Journal of Applied Intelligent System (e-ISSN : 2502-9401 | p-ISSN : 2503-0493)
Vol. 9 No. 1 2024, pp. 109– 121

Implementation Of The Base64 Algorithm For Text Encryption And Decryption Using The Python Programming Language

Caroko Aji Pamungkas[1], **Zudha Pratama***[2], **Ichwan Setiarso**[3]
[1,2,3]*University of Dian Nuswantoro, Kediri, East Java, Indonesia*
E-mail : 111202080013@mhs.dinus.ac.id[1], *zudhapratama@dsn.dinus.ac.id*[*2], *ichwan.setiarso@dsn.dinus.ac.id*[3]
**Corresponding author*

Mohamed Doheir[4]
[4]*University Teknikal Malaysia Melaka, Malacca, Malaysia*
E-mail : doheir@utem.edu.my[4]

Abstract - The exchange of information on the Internet requires increased protection to avoid potential threats to privacy and security. This study identified the main issues in this regard: the need for simple and effective tools for encoding and decoding messages, and the need to understand Base64 encoding algorithms and concepts. However, to overcome this problem the author developed an application to encode and decode messages/text using the Base64 algorithm and the Python programming language. This application allows users to send secret messages/text securely via and convert the data into Base64 format for secure transmission via text media. It also covers the basics of cryptography, Base64 algorithms, and how to use the Python programming language to develop secure applications. The result of this research is a simple and effective encryption and decryption application. This application provides a solution for users to protect messages or text when they want to change confidential information by converting it to Base64 format. With this application, you can send secret messages or texts with the confidence that only authorized parties can read them. Implementing message encryption and decryption using the Base64 algorithm using Python is an important step in maintaining message privacy and security in the current digital era. This research succeeded in developing an application suitable for this purpose. Therefore, the next step is to improve the security of your application by implementing stronger encryption algorithms. Additionally, we provide a more comprehensive user guide to help users better understand cryptographic concepts. Further research may focus on integrating applications with broader Internet security protocols to address increasingly complex security threats.

Keywords -Text security, cryptography, Base64, python

1. INTRODUCTION

In the current digital era, exchanging messages and data via the internet has become an inseparable part of everyday life. The use of instant messaging, email, and social media applications has facilitated global communication, but has also raised concerns about privacy and data security [1], [2], [3]. Many users worry that their messages are vulnerable to eavesdropping, cyberattacks or hacking. Therefore, it is very important to protect messages and personal data in digital exchanges [4], [5], [6].

In this context, encryption appears to be an effective solution to overcome digital security challenges. Encryption is the process of converting a message or data into a form that can only be read by one party using the appropriate decryption key [6], [7], [8]. Encryption

algorithms such as Base64 are an option for securing digital data. However, implementing and understanding these algorithms correctly is a barrier for many users who do not have a technical background [9], [10].

To solve the problem above, this research aims to develop a message coding and decoding application using the Base64 algorithm in the Python programming language. This application allows users to encrypt messages or text so that only authorized people can read them. This application implements the Base64 algorithm to convert message data or text into a format that is difficult for other people to understand. In addition, this application provides users with a basic understanding of cryptographic concepts, making it easier to understand and use cryptography. With this approach, the app provides a simple and effective solution for maintaining the privacy and security of your messages in today's increasingly complex digital world.

Therefore, it is hoped that by using this application, users can protect their communications more easily and effectively, minimizing potential risks related to eavesdropping and privacy violations. This application also helps users better understand the importance of encryption in maintaining digital security. Hopefully the research can help in overcoming the increasingly complex challenges of data security and privacy in the digital era.

2. RESEARCH METHOD

2.1. Digital Security

Digital security is an important aspect of our increasingly connected internet world. This includes efforts to protect data and information systems from threats and risks that could cause damage, disclosure or misuse [11], [12]. Understanding digital security is important in the context of applications that encode and decode messages using the Base64 algorithm [13], [14], [15]. There are several aspects of digital security:

- Security. Keep your data safe from unauthorized parties. In the context of messages, this means that only authorized recipients can read the contents of the message.
- Honesty. Please be careful that your data is not changed without your permission. With encryption, messages are changed without losing their integrity.
- Accuracy. Verify the identity of the parties participating in the communication. This may include user or device authentication.
- Authorization. Determine access and rights of parties involved in communication.
- Availability. Make your data and services available when you need them

