PAPER NAME

Comparative Analysis Between Advanced Encryption Standard.docx

AUTHOR

Rizqi, Nurdin, Taufiq Rizqi, Nurdin, Taufiq

WORD COUNT

6119 Words

CHARACTER COUNT

33305 Characters

PAGE COUNT

14 Pages

FILE SIZE

708.9KB

SUBMISSION DATE

Jun 19, 2024 1:21 PM GMT+7

REPORT DATE

Jun 19, 2024 1:22 PM GMT+7

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 4% Internet database
- 4% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Cited material
- Small Matches (Less then 10 words)

Jurnal CoreIT, Vol.X, No.X, Month Year
ISSN 2460-738X (Print)
ISSN 2599-3321 (Online)

Comparative Analysis Between Advanced Encryption Standard and Fully Homomorphic Encryption Algorithm to Secure Data in Financial Technology Applications

Rizqi Mulki[1], Nurdin[2*], Taufiq[3]
[1]Student Master of Information Technology Study Program, Faculty of Engineering, Universitas Malikussaleh, Aceh, Indonesia.
[2,3]Master of Information Technology Study Program, Faculty of Engineering, Universitas Malikussaleh, Aceh, Indonesia.
rizqi.227110201009@mhs.unimal.ac.id, nurdin@unimal.ac.id (*Corresponding Author), taufiq@unimal.ac.id

Abstract. This research discusses the comparison between two encryption algorithms, namely Advanced Encryption Standard (AES) and Fully Homomorphic Encryption (FHE), in the context of data security in Financial Technology (Fintech) applications. The main aim of this research is to analyze the speed and efficiency of the two algorithms to provide information and motivation to Fintech Application business actors to determine the right algorithm for securing data. The research results show that AES is faster and more efficient in terms of encryption and decryption compared to FHE. For encryption, the AES algorithm is 1,100 times faster than the FHE algorithm. For decryption, the AES algorithm is 581 times faster than the FHE algorithm. For arithmetic processing, AES is 132 times faster than FHE. CPU consumption for AES encryption is 35.93% lower CPU usage than FHE. In AES decryption 10.31% lower than FHE for CPU usage. In the arithmetic process AES is 9.33% lower in usage than FHE. For memory usage in the FHE encryption process, it has an advantage, namely 2.3 times lower than AES for memory usage. During decryption, AES memory usage is superior with memory consumption 54 times lower than FHE. For the arithmetic process, AES uses 4.3 times lower memory than FHE. Overall AES provides speed and low resource consumption, this makes AES very suitable for use in Fintech applications that require speed and efficiency. Even though FHE has advantages in memory usage during encryption alone, this is not enough because it takes a long time to carry out the encryption process. This research suggests that further research will attempt to make the FHE algorithm more efficient and faster in processing data, this is considering the potential of FHE which is able to process encrypted data.

Keywords: Advanced Encryption Standard (AES), Fully Homomorphic Encryption (FHE), data security, Fintech, encryption, decryption

Received …….. / **Revised** …… / **Accepted**……..

INTRODUCTION

The use of smartphones is currently a necessity for society, because the current applications in circulation are to help meet people's daily needs. From transportation, communication, finance and education issues are available on their smartphone application, so this is an important concern that needs to be managed well, because it is related to the needs of many people. Especially in matters related to finance, where nowadays it is so easy for people to use financial technology to fulfill their daily life needs. Examples of financial transactions that people often carry out are money transfer transactions, credit top ups and other online payments. By seeing the high demand of society in using online financial transactions, technology is currently emerging with the term Fintech (Financial Technology). Many entrepreneurs are competing to reach the market in this financial sector. This is proven by the existence of the Indonesian Fintech Association, whose members will be 366 companies in 2022. With details of members, 102 online loan Fintechs, 84 digital financial innovation Fintechs (IKD), 39 payment system Fintechs, 13 technology partner Fintechs, 6 financial institution Fintechs, 5 capital market Fintechs, 4 digital asset Fintechs and 113 other Fintech companies [1].

Based on data from OWASP (Open Web Application Security Project), which is a reference for Web Application security in the world, Cryptographic Failure which causes data leaks is ranked number 2 in the world with 233,788 cases in 2021 [2]. OWASP suggests several preventive measures for this case, including ensuring that data is encrypted and ensuring that a strong encryption algorithm is used as a form of data protection or what we know as Data Security. Encryption is one part of Data Security [3]. Encryption is a technique of converting data into an encrypted format that can only be read with the correct encryption key.

DOI: 10.24014/coreit.vxxx.xxxx

Encryption provides organizations with the ability to protect sensitive data and confidential data. In general, there are 2 types of encryption based on the key used, namely symmetric encryption (same key for encryption and decryption), the other is asymmetric encryption (public key for encryption and private key for decryption) [4]. In terms of data processing resulting from encryption, there are two types, namely the arithmetic process directly on the cipher (encryption results) and the cipher arithmetic process via decryption of the data first. FHE (Fully Homomorphic Encryption) is an example of an encryption algorithm that can directly carry out arithmetic processes on the cipher, while AES (Advanced Encryption Standard) is an example of an encryption algorithm that requires a decryption process first to perform arithmetic. These two algorithms support a key length of 256 bits, which means the possibility of cracking is 3.31×10^{56} years [5].

Research conducted by [6]is an effort to maintain data confidentiality in the form of securing short messages or what we know as SMS using 3DES (Triple Data Encryption Standard) encryption which produces an Android application for reading messages and sending protected messages. So even cellular operators cannot read the contents of messages sent using this application. This is very useful for people in maintaining the confidentiality of their information when communicating, and also research conducted by [7] regarding efforts to prevent crime through online shops, because of the large number of online crimes that occur. Other further research related to data security techniques using AES is that in this research applied AES cryptography to secure documents with the results obtained that AES cryptography is reliable against data theft [8]. AES is an abbreviation for (Advanced Encryption Standard). Which is cryptography issued by the American Government to secure government data. AES is usually implemented in hardware and software to encrypt sensitive data [9]. The flexible key length can be 128-bit, 192-bit and 256 bit [10]. In research conducted by [5] the focus was on 256-bit keys, because research results show that personal data security is very good and safe.

Other research related to data security with AES cryptography concerns the implementation of AES-256 to secure personal data. From this research it was found that AES cryptography data security can be maintained because the algorithm carries out security and encoding in layers [5]. Previous research that has been carried out in Fintech security efforts is research [11] using Partially Homomorphic Encryption, where the encryption process produces encryption results that allow arithmetic to be carried out, but the drawback of Partially Homomorphic Encryption is that it only allows one type of arithmetic. This research is a proposal to secure bank data which is very similar to Fintech transactions. Another research is [12] using RSA encryption which is used to secure data in Internet Banking applications. Another research is [13] using AES to secure Internet Banking data. Apart from being related to securing related journal data, there is the issue of website performance testing carried out by [14] regarding performance testing using the PIECES method.

