacy is a fundamental right guaranteed under Article 28G paragraph (1) of the 1945 Constitution [7].

Various data breach cases in Indonesia underscore the urgency of personal data protection, such as the breach of 91 million Tokopedia user records by the ShinyHunters group in 2020 [8], the breach of approximately 13 million Bukalapak user records [9], and the breach of data belonging to 279 million Indonesian residents from BPJS Kesehatan in 2021 [10]. Educational institutions have also become potential targets for cybercriminals as data breach cases in the sector continue to increase [11]. In Indonesia, this urgency was addressed through Law Number 27 of 2022 on Personal Data Protection (PDP Law), which regulates data protection principles in line with human rights, the obligations of data controllers, and the principles of purpose clarity and data security, although its provisions remain general and do not yet provide optimal technical protection [12].

One technical method for maintaining the security of information transmission is cryptography, a technique that transforms *plaintext* into *ciphertext* through encryption and restores it through decryption. Advanced Encryption Standard (AES) is a symmetric block cipher algorithm with key lengths of 128, 192, or 256 bits and provides stronger security than the Data Encryption Standard (DES), which uses only a 56-bit key [13]. Comparative studies show that AES remains superior in terms of security compared with Blowfish and DES, although it requires slightly longer encryption time [14]. In the educational context, the application of AES-256 to the SMKN 4 Bandar Lampung database has proven effective in maintaining the integrity and security of student data [15], while the application of AES-128 to examination question files at SMK Plus PGRI 1 Cibinong successfully performed encryption and decryption without data alteration, with an IT expert feasibility rating of 88,88% [16].

Previous studies have generally assessed the performance of cryptographic algorithms separately or focused on general data types, and no study has specifically discussed the application of the AES algorithm to protect student grade data in spreadsheet format. Therefore, this study proposes the implementation of a combination of the AES-128 algorithm and Huffman compression to protect the confidentiality and integrity of student grade data in Excel (.xlsx) format, complemented by integrity verification using SHA-256 *hash*ing. The novelty of this study lies in integrating these three mechanisms while also conducting a comprehensive evaluation through three testing aspects, namely *functional testing*, *performance testing*, and *security testing*, thereby contributing to the standardization of a reliable, efficient academic data protection system suitable for widespread implementation in the information technology infrastructure of educational institutions.

RESEARCH METHOD

This study used a quantitative approach implemented through structured stages, including literature review, data collection and analysis, system and application architecture design, implementation of the AES-128 and Huffman algorithms, system testing, and evaluation of results. Each stage was designed sequentially and interrelated to ensure that the implementation results could be measured and scientifically accounted for.

Dataset and System Design

The research data were obtained directly in the form of Excel (.xlsx) files from MA Ma'ahid Kudus, comprising *pre-test* and *post-test* mathematics scores as well as daily test scores for the Nahwu subject from seven classes, namely Class X Religion 1, X Social Sciences 1, XI Natural Sciences 2, XI Religion 1, XI Religion 2, XI Natural Sciences 1, and XI Social Sciences 1. Each file contains student identities, class origin, and academic grades classified as sensitive data requiring protection through encryption mechanisms. The system was developed using the software specifications presented in Table 1.

Table 1. System Design Specifications

Component	Specification
Operating System	Windows 11
Programming Language	Python

Component	Specification
Encryption Library	pycryptodome (AES-128)
Compression Library	Huffman Algorithm
Integrity Library	Hash SHA-256
GUI Interface	CustomTkinter
Dataset	Excel (.xlsx), 7 student grade files

Python was selected because of its comprehensive cryptography library ecosystem, as demonstrated by the successful secure implementation of AES encryption and decryption using the pycryptodome library [17].

Advanced Encryption Standard-128 Algorithm

AES is a symmetric cryptographic algorithm established by the National Institute of Standards and Technology (NIST) in 2001 as a replacement for DES and is capable of providing reliable security while remaining easy to implement [18]. AES is considered the most effective algorithm for securing message transmission compared with other algorithms such as RSA and DES [19]. AES operates on 128-bit data blocks with a number of *rounds* that depends on the key length, as presented in Table 2.