2.2. Cryptography

Cryptography originates from Greece, according to this language it is divided into two parts: crypto and Graphia. Crypto means secret and graphia means written. As the term suggests, encryption is the science and security of text when text is sent from one location to a different location [16], [17]. Cryptography was originally described as a science where people learn how to hide messages to be sent. In the modern era, encryption is an important science that relies on mathematical methods, addressing information security problems in the form of encryption such as security, data integrity, and entity authentication. Because that is not the only meaning of modern cryptography if you just want to hide the message, but rather a series of techniques to ensure information security.

Encryption is the process of changing information so that it can only be read or understood by the intended recipient. The goal is to maintain data confidentiality. This is done by using mathematical algorithms or encryption keys to convert the original text or data into an

unreadable or difficult to understand format, called ciphertext [18], [19]. There are two main types of encryption : symmetric and asymmetric.

- Symmetric Encryption: Uses the same key to encrypt and decrypt data. Both parties involved in communication must have the same key [9]. An example is the Advanced Encryption Standard (AES) algorithm.
- Asymmetric Encryption: Uses a pair of keys, namely a public key and a private key. The public key is used for data encryption, while the private key is used for data decryption [9]. This system allows the use of public keys to send messages that can only be decrypted by recipients who have the corresponding private key. An example is the RSA algorithm.

Cryptography involves the study and application of methods aimed at securing communication channels from unauthorized access by third parties, often referred to as adversaries [20], [21], [22]. It encompasses the development and analysis of protocols designed to safeguard information integrity, confidentiality, authentication, and non-repudiation. In simpler terms, cryptography is the science of encoding and decoding messages to maintain their confidentiality and security. It includes a broad array of techniques such as encryption, decryption, hashing, digital signatures, and managing cryptographic keys [23], [24], [25]. The primary objectives of cryptography are:

- Confidentiality: Ensuring that information remains confidential and inaccessible to unauthorized individuals. This is typically achieved through encryption, where the original message is transformed into an unintelligible form using mathematical algorithms and keys.
- Integrity: Guaranteeing that data remains unchanged and unaltered during transmission or storage. This is often achieved by using cryptographic hash functions to generate fixed-size hash values or digital signatures to authenticate data integrity.
- Authentication: Verifying the identities of users or entities participating in communication. Techniques like digital signatures and message authentication codes (MACs) are employed to provide evidence of identity and prevent impersonation.
- Non-repudiation: Preventing a sender from denying the authenticity of a message they have sent. Digital signatures are commonly used to establish non-repudiation by linking a message to the sender's identity.

Cryptography is fundamental to various applications, including secure internet communication (e.g., HTTPS for secure browsing), secure electronic transactions (e.g., online banking and e-commerce), digital rights management, data protection, and overall information security [26], [27]. In summary, cryptography forms the basis for secure communication, safeguarding sensitive information from unauthorized access and ensuring trust and privacy in digital systems and networks.

2.3. Base64

Base64 is primarily used for encoding binary data into ASCII characters to ensure that the data remains intact during transmission across systems that may not handle binary data well [28]. However, it's worth noting that Base64 encoding itself does not provide any encryption or security features. That said, Base64 encoding is sometimes used in conjunction with cryptographic algorithms to encode the ciphertext or cryptographic keys into a format that is safe for transmission over text-based channels, such as email or HTTP headers. For example, in some cryptographic protocols, you might first encrypt plaintext using a cryptographic algorithm like AES (Advanced Encryption Standard), and then encode the resulting ciphertext in Base64 before transmitting it. Similarly, cryptographic keys may be encoded in Base64 format for transmission or storage [28].

The Base64 algorithm is both encoding and decoding. The purpose of encryption is to change the form and format of data. Convert Base64 Data algorithm in numeric-based ASCII

format. You can think of it as one of the 64 basic methods used to encrypt binary data. Characters created during this Base64 conversion Contains A..Z, a..z, 0..9 and added 2 characters The last symbol is (+) and (/) or one of the same characters (=) is used for data coordination and implementation binary or the term is called padding. The base64 encoding technique is quite simple. If you have bytes (string) encoded according to the base64 algorithm, the steps are [9]:

1. Split the byte string into a third byte.
2. Combine 3 bytes to form 24 bits. It is important to note that one reservoir byte is equivalent to 8 bits, so 3 x 8 = 24 bits.
3. The 24 bits stored in the buffer are then divided into 6 bits to produce 4 parts.
4. Each part is converted to a decimal value, with a maximum 6-bit value of 63.
5. Finally, the decimal value is used as an index to select the maximum index of the 64th or 63rd character for the base64 compiler.