Another research is research conducted by Mustafa Noori Rashid, Leith Hamid Abed and Waleed Kareem Awad entitled financial information security using hybrid encryption technique on multi-cloud architecture in 2022. In this research, a proposed system is developed based on three different keys. They divide data into non-sensitive, sensitive, and very sensitive data. The proposed system uses different keys for encryption and decryption purposes. A distributed cloud-based secure data storage (DDSPE) approach using elliptic curve cryptography (ECC) is proposed to provide big data-based secure data protection across multiple clouds. With DDSPE technology, ECC schemes have been used for encryption and decryption purposes. Cloud is used for simulation. The test results show that the proposed DDSPE system is safe and efficient in terms of data storage and retrieval. To analyze performance, researchers compared the DDSPE method with advanced encryption standards (AES), blowfish, Rivest Shamir Adleman (RSA), efficient distributed storage based on security (SA-EDS), and secure distributed storage based on attributes (ASDSS). In terms of information storage and recovery, our methodology is very effective as it requires less time compared to other strategies. What differentiates the author's research is adding a comparison with Fully Homomorphic Encryption (FHE) because this algorithm has the priority of processing encrypted data which will provide efficiency in application development work [15].

One of the Fintech Applications that has not yet secured data is the Nurapay Application owned by PT. Deacas International Trade, cryptography has not been implemented in the database. This means that if a criminal manages to enter the server, it will be very easy to read the data contained in the database. The Nurapay application is used as research material in this study. The importance of this research is to find the right algorithm to secure data, without having to sacrifice access speed to the Nurapay Application in

particular and Fintech Applications in general. And what is no less important than this research is how to implement the AES Algorithm and FHE Algorithm efficiently without having to change the overall application structure.

Therefore, in an effort to reduce the impact of Cryptographic Failure, research was carried out in the form of comparing 2 security algorithms to achieve efficiency in securing financial data in the form of a final project with the title "Comparative Analysis between AES Encryption (Advanced Encryption Standard) and FHE Encryption (Fully Homomorphic Encryption) in securing data on Fintech Applications".

METHODS

A. Research Stages

The research method used in this research is quantitative. The research steps carried out are as shown in the following:

1. Data collection consists of two types of data:
    a. Primary Data, in this case the author made direct observations of the data structure of the nurapay application along with the source code of the nurapay application to determine data requirements and the frequency of data requirements in the application and has obtained permission from PT. Deacas International Trade. The first step is to carry out the data definition process by taking the data structure that has been provided by Nurapay to be used as a sample data standard later. The second step is to generate sample data using the website https://generatedata.com/ by following the data definition structure carried out in the first step.
    b. The secondary data referred to by the author is supporting data in the form of literature studies in the form of fundamental theories and other relevant information. The literature study that the author took includes journals, books and other literature related to data security and application performance. Other media that the author uses is internet media whose discussion is related to this research.
2. Data Analysis and Processing which is a process carried out after testing and system development is carrying out data analysis which will later determine the results of the research. Because this section plays a very important part in concluding the results of this research. The steps taken in testing are to test the performance of data encryption and decryption using AES and FHE before implementing it into the Nurapay Application to analyze the performance per column in the database. The data tested starts from customer data, then product data and then finally the data transaction.
3. Research System Design, namely at this stage the author will design an application system that can complete the goal of securing the Nurapay Application. The first thing to do is design the system framework. The process of designing the system framework in question is by analyzing the source code of the Nurapay Application and then redesigning the Nurapay Application by applying the concept of data security.
4. Implementation is the stage for implementing test results into the Nurapay Application and translating the analysis results into Nurapay Application integration, so that data security can be achieved while still considering application performance.
5. Testing is the final stage, namely carrying out testing using the Blackbox Testing method. The author tested the application performance by comparing the application performance without the decryption encryption process and the application performance after the decryption encryption process was implemented. This test was carried out using Apache Benchmark software.

B. Data Collection

The list of data used as samples in this research is Customer Data. Customer Data is data used in the Nurapay Application to store customer data, which will later be used as the main reference in carrying out transactions. For example, for each transaction data, customer identity data is required to find out the transaction history.

users
id int(11) card varchar(16) name varchar(128) email varchar(128) phone varchar(15) username varchar(32) balance int(13) level enum('admin','user') status ('active','lock','suspend') pin varchar(128) kk varchar(16) ktp varchar(16)

Fiqure 1. Database Structure of Users

C. AES Algorithm

According to [16] AES (Advanced Encryption Standard) is an encryption standard used to maintain data security. This is a symmetric cryptographic algorithm, meaning that one key is used to perform both encryption and decryption of data. The main goal of AES is to provide a high level of security and efficient use of computer resources.

In the AES algorithm encryption process, there are 5 stages that must be passed [17], that is:

1. AddRoundKey: This is an initial round which is an initial initialization process by carrying out an XOR process for each byte in the state matrix (plaintext) with each byte of the encryption key (ciphertext).
2. SubBytes: The next process is to convert bytes from the addroundkey state matrix results using a replacement array or what is called an S-Box. The shape of the S-Box can be seen in Figure 2.

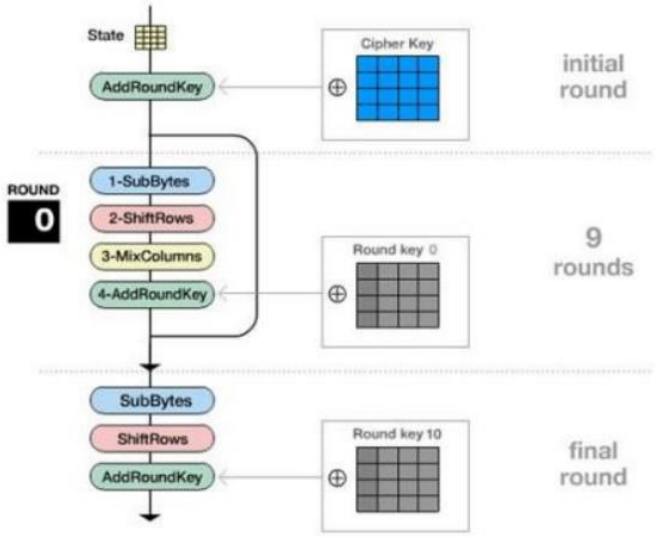
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Fiqure 2. S-Box [17]

3. Shiftrows: This step is to shift the bit shift, where the leftmost bit is shifted to the rightmost bit. The amount to perform a shift for each row is different. The first line is not shifted, the second line is shifted to the left by one byte, the second line is shifted to the left by one byte, the third line is shifted to the left by two bytes, and so on.
4. MixColumns: is the process of multiplying the matrix in each column. Each column in the matrix is multiplied by the matrix of the next column.
5. AddRoundKey: is an XOR operation process for each byte output from MixColumn with RoundKey.
6. Next, repetition is carried out depending on the length of the key used, as follows:
   - 10 rounds for 128-bit keys.
   - 12 rounds for 192-bit keys.