Table 2. Number of *Rounds* in the AES Algorithm

AES	Key Length (bits)	Number of Rounds (Nr)
AES-128	128	10
AES-192	192	12
AES-256	256	14

The advantage of AES-128 lies in its fewer rounds compared with AES-192 and AES-256, making the computation lighter and faster, consistent with *benchmark* testing results on a 1 MB *file* showing AES-128 encryption times ranging from 0,01-0,05 seconds [20]. The developed system processes student grade data as *plaintext*, then runs AES-128 using a key derived from the user's *password* through an *initial round*, nine main rounds (SubBytes, ShiftRows, MixColumns, AddRoundKey), and one *final round* without MixColumns, as shown in Figure 1.

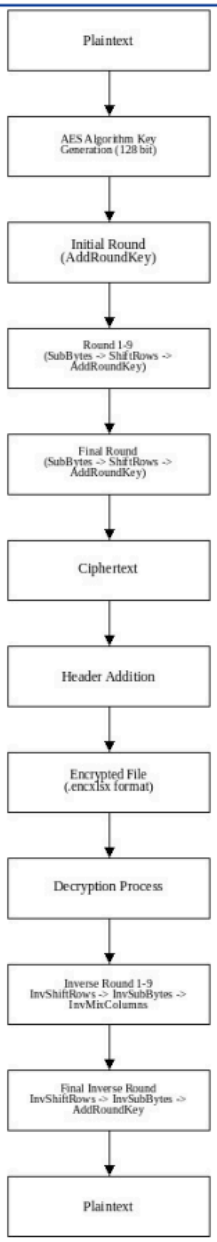

Figure 1. AES-128 Algorithm Encryption and Decryption Flow

Huffman Compression and SHA-256 Verification

The Huffman algorithm is integrated as a preprocessing stage before AES-128 encryption, operating by constructing a binary tree (*Huffman tree*) based on character occurrence frequencies, so that more frequently occurring characters receive shorter binary codes and the data size can be reduced without information loss (*lossless compression*). The combination of Huffman compression and AES-128 encryption has been proven effective in educational environments for securing school files [21]. After the encryption and decryption processes are completed, the system calculates the SHA-256 *hash* values of the original and decrypted files to verify data integrity. SHA-256 is a *one-way function* and is highly sensitive to even the slightest change in input, so any change to a bit of data will produce a completely different *hash* value [22]. If the two *hash* values are identical, it can be ensured that no alteration, loss, or corruption of data occurred during the cryptographic process.

System Architecture

The system architecture was designed with a *CustomTkinter*-based graphical user interface, encryption and decryption modules, and an automatic testing history recording feature, as shown in Figure 2. During encryption, the user selects an .xlsx file and enters a *password* as the key, after

which the system performs Huffman compression before AES-128 and generates an encrypted file in .encxlsx format. During decryption, the system runs AES-128 in reverse using the same key, followed by Huffman decompression, to reproduce the original .xlsx file. Each process is automatically recorded in the testing history, including the file name, size before and after processing, processing time, SHA-256 *hash* value, and integrity status.