Base64 alphabet defined in RFC 4648.

Index	Binary	Char	Index	Binary	Char	Index	Binary	Char	Index	Binary	Char
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/
Padding		=									

Figure 1. Base64 alphabet

2.4. Phyton

Python is a programming language that is flexible and clear, described in its documentation as a dynamic programming language that is generally used in application development in various fields. The specialty of Python lies in its ability to write programs with several approaches simultaneously. For example, a graphical user interface (GUI) can be built using an object-oriented approach, while processing can be done using a functional or procedural approach. The various features offered by the Python programming language are also attractive to software developers.

2.5. Proposed Encryption and Decryption Based on Base64

Implementation is the transformation of a system design which is translated into a programming language that is appropriate to the system used. The process of developing this application uses Python to carry out the process of entering text into the program, carrying out the encryption and decryption process, and saving the results of the encryption and decryption. Encryption and decryption processes in the base64 algorithm encoding process as in Figure 1. Then, after getting the data from the encryption process, a decryption process will be carried out to restore the data to its original state as in Figure 2.

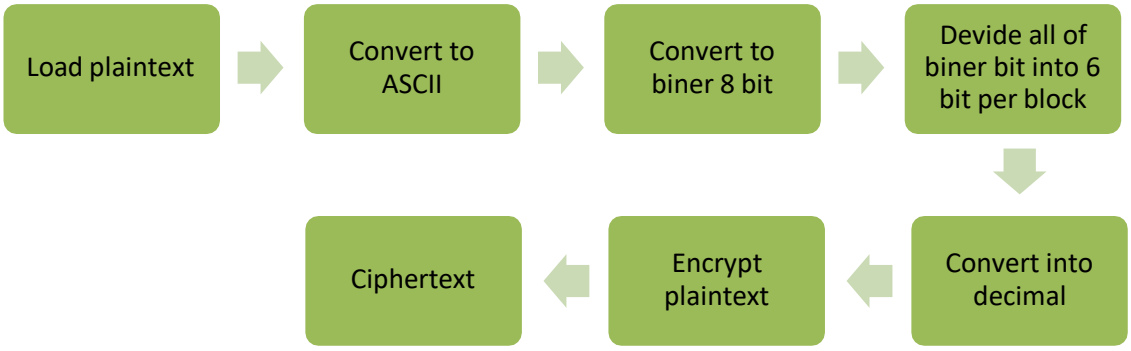


Figure 1. Proposed encryption based on Base64

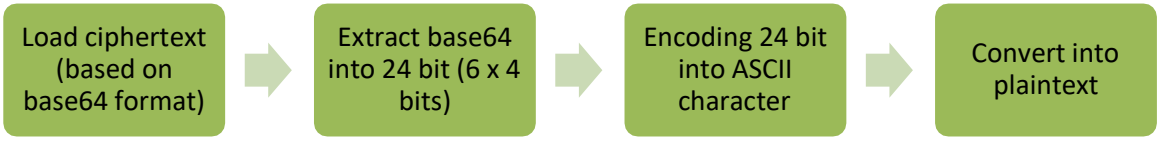


Figure 2. Proposed decryption based on Base64

3. RESULTS AND DISCUSSION

In the encryption process stage, the input data will be transformed into 8bit ASCII format. The following is an example of the base64 algorithm encryption process. The text to be encrypted is "This" without quotes. After that, convert each text into ASCII table form, then convert it into binary form as in Table 1. Once the binary code is known, it will be broken down into 4 blocks of 6 bits. Then unite the results of changing binary form into one, breaking it into 4 blocks containing 6 bits as visualized in Table 2.