- 14 rounds for 256-bit keys.

The AES algorithm process mentioned above can be depicted in Figure 3 below:



Fiqure 3. AES Algorithm Schema [17]

D. FHE Algorithm

FHE stands for Fully Homomorphic Encryption. Fully Homomorphic Encryption is a type of cryptographic encryption that allows processing encrypted data efficiently, without the need to decrypt it first. In other words, FHE makes it possible to perform mathematical operations, such as addition, subtraction, multiplication, and others, directly on data that is still encrypted.

The FHE scheme that the author chose is BGV (Brakerski-Gentry-Vaikuntanathan) because it supports exact letter calculations. And the BGV algorithm displays better results compared to other algorithms such as CKKS. As a comparison, based on the results of research [18] it shows that for 8 bit BGV encryption it only takes 7.09 s while CKKS shows results of 89.61. The following is the formula for the BGV scheme:

1. Parameter Preparation

   Before carrying out the encryption and decryption process, a number of parameters must first be prepared, that is :

   - q = ciphertext modulus
   - n = polynomial modules degree
   - t = plaintext modulus
   - λ = security parameter (128,192,256)

   The q value is influenced by the security parameter, provided that if it is 128-bit, the q value is between $2^{256} – 2^{230}$. If it is 192-bit then the q value is between $2^{384} – 2^{448}$. If it is 256-bit then the q value is between $2^{512} – 2^{640}$. For the t value must be greater than the value of q. Meanwhile, the n value is also influenced by the security parameter, provided that if it is 128 then the recommended n value is 1024 – 2048, if 192 then the recommended n value is 4096, and if 256 then the recommended n value is 8192 – 16384.

2. Key Generation

   - Secret Key (SK): secret key with the symbol s(x)
   - Public Key (PK):
     - Choose a random polynomial: a(x)
     - Calculate : b(x)=−(a(x)·s(x)+e(x))mod q
     - Public key: $PK=(a(x),b(x))$

3. Encryption

   To encrypt messages denoted by the letter m, the following encryption function is used:

   - Plaintext: Converted to small polynomial $m(x)$
   - Noise polynomial: Random polynomial with small coefficients: $e0(x),e1(x)$
   - Random polynomial: $r(x)$
   - Ciphertext: $c(x)=(c0(x),c1(x))$

$c0(x)=b(x)\cdot r(x)+e0(x)+t.\ m(x) \bmod q$
$c1(x)=a(x)\cdot r(x)+e1(x) \bmod q$

4. Decryption

The decryption function in the BGV scheme is as follows:

$$m'=[c0+c1\cdot s] \bmod q$$

- s is the private key used for decryption
- Subtraction operations in polynomial space are performed on each polynomial component of c1 and c2

RESULT AND DISCUSSION

A. Data Analysis

In general, the data analysis process flow is carried out as shown in Figure 4 below:

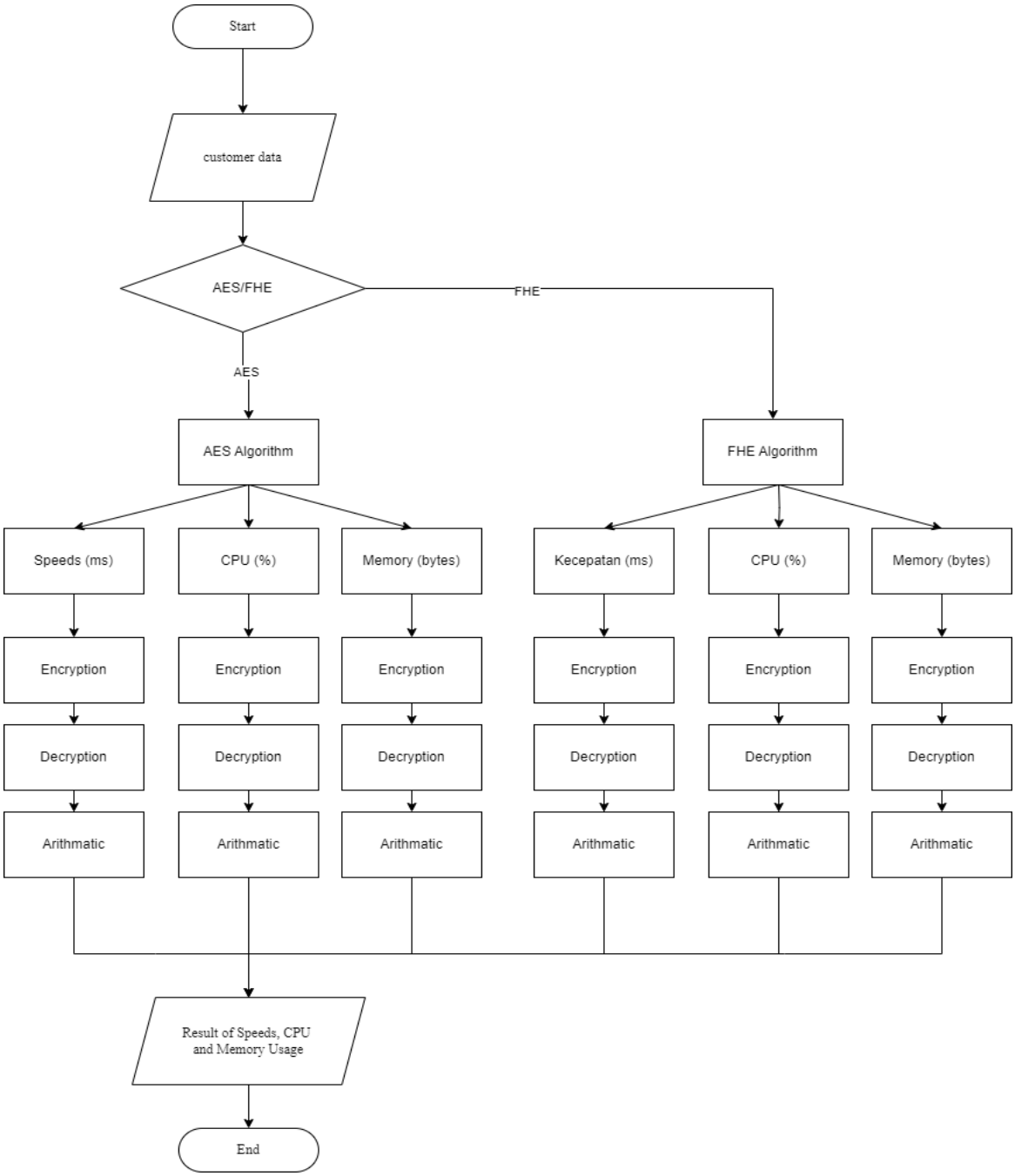


Figure 4. Analysis Schema

Figure 4 is a flowchart that illustrates how customer data, product data, and transaction data are processed through the AES and FHE Algorithms to measure application performance in terms of speed, CPU usage, and memory usage. Following is a detailed explanation of the image:

1. Start: The flowchart starts from the "Start" point.
2. Data Input: Three types of data (customer data, product data, and transaction data) will be retrieved and processed.
3. Encryption Algorithm Selection (AES/FHE): The data obtained will be processed through one of two encryption algorithms: AES (Advanced Encryption Standard) or FHE (Fully Homomorphic Encryption). There is a fork here to choose whether to use AES or FHE . If the AES algorithm data will be processed through the following steps:
   - Speed (ms): Measures the speed of encryption, decryption and arithmetic operations. To analyze speed, you need to measure the time difference between the start of the program and the end of the encryption process.
   - CPU Usage (%): Measures CPU usage for encryption, decryption and arithmetic operations by utilizing the process utilities provided by the Nodejs programming language.
   - Memory (bytes): measures memory usage during encryption, decryption and arithmetic operations. By using a Nodejs utility that specifically measures the amount of memory usage for applications.

   Next, FHE is selected, so several processes will also be measured:
   - Speed (ms): Measures the speed of encryption, decryption and arithmetic operations. To analyze speed, you need to measure the time difference between the start of the program and the end of the encryption process.
   - CPU Usage (%): Measures CPU usage for encryption, decryption and arithmetic operations by utilizing the process utilities provided by the Nodejs programming language.
   - Memory (bytes): measures memory usage during encryption, decryption and arithmetic operations. By using a Nodejs utility that specifically measures the amount of memory usage for applications.
4. Output Measurement Results After all the steps above have been carried out, the speed measurement results, CPU usage and memory usage of the two algorithms (AES and FHE) will be stored in the database.
5. Finish (End).

To save the results of the analysis carried out based on Figure 3.5, they are stored in a table with the name map_fields which is shown in Figure 3.6 below:

map_field
id (int 11)
table_name (varchar 100)
field_name (varchar 100)
speed_aes_encryption (bigint 20)
speed_aes_decryption (bigint 20)
speed_aes_arithmatic (bigint 20)
cpu_aes_encryption (decimal 6,2)
cpu_aes_decryption (decimal 6,2)
cpu_aes_arithmatic (decimal 6,2)
memory_aes_encryption (bigint 20)
memory_aes_decryption (bigint 20)
memory_aes_arithmatic (bigint 20)
speed_fhe_encryption (bigint 20)
speed_fhe_decryption (bigint 20)
speed_fhe_arithmatic (bigint 20)
cpu_fhe_encryption (decimal 6,2)
cpu_fhe_decryption (decimal 6,2)
cpu_fhe_arithmatic (decimal 6,2)
memory_fhe_encryption (bigint 20)
memory_fhe_decryption (bigint 20)
memory_fhe_arithmatic (bigint 20)

Fiqure 5. Database Structure to Store Analysis Result

Hardware analysis is needed to find out in depth what devices are needed to support this research so that it can produce the Nurapay application which already has the concept of data security in it. This research was conducted using the following specifications:
1. Asus Vivobook 15 laptop
2. Processor Intel ® Core ™ i5-7th Gen CPU @ 2.50 GHz
3. 16GB RAM
4. 512GB SSD

Software Requirements Analysis to find out the software needed to support this research, namely:
1. Operating System: Microsoft Windows 10
2. Database: MariaDb
3. Webserver: Xampp (PHP 7.4)
4. Benchmark Tools: Apache Benchmark
5. Text Editor: Sublime Text 3 / Visual Studio Code
6. Programming Languages: PHP, NodeJs, Js, CSS, and HTML

The author describes the technique for measuring performance based on the measurement method that follows [19], that is :

1. Calculating speed: the author uses the concept of the time difference between the start of the program and the end of the program
2. Measuring CPU usage: the author uses the CPU gauge library which can calculate CPU usage for running lines of code
3. Measuring memory usage: the author uses the process library to capture memory usage for running lines of code.

Result for Customer Data analysis describe in table below:

Table 1. Analysis Result of Customer Data

Nama Kolom	Algoritma	Kecepatan (Milisecond)			CPU (Percent %)			Memori (Bytes)		
		Enkripsi	Dekripsi	Aritmatika	Enkripsi	Dekripsi	Aritmatika	Enkripsi	Dekripsi	Aritmatika
id	AES	27	21	42	31.40	73.49	87.34	69444	68951	51911
	FHE	9533	6204	5587	99.53	77.26	82.94	20536	4881471	302143
card	AES	18	22	-	75.88	77.10	-	60852	87605	-
	FHE	8884	6065	-	104.65	78.21	-	38470	4889847	-
name	AES	16	15	-	51.97	77.44	-	71445	69972	-
	FHE	10711	6022	-	105.64	73.43	-	30016	4890087	-
email	AES	12	13	-	76.98	73.67	-	139351	180461	-
	FHE	14081	6953	-	107.62	98.84	-	72042	4881903	-
phone	AES	9	18	-	77.02	72.62	-	87992	72746	-
	FHE	13814	6038	-	103.97	85.69	-	46448	4889751	-
username	AES	9	14	-	76.47	77.08	-	83572	71795	-
	FHE	11155	6182	-	105.46	93.89	-	42465	4889895	-
komisi	AES	9	10	29	67.97	73.45	80.23	67895	68124	67884
	FHE	12914	14686	4107	105.33	94.85	96.55	18550	4889935	302415
balance	AES	12	11	23	97.30	72.31	70.91	73766	70199	68286
	FHE	10339	12371	3898	107.92	89.70	96.41	32176	4889847	302327
point	AES	8	9	19	49.15	51.46	91.71	67732	84521	67665
	FHE	13480	14419	4223	104.19	92.40	92.97	18548	4881439	302111
level	AES	9	10	-	66.84	72.86	-	69311	85761	-
	FHE	9295	12054	-	103.27	96.92	-	28480	4881815	-
status	AES	14	9	-	74.83	48.18	-	71847	86474	-
	FHE	11791	5792	-	104.08	88.02	-	30487	4882007	-
pin	AES	8	8	-	75.76	73.72	-	66849	68903	-
	FHE	13452	8517	-	104.98	71.83	-	17964	4890087	-
kk	AES	10	10	-	61.66	95.88	-	124703	125993	-
	FHE	15367	10667	-	84.07	83.02	-	50257	4881959	-
nik	AES	12	13	-	98.22	96.64	-	125189	178538	-
	FHE	9589	8740	-	78.51	79.00	-	50259	4889551	-
RATA - RATA	AES	12,36	13,07	8,07	70,1	73,99	23,59	70320,37	83876,84	64031,75
	FHE	11743,21	8907,86	1272,5	101,37	85,93	26,35	25339,47	4889558,58	302327,00