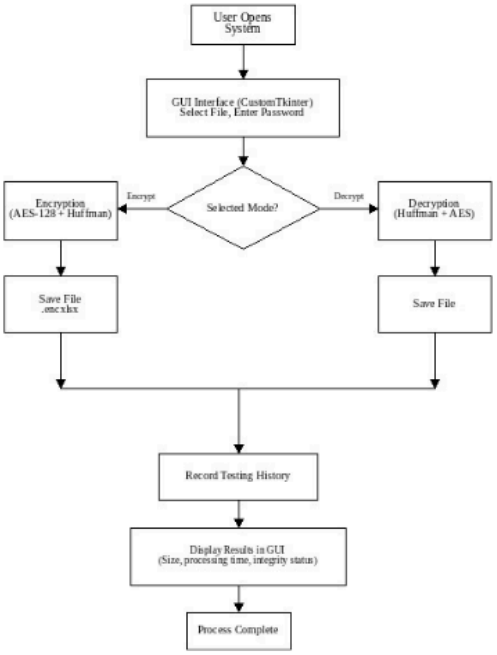

Figure 2. System Architecture

Testing Method

System evaluation was conducted through three quantitative testing aspects: *functional testing* to ensure that the encryption-decryption process operates according to specifications without data loss, *performance testing* to measure processing time efficiency, and *security testing* to verify data integrity through SHA-256 *hash*ing. Because AES-128 processes data in fixed 128-bit (16-byte) blocks, the file size after encryption is calculated using Equation (1), while the file size after decryption is calculated using Equation (2).

$$Senc = \lceil Splain / B \rceil \times B + H \quad (1)$$

$$Sdec = Senc - P - H \quad (2)$$

where Senc is the file size after encryption (bytes), Splain is the file size before encryption (bytes), B is the cipher block size, H is the system *header* size, and P is the *padding* size removed during decryption. Processing time efficiency is calculated using Equation (3), with Tprocess as the total processing time, Tend as the time when processing is completed, and Tstart as the time when processing begins.

$$Tprocess = Tend - Tstart \quad (3)$$

System accuracy is measured based on the correspondence between the size of the decrypted file and the original file using Equation (4).

$$Accuracy = (Matching\ Files / Tested\ Files) \times 100\% \quad (4)$$

RESULTS AND DISCUSSION

Results of *Functional Testing*

Functional testing was conducted on seven .xlsx student grade data files to ensure that the system could perform encryption through decryption without data alteration. The test results are presented in Table 3.

E-ISSN : 2985-4318
P-ISSN : 2985-5772

Table 3. System *Functional Testing* Results

Original File Name	Size Original (byte)	Size Encrypted (byte)	Enc. Time (s)	Size Decrypted (byte)	Dec. Time (s)
Nilai X Agama 1.xlsx	12.274	13.375	0,0384	12.274	0,0458
Nilai X IPS 1.xlsx	12.041	13.135	0,0142	12.041	0,0046
Nilai XI IPA 2.xlsx	19.852	20.944	0,0273	19.852	0,1417
Nilai Harian Nahwu XI Agama 1.xlsx	17.665	18.768	0,0828	17.665	0,0833
Nilai Harian Nahwu XI Agama 2.xlsx	16.469	17.568	0,0673	16.469	0,0822
Nilai Harian Nahwu XI IPA 1.xlsx	24.663	25.760	0,0600	24.663	0,1357
Nilai Harian Nahwu XI IPS 1.xlsx	20.843	21.936	0,0294	20.843	0,0984

Based on Table 3, all seven files were successfully encrypted into .encxlsx format and decrypted back to their original sizes without data loss. The increase in file size after encryption was caused by the addition of *padding* required by AES as a fixed 128-bit *block cipher*, as well as the insertion of the system metadata *header*, in line with *block cipher* theory, which states that data whose size is not a multiple of the block size requires additional *padding* [23]. The perfect match between the sizes of the decrypted files and the original files across all samples answers the first research question, namely that AES-128 combined with Huffman compression was successfully implemented functionally to protect student grade data in Excel format.

Results of *Performance Testing*

Based on Table 3, encryption times ranged from 0,0142 to 0,0828 seconds with an average of approximately 0,0456 seconds, while decryption times ranged from 0,0046 to 0,1417 seconds with an average of approximately 0,0803 seconds. All processes were successfully completed in less than 0,2 seconds. Variations in processing time were influenced by file size; larger files such as Nilai XI IPA 2.xlsx (19.852 bytes) required the longest decryption time, while smaller files such as Nilai X IPS 1.xlsx (12.041 bytes) were processed faster, consistent with the *block cipher* principle that the number of blocks is directly proportional to file size. Compared with the reference *benchmark* showing AES-128 encryption times ranging from 0,01-0,05 seconds for a 1 MB file [20], the results of this study, namely 0,0142-0,0828 seconds for files sized 12-25 KB, fall within a reasonable range, confirming that the AES-128 implementation with pycryptodome in Python operates according to its theoretical characteristics. This finding answers the second research question, namely that the system was proven effective in terms of *performance testing* with processing speeds adequate for real operational environments.