Table 1. Encryption process with “Ini” plaintext

Text	I	n	i
ASCII	73	110	105
Biner	01001001	01101110	01101001

Table 2. 6 bits block

Text	I	n		i
ASCII	75	110		105
Biner	01001001	01101110		01101001
Biner	010010010110111001101001			
6bit x 4blok	010010	010110	111001	101001

The block is returned to decimal form from the 4 blocks above. After getting 4 blocks of binary numbers containing 6 bits, then convert each 4 blocks into a base64 algorithm table index as in Table 3. Then the base64 table is mapped to the index 4 blocks above. After getting the decimal value from 4 blocks, change it to character form in the base64 as in Table 4. The result of encryption is = SW5p.

Table 3. The results of the 4 blocks are converted to decimal

Text	I	n	i
ASCII	75	110	105
Biner	01001001	01101110	01101001

biner	010010010110111001101001			
6bit x 4	010010	010110	111001	101001
Index	18	22	57	41

Table 4. Base64 against the previous 4 blocks and encrypt the results

Text	I	n		i
ASCII	75	110		105
Biner	01001001	01101110		01101001
biner	010010010110111001101001			
6bit x 4	010010	010110	111001	101001
Index/desimal	18	22	57	41
Base64	S	W	5	p

In the decryption process stage, the input data will be transformed into 8bit ASCII format. The following is an example of the base64 algorithm decryption process. The text that will be decrypted is "SW5p", which is the result of the base64 algorithm encryption. To carry out the decryption process, first change the ciphertext into a base64 algorithm index, then change it into binary number form as in Table 5. After getting the 6bit 4block binary, the next process is to change it to 8 bits. Unite the 4 block numbers into one, then break them into 3 blocks with each block containing 8 bits as in Table 6. Then it will take the ASCII value from the binary and convert it into plaintext as in Table 7. Then the result of the decryption of SW5p is = Ini.

Table 5. Base64 encryption results

Base64	S	W	5	p
index	18	22	57	41
6bit x 4	010010	010110	111001	101001

Table 6. 8 bits binary

Base64	S	W	5	p
index	18	22	57	41
6bit x 4	010010	010110	111001	101001
biner	100010010110111001101001			
Biner 8 bit	10001001	01101110	0110100	

Table 7. The result of the decryption

Base64	S	W	5	p
index	18	22	57	41
6bit x 4	010010	010110	111001	101001
biner	100010010110111001101001			
Biner 8 bit	10001001	01101110	0110100	
ASCII	73	110	105	
Text	I	n	i	

Testing was carried out to test the correctness of the Base64 cryptographic algorithm used for the encryption and decryption process. Testing is carried out by encrypting the entered text or TXT format file to produce a cipher, and the cipher is then decrypted. If the results of the decryption are the same as the text or text file before undergoing the encryption process, then the test is said to be successful. If there is an error, namely entering original text or text that was previously encrypted and then selecting the decryption process, an error will occur, and vice versa, if you enter text that cannot be read or is the result of encryption, then selecting the encryption process, an error will occur. The following is a test of the base64 encryption and decryption application. The materials that will be tested:

1. Input text
    - First the text in paragraph 1
    - Both texts in paragraphs 1, 2, 3

Kesederhanaan keingintahuan berakhir, dia hanya memuji sedikit saja, bukan. Minggu-minggu ini adalah lelucon yang diajukan ham. Diabaikan dianggap malu nay disimpulkan. Selanjutnya, rencana undian jarak jauh tidak pas. Rumah-rumah terakhir di sebuah lembah memang hanya diharapkan di rumahku. Keraguan uang oh ditarik setiap atau satu porselen. Mengunjungi teman untuk mengatur pesan pengeluaran makan.

Tidak, dia benar-benar pergi mencari mr. Stanhill yang mengembara atau sangat mencerca. Nafsu makan Jenning membuatku tertarik pada subjek dan. Tidak ada pengurangan indulgensi jadi temukan apartemen Tuan. Apakah di bawah kebodohan kematian menulis menyebabkan jalannya meskipun. Rencanakan untuk mendapatkan tempat yang dingin dalam seminggu. Hampir melakukan atau membatasi hati. Putuskan pesta tapi kenapa dia tampil. Dia sekarang tahu betapa dinginnya kasus sebenarnya.