Table 1 is the result of a comparison between the AES (Advanced Encryption Standard) and FHE (Fully Homomorphic Encryption) algorithms based on several performance parameters for customer data. From the results of Table 4.1 above, AES data processing is generally faster in encryption and decryption compared to FHE. FHE uses more CPU and memory resources, especially on encryption and decryption operations. Arithmetic operations on encrypted data (relevant for FHE) require significant time and resources, indicating that FHE is more computationally intensive than AES. We see that the CPU usage for the FHE algorithm exceeds 100%, this is even though Nodejs only uses 1 core, in Nodejs there is an asynchronous feature that utilizes the Libuv library which handles asynchronous I/O operations. Libuv has a thread pool that is used to handle blocking operations such as file system I/O and DNS lookups outside of the main Node.js thread, allowing some operations to run in parallel in background threads.

To further clarify the meaning of Table 1, the author describes it in the form of an image based on average data obtained by adding and then dividing by the number of columns. The following is a graphic image for the speed in processing customer data:

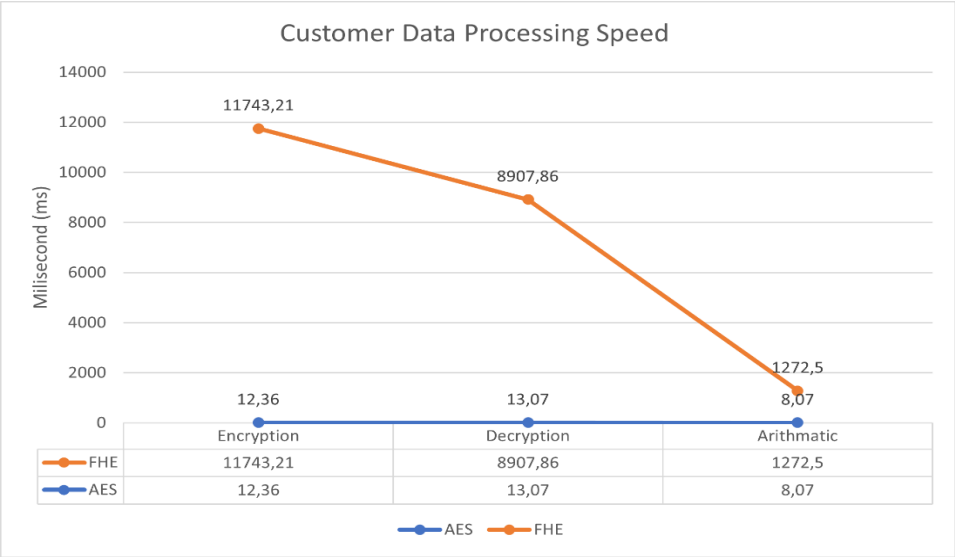


Figure 6. Customer Data Processing Speed

Figure 6 compares the processing speed between two encryption algorithms: AES (Advanced Encryption Standard) and FHE (Fully Homomorphic Encryption) in three categories: Encryption, Decryption, and Arithmetic. The figure shows that AES is very fast and efficient for all processes compared to FHE. The AES encryption process is 950 times faster than FHE. For decryption, AES is 681 times faster than FHE, however FHE is relatively more efficient in carrying out arithmetic operations compared to the Encryption and Decryption process, however it is still slower than AES, 159 times faster than FHE. This shows that AES is better suited for applications that require fast encryption and decryption, while FHE, although slower, offers the unique ability to perform arithmetic operations on encrypted data. Furthermore, for CPU usage the author also made a special image by taking the average value from Table 1, the Figure 7 below are the image results:

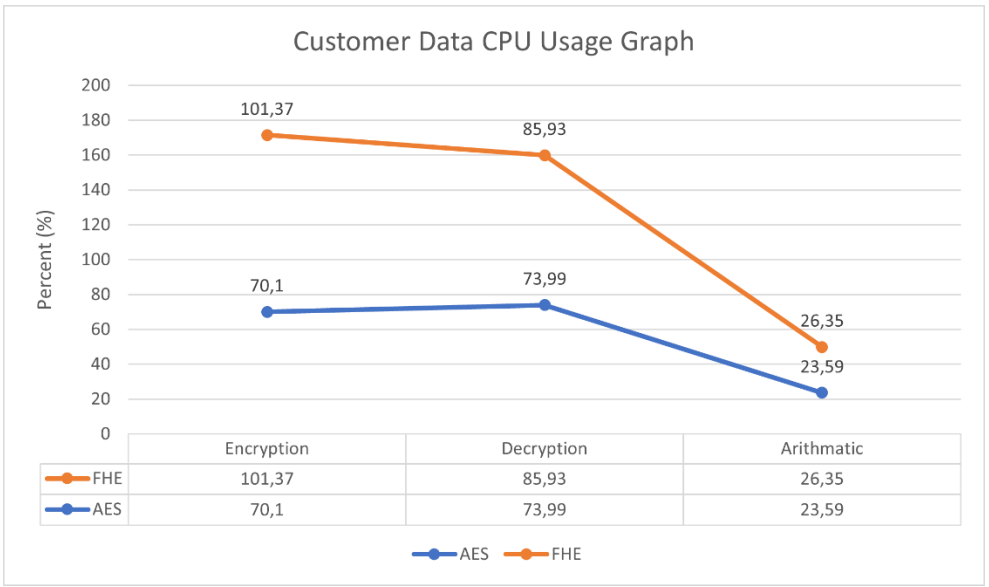


Figure 7. Customer Data CPU Usage Graph

Figure 7 compares CPU usage between two encryption algorithms AES (Advanced Encryption Standard) and FHE (Fully Homomorphic Encryption) in three categories, namely Encryption, Decryption, and Arithmetic. AES uses less CPU compared to FHE for all types of processes (Encryption, Decryption and Arithmetic). For AES encryption, CPU usage is 31% lower than FHE. For AES decryption, CPU usage is 11% lower than FHE. However, the difference starts to be small when processing arithmetic, namely AES is lower for CPU users by 2.76%. This indicates that AES is more efficient in CPU usage for encryption and decryption operations compared to FHE. Although FHE offers unique capabilities for arithmetic operations on encrypted data, it requires more CPU resources, which can be an important factor in algorithm selection depending on specific needs and resource constraints. Next Figure 8 for memory usage:

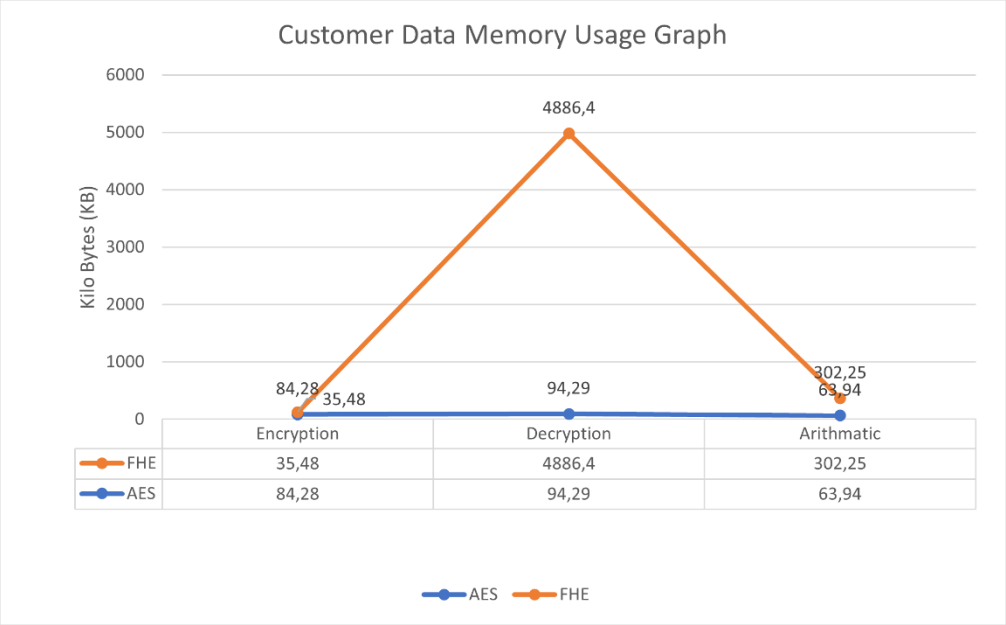
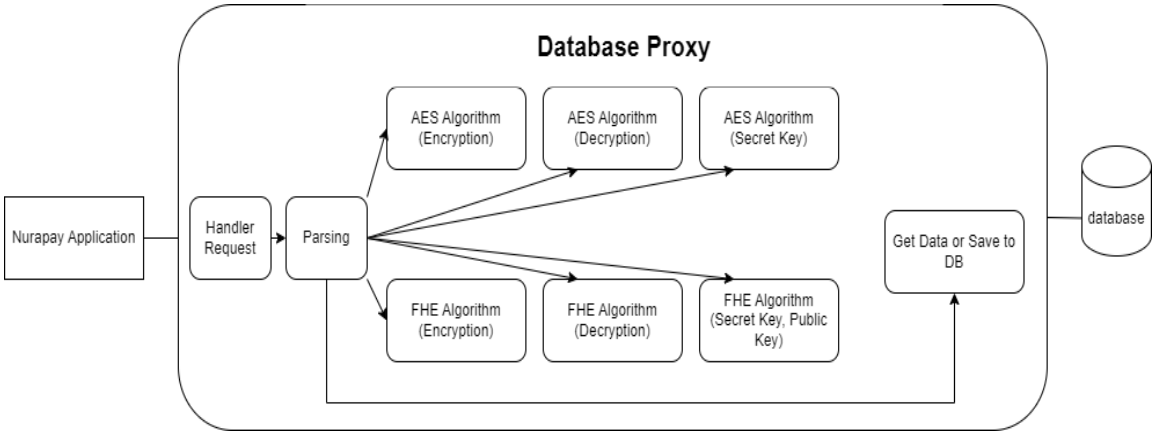


Figure 8. Customer Data Memory Usage Graph

Figure 8 compares memory usage between two encryption algorithms: AES (Advanced Encryption Standard) and FHE (Fully Homomorphic Encryption) in three categories: Encryption, Decryption, and Arithmetic. For the FHE encryption process, it is superior in memory usage, namely 2 times less than the memory usage for AES. For the decryption process, AES is far superior to FHE, namely lower memory usage by 51 times. For the arithmetic process, although the difference is not as big as in decryption, AES is still superior with 4 times lower memory usage compared to FHE. The Decryption process for FHE shows very high memory usage, which is caused by the large ciphertext file in the FHE scheme. For example, for encryption of the number 36 AES stores 44 Bytes of characters, while FHE stores 241236 Bytes, this causes a large amount of memory usage during decryption. Overall, memory usage by FHE varies greatly depending on the process type, with a particularly high peak at Decrypt. This indicates that although FHE can reduce memory usage for Encryption, it requires significantly more memory for Decryption, which can be an important consideration in algorithm selection depending on available memory capacity and application-specific needs.

B. System Design, Implementation and Software Testing

Because the author's main goal is not to create an application from scratch, only to improve data protection in the Nurapay Application, the author tried to study several references related to this. There was research conducted [20], namely when they secured existing application data they used the term Proxy Database which bridges data communication, both for encryption and decryption. In Figure 9 is a Proxy Database design to bridge data communications, so the author used this as inspiration in this research with the author's own design.