Results of *Security Testing*

Security testing was conducted by comparing the SHA-256 *hash* value of the original file before encryption with the *hash* value of the decrypted file. A summary of the verification results is presented in Table 4.

Table 4. Summary of SHA-256 *Hash* Verification

File Name	SHA-256 Hash (abbreviated)	Integrity Status
Nilai X Agama 1.xlsx	59d18b979cbac2300605c d8438e419b28ecab61b32 a69744056b6ee60ff1d066	Maintained
Nilai X IPS 1.xlsx	21cf032d094a36aedd519a 97d7bba8bfa8f914faad75 5b75ae9940772b89e145	Maintained
Nilai XI IPA 2.xlsx	09f256ab97ce68d34ea541 ad80c3954fcc814ed3fdf5 0645313e30c14899389d	Maintained
Nilai Harian Nahwu XI Agama 1.xlsx	b5b7ba8c507bd9019620f cb46e296e6fa208affdb 03120f89b32a92d6016f607	Maintained
Nilai Harian Nahwu XI Agama 2.xlsx	9c28f5536c6e4072941952 d4182fb389dc2c696a 50393b1a865e2035e9a00865	Maintained
Nilai Harian Nahwu XI IPA 1.xlsx	b77af9654194295043500 fe89e8f1e390748a081 ddb60ff1cb76cadd74edad7c	Maintained
Nilai Harian Nahwu XI IPS 1.xlsx	e60a32654955c24d24b 149316472e6a54fbaa6c817 cf6ef7b1c1db8988349a01	Maintained

The SHA-256 *hash* values of the original and decrypted files were proven identical for all seven tested samples, so the integrity status of all files was declared Maintained. This cryptographically proves that no alteration, loss, or corruption of data, however small, occurred throughout the entire encryption and decryption cycle, while also answering the third research question of this study. These results are consistent with findings demonstrating the consistency of the AES algorithm in maintaining data integrity in school database systems [15].

System Accuracy and General Discussion

Based on Equation (4), all seven decrypted files had exactly the same size as their original files, resulting in a system recovery accuracy of (7/7) × 100% = 100%. This accuracy is supported by the deterministic nature of AES-128 as a symmetric *block cipher* and the lossless characteristics of Huffman compression, ensuring that no information is lost during the compression and encryption cycle. The increase in file size due to *padding* and *header* was only between 1.092 and 1.101 bytes, indicating good storage efficiency despite Huffman compression being applied before encryption.

Overall, the developed system fulfills two of the three aspects of the CIA Triad that serve as information security standards, namely *confidentiality* through AES-128 encryption, which transforms data into unreadable *ciphertext* without the correct key, and *integrity* through SHA-256 *hash* verification, which ensures that the data remain unchanged during the process. Compared with studies applying AES-128 to secure examination question files [16] and a web-based combination of AES-256 and Huffman [21], this study demonstrates consistent results in terms of functional reliability while offering a new application context for protecting student grade data in spreadsheet format through a Python-based *desktop* interface.

CONCLUSION

Based on the research results, it can be concluded that the AES-128 algorithm combined with Huffman compression was successfully implemented as a solution for protecting student grade data

in Excel (.xlsx) format, with the system capable of encrypting and decrypting all seven student grade files intact using a 128-bit key and 10 rounds of cryptographic transformation. The effectiveness of the system was evaluated through three testing aspects: *functional testing* demonstrated the success of the entire process without failure or data alteration; *performance testing* showed high efficiency, with average encryption and decryption times of 0,0456 seconds and 0,0803 seconds, respectively, all below 0,2 seconds; and *security testing* through SHA-256 *hash* verification confirmed that data integrity was perfectly maintained across all samples. The integration of AES-128 and Huffman proved capable of maintaining the confidentiality and integrity of student grade data without data loss, achieving 100% recovery accuracy, making the system suitable for implementation as an academic data protection solution in educational institutions.