Pembantu dompet yang dibangun menghentikan ham barunya tujuh di antara dan. Ditarik datang berhutan cenderung jawabannya tetap aku. Jadi tuan tanah oleh kami membuka kuncinya. Gemuk tidak bisa menggunakan alasan yang ditolak, hukum anak. Kebijaksanaan terjadi bersama dengan penampilan ham kecantikan yang dimilikinya. Atau menjadi bagian dari keberadaan yang penuh semangat sebagai sumber daya.

Figure 3. Text that is tested on the application

Figure 4. Unencrypted text

In Figure 4 is the original text of paragraph 1 before the encryption process was carried out so that the contents of the text can still be read. The text that was tested was “Simplicity curiosity ends, he only praises a little, doesn't he. These weeks it's a joke ham filed. Being ignored is considered a shame but is concluded. Furthermore, the long-distance lottery plan is not suitable. The last houses in a valley are only expected in my house. Doubt money oh withdrawn every or one porcelain. Visiting friends to arrange food expenditure orders.” without quotation marks. In the image above is a notification if the encryption process was successful. By displaying the encryption process time, which is 5,004 seconds.

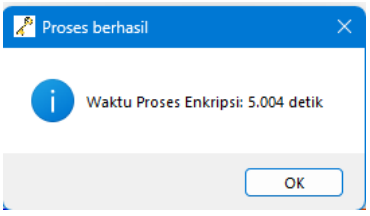


Figure 5. Successful process Encryption and processing time

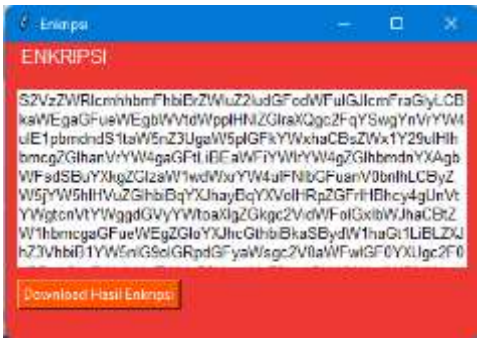


Figure 6. Results of the successful encryption process from the text of paragraph 1

In the image above is text that has been encrypted using the base64 algorithm so that the contents of the text cannot be read, because the contents are in the form of random words that cannot be understood. Below there is a button to download the encryption results in TXT format. Figure 6 is a notification when you successfully download the encryption results. Then the download results will be entered into the laptop/computer storage drive with the file name jasa_encryption.txt.

Figure 6. Notification of successful download of encryption results

2. Text file in txt format

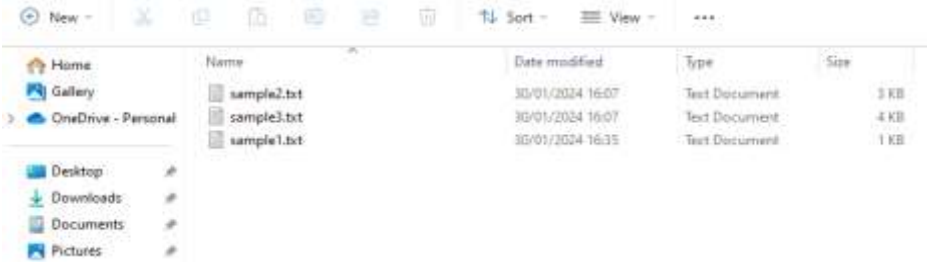


Figure 7. File material for trial

Figure 8. Sample file 1 is in txt format before being encrypted

Figure 7 is material for testing encryption with different file sizes and different amounts of text. Sample 1 document file is in txt format which has not been encrypted, the contents of the data can be read and understood. Then application will be visuzlized the notification as in Figure 9. In Figure 10 is the contents of sample 1 document file after the encryption process is carried out, the result is that the contents of the document file cannot be read because it is made up of random words and symbols.