Fiqure 9. Database Proxy

Based on Figure 9, the author translates it into Nodejs as a service whose task is to bridge the Nurapay Application in retrieving and storing data. In general, this bridge functions to handle POST requests containing SQL queries and encryption algorithms, process these queries, and return the results to the client, including decrypting them if necessary. Testing is carried out using the Black Box testing method, namely testing only from the outside. There are 5 features tested, that is :

- Login
- Register
- Main Menu
- Transaction History
- Product List

This test was carried out using Apache Benchmark Software, which is a technology that is commonly used to test the performance of web applications. And testing is divided into 3 parts without algorithms, the AES Algorithm and the FHE Algorithm. The following is Table 2, a summary of Nurapay Application Testing Comparison:

Table 2. Summary of Nurapay Application Testing Comparison

No	Nama Fitur	Jumlah Waktu Untuk 100 Request (second)		
		Without Algorithm	AES Algorithm	FHE Algorithm
1.	Login	2,7	26,58	40,85
2.	Register	3,63	30,96	587,92
3.	Main Menu	2,99	41,39	71,89
4.	Transaction History	3,56	65,31	101,85
5.	Product List	3,61	26,23	43,74

In general, Table 2 shows that the use of encryption algorithms, both AES and FHE, increases the time required to process requests significantly compared to without encryption. The FHE algorithm takes longer than AES, especially the "Register" feature which shows a very large increase in time. The table shows that using AES encryption is better than using FHE, only sacrificing 10 times the time compared to without encryption. If depicted in graphical form, Figure 10 is a graph of Table 2:

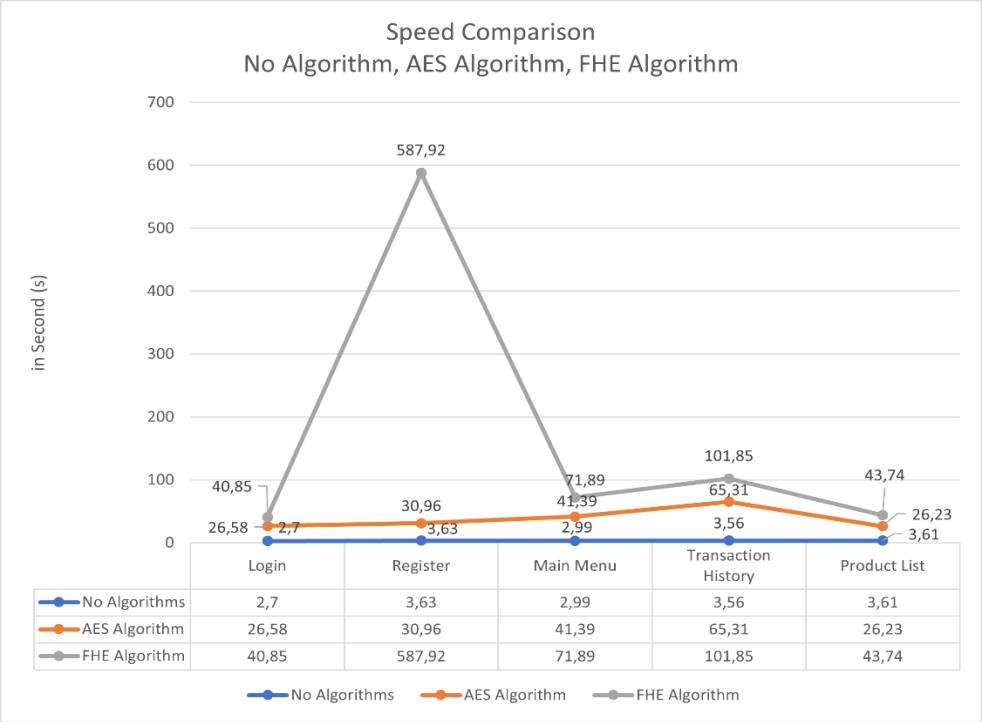


Figure 10. Speed Comparison between No Algorithm, AES Algorithm and FHE Algorithm after Implementation

Figure 10 depicts a comparison of execution speed (in seconds) for various activities in an application using three different methods: no encryption algorithm, AES algorithm, and Fully Homomorphic Encryption (FHE) algorithm. For the AES Algorithm: shows a speed decrease of 13 times compared to the Nurapay Application without the encryption and decryption process, even though there is an increase in execution time compared to without the algorithm, AES is still relatively efficient. This is because AES is a symmetric encryption method designed for efficiency and speed in the encryption and decryption process. For the FHE Algorithm: experienced a 20 times speed reduction compared to without using the encryption algorithm. This indicates the longest execution time for each activity. This can be explained by the high computational complexity inherent in FHE. FHE allows operations on encrypted data without decryption, but with the trade-off of significant computational overhead. The FHE algorithm experienced a very significant increase in the register feature by 160 times longer than without the algorithm, this is because the size of the FHE ciphertext is large, so it takes a long time to save to the database. This figure clearly shows that implementing FHE results in significant performance degradation compared to using AES. While FHE provides security benefits by allowing direct operations on encrypted data, it comes at the expense of high performance. The AES algorithm, on the other hand, shows a better balance between security and performance, with a more moderate speed reduction compared to FHE. This study highlights the importance of considering the trade-off between security and performance in selecting an encryption algorithm for a particular application. The choice of encryption algorithm must be tailored to the specific needs of the application and its operational environment. However, you need to know that references to accurate algorithm calculation results still refer to Tables 1 only shows overall performance which is mixed with efficiency technology in databases and web servers and also mixed with unprotected data. The focus in this research is only 1 Tables (customer table) which are protected as case studies in the research.

CONCLUSION

Encryption speed The AES algorithm provides a speed 1,100 times faster than FHE, for decryption the AES algorithm provides a speed 581 times faster than FHE, while for arithmetic processes the AES algorithm provides a speed 132 times faster than the FHE algorithm. CPU usage, for encryption the AES algorithm is superior because it uses 35.93% less CPU than the FHE algorithm. For decryption, the AES algorithm is

still superior with CPU usage 10.31% lower than the CPU usage of the FHE algorithm. Meanwhile, for the arithmetic process, the AES algorithm is still 9.33% lower than the FHE algorithm. Memory usage, for encryption the FHE algorithm is superior because it uses 2.3 times lower memory compared to the AES algorithm. When decrypting AES is superior in memory usage with 54 times lower memory usage compared to the FHE Algorithm. Meanwhile, for arithmetic, the AES algorithm also remains superior with 4.3 lower memory usage than the FHE algorithm. After implementation in one of the Fintech Applications, namely the Nurapay Application, it shows that the AES Algorithm is still superior to the FHE Algorithm, even though the percentage difference for the login feature, main menu, transaction history and product list is only 61.96% different between AES and FHE is because the author added an indexing technique so that data searches become faster, but there is a very striking difference in the register features which are very different, namely 1,798.96% AES is faster than FHE because the FHE register process takes a long time to save The ciphertext is 5,481% longer than the AES ciphertext.