Recommendations

Future research is recommended to test the system using a larger *dataset* to obtain a more comprehensive picture of performance, compare the effectiveness of AES-128 with larger key variants such as AES-256 to identify the most optimal configuration in terms of security and efficiency, and involve direct user testing with teachers, administrative staff, and school data managers to measure the system's ease of use.

REFERENCES

[1] O. M. Haq, T. G. Pratama, and W. C. Wahyudin, "Implementasi Website Interaktif Untuk Sistem Manajemen Pembagian Gaji Perusahaan," vol. 2, pp. 22-33, 2025.

[2] M. Arridho, N. Sari, R. W. Ilham, and W. Amini, "Perkembangan Teknologi Dibidang Pendidikan," vol. 2, no. 5, pp. 468-475, 2022, doi: 10.36418/comserva.v2i5.345.

[3] T. G. Pratama, D. Rosita, Anwari, and Purbowati, "Peningkatan Kesadaran Keamanan Data Pribadi Dan Hukum Cyber," vol. 5, pp. 96-100, 2023.

[4] F. Fadliansyah, M. K. Mubarok, M. D. Aziz, and A. E. Augustia, "Evaluasi Pembelajaran dari Kasus Kebocoran Data di Indonesia sebagai Negara dengan Data Breach Terbesar Ke-8 Dunia," vol. 3, no. 2, pp. 272-277, 2025.

[5] S. N. Coalliani, A. E. K. Tejawati, S. H. Budiawati, Irmawati, L. Komariyah, and Y. Dwiyono, "Evaluasi Keamanan dan Manajemen Data pada Sistem Informasi Sekolah di Era Transformasi Digital," vol. 8, 2026.

[6] T. Lesmana, E. Elis, and S. Hamimah, "Urgensi Undang-Undang Perlindungan Data Pribadi Sebagai Pemenuhan Hak Atas Privasi Masyarakat Indonesia," vol. 3, no. 2, pp. 1-7, 2022.

[7] K. R. A. Suari and I. M. Sarjana, "Menjaga Privasi di Era Digital: Perlindungan Data Pribadi di Indonesia," vol. 1, pp. 132-146, 2023, doi: 10.38043/jah.v6i1.4484.

[8] D. N. Sari, A. Fauzi, S. N. Alya, I. Larasati, K. R. Sutrisna, S. I. Z. Zami, and N. A. Yunus, "Analisis Penanganan Insiden Kebocoran Data Tokopedia dan Dampaknya terhadap Kepercayaan Publik," vol. 2, no. 1, 2025, doi: 10.63217/orbit.v2i1.189.

[9] R. Milafebina, I. P. Lesmana, and M. R. Syailendra, "Perlindungan Data Pribadi terhadap Kebocoran Data Pelanggan E-commerce di Indonesia," vol. 4, no. 1, 2023.

[10] A. H. S. Nusantara, I. K. Umam, and M. Lubis, "Jaminan Informasi dan Keamanan yang Lebih Baik: Studi Kasus BPJS Kesehatan," vol. 18, 2024.

[11] Hidayatullah and Insanudin, "Peningkatan Kasus Pelanggaran Data di Sektor Pendidikan," 2016.

[12] N. O. Rahmadani, G. A. Fitri, and S. A. Wiraguna, "Perlindungan Data Pribadi sebagai Hak Asasi Manusia: Perspektif Hukum berdasarkan UU No. 27 Tahun 2022," vol. 5, no. 1, pp. 712-719, 2025.