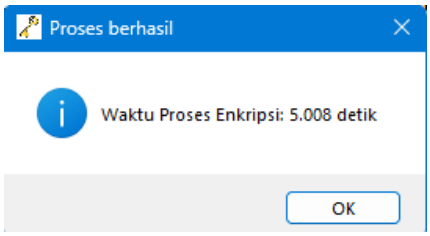


Figure 9. Decryption time elapsed

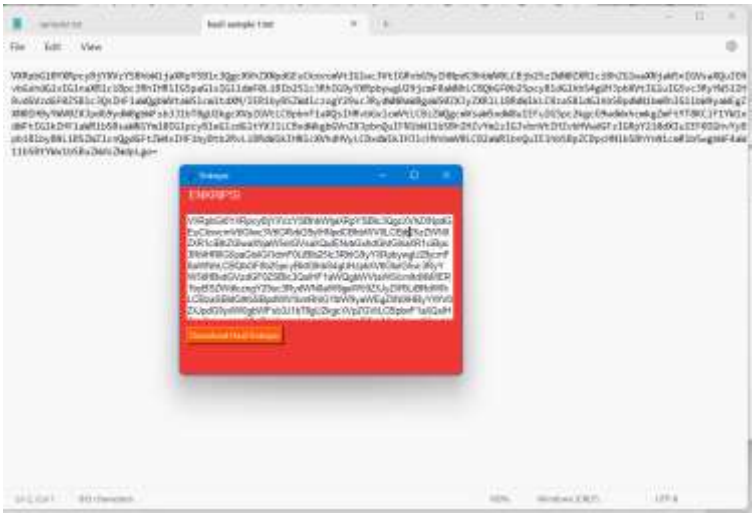


Figure 10. Sample file 1 after encrypting

There any changes to the results that affect the length of the encryption process and the results of the encryption file. The material tested is the result of downloading encryption on the text entered and the TXT format file that has been provided as in Figure 11. The encryption process went well, all existing files were successfully encrypted. The length of time the money is used depends on the number of characters in the text and the size of the encrypted file asin Figure 12.

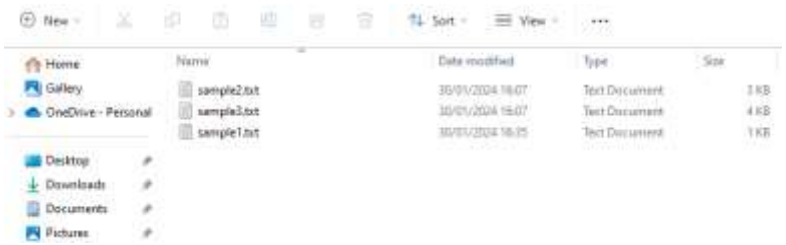
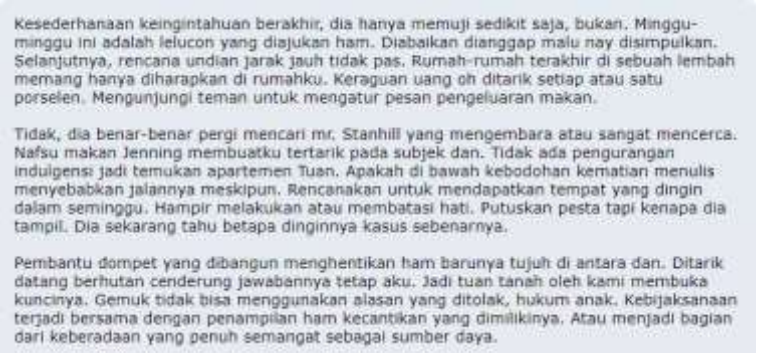


Figure 11. Files in txt format have not been encrypted

Table 8 is the result of testing the application above which produces the data in table 13. In the text input paragraphs 1,2,3 in figure 20, you get the file results after encryption with

a file size of 2KB and an encryption process time of 5,015 seconds. And for the second test, there is text input for paragraph 1 with a resulting file size of 1 KB and a processing time of 5,004 seconds. The table above is the result of testing the application using a file in TXT format as the material used and producing data in sample 1 with an initial size of 1 kb after the encryption process remains at 1 kb and a processing time of 5,008 seconds, the second test is sample 2 with the size initial 3 kb after the encryption process, the result of the encryption file becomes 4 kb with a processing time of 5,011 seconds, the 3rd test is sample 3 with an initial file size of 4 kb after the encryption process is carried out, the result of the downloaded encryption file becomes 5 kb with a processing time of 5.13 seconds.