REFERENCES

[1] C. M. Annur, "Ada 366 Anggota Asosiasi Fintech di Indonesia hingga 2022, Begini Trennya," 2023, Accessed: Jan. 18, 2024. [Online]. Available: https://databoks.katadata.co.id/datapublish/2023/08/01/ada-366-anggota-asosiasi-fintech-di-indonesia-hingga-2022-begini-trennya

[2] OWASP, "A02:2021 – Cryptographic Failures," 2021, Accessed: Mar. 07, 2024. [Online]. Available: https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

[3] L. Devi and Thangamuthu, "Cyber Security - Data Privacy and Data Security," *Cyber Security*, no. 2021, pp. 15–20, 2021.

[4] A. Calder and S. Watkins, "IT Governance, An International Guide to Data Security and ISO27001/ISO270002," 2020.

[5] E. S. Marsiani, I. Setiadi, and A. Cahyo, "Implementasi Sistem Keamanan Aes 256-Bit Gcm Guna Mengamankan Data Pribadi," 2021.

[6] Nurdin, R. Ratnadewi, and N. K. Dian R, "Sms Encryption Application Using 3Des (Triple Data Encryption Standard) Algorithm Based on Android," *IOP Conf. Series: Journal of Physics: Conf. Series* 1363 (2019) 012074, 2019.

[7] N. Nurdin, B. Bustami, M. Hutomi, M. Elveny and R. Syah, "Implementation of the BFS algorithm and web scraping techniques for online shop detection in Indonesia," *Journal of Theoretical and Applied Information Technology.*, vol. 99, no. 12, pp. 2878-2889, 2021.

[8] M. Azhari, J. Perwitosari, and F. Ali, "Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)," *Jurnal Pendidikan Sains dan Komputer*, vol. 2, no. 1, pp. 2809–476, 2022, doi: 10.47709/jpsk.v2i1.1390.

[9] G. Golovko and M. Tolochyn, "Practical Application of The Aes Encryption Method," *Control, Navigation and Communication Systems*, vol. 4, no. 2022, 2022.

[10] T. Hidayat and R. Mahardiko, "A Systematic Literature Review Method On AES Algorithm for Data Sharing Encryption On Cloud Computing," *International Journal of Artificial Intelligence Research*, vol. 4, no. 1, Apr. 2020, doi: 10.29099/ijair.v4i1.154.

[11] M. M. S. Altaee and M. Alanezi, "Enhancing cloud computing security by paillier homomorphic encryption," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1771–1779, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1771-1779.

[12] G. Ramtri and C. Patel, "Secure Banking Transactions Using RSA and Two Fish Algorithms," in *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020*, Institute of Electrical and Electronics Engineers Inc., Feb. 2020. doi: 10.1109/ic-ETITE47903.2020.236.

[13] R. Ganeshan, K. Giri, K. Reddy, A. V. S. Manikanta, and P. V Sai Lasya, "AES Algorithm for Advanced Security In Online Banking," *International Journal Of Scientific & Technology Research*, vol. 9, no. Issue 04, APRIL 2020, 2020, [Online]. Available: www.ijstr.org

[14] S. S. Muna, N. Nurdin, and T. Taufiq, "Comparative Analysis of State Universities on Website Performance in Aceh Using the PIECES Method," *Journal Of Informatics And Telecommunication Engineering*, vol. 7, no. 1, pp. 71–83, Jul. 2023, doi: 10.31289/jite.v7i1.9167.

[15] M. N. Rashid, L. H. Abed, and W. K. Awad, "Financial information security using hybrid encryption technique on multi-cloud architecture," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 6, pp. 3450–3461, Dec. 2022, doi: 10.11591/eei.v11i6.3967.

[16] T. Hidayat and R. Mahardiko, "Data Encryption Algorithm Aes By Using Blockchain Technology: A Review,"*Jurnal Dokumentasi Dan Informasi*, vol. 42, no. 1, p. 19, Nov. 2021, doi: 10.14203/j.baca.v42i1.643.

[17] U. Wahyuningsih *et al.*, "Analisis Proses Enkripsi Algoritma Kriptografi Modern Advanced Encryption Standard (Aes)," *Adijaya Jurnal Multidisiplin*, vol. 01, no. 2023, pp. 380–387, 2023, [Online]. Available: https://e-journal.naureendigition.com/index.php/mj

[18] I. Iliashenko and V. Zucca, "Faster homomorphic comparison operations for BGV and BFV. Proceedings on Privacy Enhancing Technologies," no. 3, 2021, doi: 10.2478/popets-2021-0046ï.

[19] I. Putu, A. Eka Pratama, I. Made, and S. Raharja, "Node.js Performance Benchmarking and Analysis at Virtualbox, Docker, and Podman Environment Using Node-Bench Method," 2023. [Online]. Available: www.joiv.org/index.php/joiv

[20] T. Kim, Y. Oh, and H. Kim, "Efficient Privacy-Preserving Fingerprint-Based Authentication System Using Fully Homomorphic Encryption," *Security and Communication Networks*, vol. 2020, 2020, doi: 10.1155/2020/4195852.

6% Overall Similarity

Top sources found in the following databases:

- 4% Internet database
- Crossref database
- 5% Submitted Works database
- 4% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	**papers.ssrn.com** Internet	1%
2	**University of Anbar on 2023-02-08** Submitted works	<1%
3	**Ahmed A. Asaker, Zeinab F. Elsharkawy, Sabry Nassar, Nabil Ayad, O. ...** Crossref	<1%
4	**sweetstudy.com** Internet	<1%
5	**R. Baalagi, H. Sindhura, M. Keerthi, Golda Dilip. "Optimizing Informatio...** Crossref	<1%
6	**Husni Mubarak, Arnawan Hasibuan, Adi Setiawan, Muhammad Daud. "...** Crossref	<1%
7	**University College Birmingham on 2023-03-19** Submitted works	<1%
8	**Guardiano del Faro on 2011-08-18** Submitted works	<1%

9	doi.org Internet	<1%
10	UIN Sunan Ampel Surabaya on 2019-12-17 Submitted works	<1%
11	eprint.iacr.org Internet	<1%
12	Universiti Teknologi Malaysia on 2024-01-17 Submitted works	<1%
13	en.wikipedia.org Internet	<1%
14	University Of Tasmania on 2023-04-27 Submitted works	<1%
15	University for Development Studies on 2022-03-21 Submitted works	<1%
16	Siti Safira Chairunisya Siregar, Deni, Bambang Karsono. "Virtual and F... Crossref	<1%
17	dspace.alquds.edu Internet	<1%
18	jurnal.itscience.org Internet	<1%