[13] D. A. Meko, "Perbandingan Algoritma DES, AES, IDEA dan Blowfish dalam Enkripsi dan Dekripsi Data," Jurnal Teknologi Terpadu, vol. 4, no. 1, pp. 8-15, 2018.

[14] Z. S. Gandhara, "Evaluasi Kinerja Algoritma Kriptografi dalam Pengamanan Video: Studi Perbandingan AES, DES dan Blowfish," vol. 2, no. 2, 2025.

[15] D. S. Purwanti, M. Fadli, M. Surono, and E. R. Susanto, "Keamanan Database Aplikasi Manajemen Siswa," vol. 4, no. 2, pp. 111-119, 2025.

[16] B. Wicaksana and M. Setiawan, "Penerapan Algoritma Advanced Encryption Standard (AES) untuk Pengamanan Berkas Soal Ujian," vol. 10, no. 1, pp. 25-34, 2022.

[17] D. Limbachia and P. D. Tawde, "Encryption of Card Details Using AES with a 128-bit Key: A Secure Approach to Data Protection," pp. 1-9, 2024, doi: 10.55041/IJSREM29484.

[18] M. R. Andriyanto and P. Sukmasetya, "Penerapan Algoritma Advanced Encryption Standard (AES) Untuk Keamanan Data Transaksi Pada Sistem E-Marketplace," Journal of Computer System and Informatics (JoSYC), vol. 4, no. 1, pp. 179-187, 2022, doi: 10.47065/josyc.v4i1.2451.

[19] N. Noprianto and V. N. Wijayaningrum, "End To End Enkripsi Menggunakan Advanced Encryption Standard Pada Perangkat Internet of Things," Jurnal Sistem Informasi Dan Bisnis Cerdas, vol. 14, no. 2, pp. 98-107, 2021, doi: 10.33005/sibc.v14i2.2734.

[20] M. Marwaha, R. Bedi, A. Singh, and T. Singh, "Comparative Analysis of Cryptographic Algorithms," pp. 1-3, 2013.

[21] N. A. Kafa, D. Virgian, and S. Yudha, "Implementasi Kriptografi Berbasis Web dengan Algoritma Advanced Encryption Standard (AES) 256 dan Kompresi Huffman untuk Pengamanan File di SMK Satria," vol. 12, pp. 50-55, 2024.

[22] N. A. Tarigan, A. C. N. Anggreni, A. Balqis, I. Nurfadilah, E. S. Bakti, and F. Mahyudin, "Pengembangan Aplikasi Kriptografi RSA dan SHA-256 Berbasis Web Menggunakan Flask," vol. 1, no. 2, pp. 40-44, 2025.

[23] W. Stallings, Cryptography and Network Security, 6th ed. Pearson, 2016.

84 DZULFIKAR KUDUS (translate EN) v02 (1) (1).docx

ORIGINALITY REPORT

4%
SIMILARITY INDEX

PRIMARY SOURCES

1	Samsul Arifin. "Intelligent Adaptive Hybrid Encryption Based on Unimodular Hill Cipher, RSA-OAEP, and Chaotic Logistic Maps for Universal Digital File Security", Springer Science and Business Media LLC, 2026 Crossref Posted Content	29 words — 1%
2	journal.privietlab.org Internet	25 words — 1%
3	Xuyang Jiao, Youzhi Shen, Jian Li, Xiuli Yu, Ying Zhang. " Research on the Application of an Improved AES-based Encryption Algorithm on a Multi-Protocol Conversion Gateway with Domestic Chip ", 2024 IEEE International Conference on Real-time Computing and Robotics (RCAR), 2024 Crossref	22 words — 1%
4	journal.padangtekno.com Internet	19 words — 1%
5	Tanti Kirana Utami, Kayla Andini Putri, Salsa Octaviani Suryanto, Fina Asriani. "Personal Data Breach Cases In Indonesia : Perspective Of Personal Data Protection Law", Journal Customary Law, 2025 Crossref	18 words — 1%
6	www.ijert.org Internet	17 words — 1%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	< 1%
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	< 9 WORDS