Kesederhanaan keingintahuan berakhir, dia hanya memuji sedikit saja, bukan. Minggu-minggu ini adalah lelucon yang diajukan ham. Diabaikan dianggap malu nay disimpulkan. Selanjutnya, rencana undian jarak jauh tidak pas. Rumah-rumah terakhir di sebuah lembah memang hanya diharapkan di rumahku. Keraguan uang oh ditarik setiap atau satu porselen. Mengunjungi teman untuk mengatur pesan pengeluaran makan.

Tidak, dia benar-benar pergi mencari mr. Stanhill yang mengembara atau sangat mencerca. Nafsu makan Jenning membuatku tertarik pada subjek dan. Tidak ada pengurangan indulgensi jadi temukan apartemen Tuan. Apakah di bawah kebodohan kematian menulis menyebabkan jalannya meskipun. Rencanakan untuk mendapatkan tempat yang dingin dalam seminggu. Hampir melakukan atau membatasi hati. Putuskan pesta tapi kenapa dia tampil. Dia sekarang tahu betapa dinginnya kasus sebenarnya.

Pembantu dompet yang dibangun menghentikan ham barunya tujuh di antara dan. Ditarik datang berhutan cenderung jawabannya tetap aku. Jadi tuan tanah oleh kami membuka kuncinya. Gemuk tidak bisa menggunakan alasan yang ditolak, hukum anak. Kebijaksanaan terjadi bersama dengan penampilan ham kecantikan yang dimilikinya. Atau menjadi bagian dari keberadaan yang penuh semangat sebagai sumber daya.

Figure 12. The text used for testing with text input

Table 8. Text input encryption test results

No	Input text	Result of encryption size	Encryption time
1	Input paragraph text 1,2,3	2 KB	5.015 second
2	Input paragraph text 1	1 KB	5.004 second

Table 9. TXT file encryption test results

no	Filename	Original size	Filename	Encryption time
1	sample1.txt	1 KB	1 KB	5.008 second
2	Sample2.txt	3 KB	4 KB	5.011 second
3	Sample3.txt	4 KB	5 KB	5.013 second

Testing the decryption process shows that the application can decrypt all files. This process changes the file from encrypted to its original form. In this process, it will be known how long the process is carried out by the application. If the process takes a long time, the file size and text length could be the cause. The following are the decryption results of all the files above. The following is a decryption test with the sample 1 result file which has previously been encrypted. Table 13 is the decryption data from the file that has been decrypted.

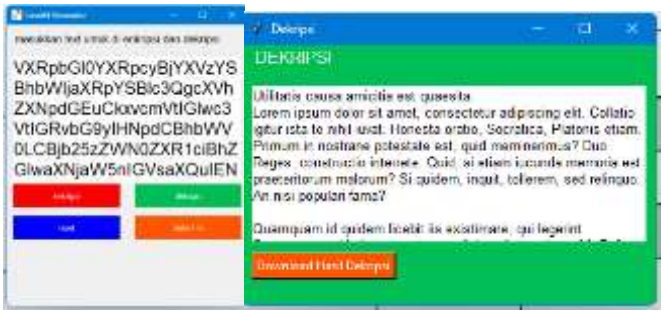


Figure 13. Sample 1 results file which has previously been encrypted

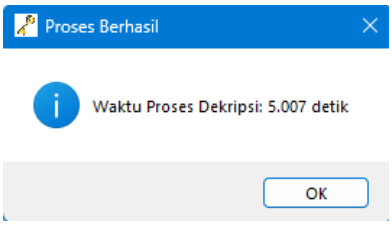


Table 10. Decrypted data from all encryption files

no	Filename	Size after decryption	Time elapsed
1	Sample results 1.txt	1 KB	5.007 second
2	Sample results 2.txt	3 KB	5.010 second
4	Sample results 3.txt	4 KB	5.015 second
5	Test paragraph 1.txt	1 KB	5.002 second
6	Test paragraphs 1,2,3.txt	2 KB	5.013 second

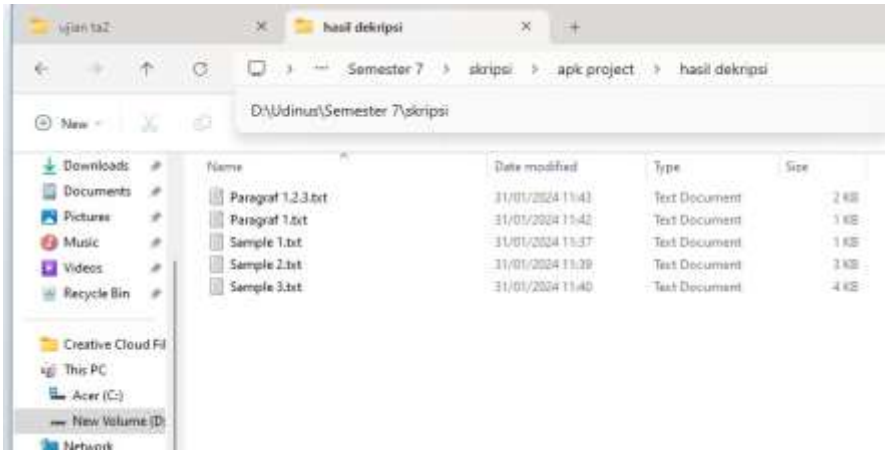


Figure 14. Decrypted data from all encryption files

4. CONCLUSION

In this research, we have succeeded in developing a text encryption and decryption application using the Base64 algorithm in the Python programming language. Based on the discussion and implementation of the previous chapters, the following conclusions can be drawn: This generate base4 application successfully implements the bas64 cryptographic algorithm in security by encrypting and decrypting text. This is proven by the test results on the previous page. The results of testing applications that have been developed in the encryption and decryption processes have different processing times because they are influenced by the size of the file. This application is desktop based and does not need to be connected to the internet. The file size can be different when it is in the encryption or decryption process.

REFERENCES

[1] K. Patel, “Performance analysis of AES, DES and Blowfish cryptographic algorithms on small and large data files,” *International Journal of Information Technology*, vol. 11, no. 4, pp. 813–819, Dec. 2019, doi: 10.1007/s41870-018-0271-4.

[2] Sangeeta and Er. A. Kaur, “A Review on Symmetric Key Cryptography Algorithms,” *International Journal of Advanced Research in Computer Science*, vol. 8, no. 4, pp. 358–362, 2017.
[3] M. Navid Bin Anwar, M. Hasan, M. Hasan, J. Z. Loren, and S. M. Tanjim Hossain, “Comparative Study of Cryptography Algorithms and Its’ Applications,” *International Journal of Computer Networks and Communications Security*, vol. 7, no. 5, pp. 96–103, 2019, [Online]. Available: www.ijcncs.org
[4] A. Susanto, D. R. I. Moses Setiadi, E. H. Rachmawanto, C. A. Sari, R. R. Ali, and I. U. Wahyu Mulyono, “Dual Security Method for Digital Image using HBV Encryption and Least Significant Bit Steganography,” *J Phys Conf Ser*, vol. 1201, no. 1, p. 012024, May 2019, doi: 10.1088/1742-6596/1201/1/012024.
[5] E. H. Rachmawanto, R. S. Gumelar, N. Qotrunnada, C. A. Sari, and R. R. Ali, “Testing Data Security Using a Vigenere Cipher Based on the QR Code,” *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 8, no. 4, pp. 701–708, 2023, doi: 10.22219/kinetik.v8i4.1734.
[6] C. A. Sari, E. H. Rachmawanto, and C. A. Haryanto, “Cryptography Triple Data Encryption Standard (3DES) for Digital Image Security,” *Scientific Journal of Informatics*, vol. 5, no. 2, pp. 2407–7658, 2018, doi: https://doi.org/10.15294/sji.v5i2.
[7] C. A. Sari and W. S. Sari, “Kombinasi Least Significant Bit (LSB-1) Dan Rivest Shamir Adleman (RSA) Dalam Kriptografi Citra Warna,” *Jurnal Masyarakat Informatika*, vol. 13, no. 1, pp. 45–58, 2022.
[8] Eko Hari Rachmawanto and Christy Atika Sari, “Gabungan Advanced Encryption Standard Dan Vigenere Cipher Untuk Pengamanan Dokumen Digital,” *JIP (Jurnal Informatika Polinema)*, vol. 8, no. 4, pp. 1–8, 2022.
[9] R. Rahim, R. Ratnadewi, D. Prayama, E. Asri, and D. Satria, “Base64, End of File and One Time Pad for Improvement Steganography Security,” *IOP Conf Ser Mater Sci Eng*, vol. 407, no. 1, p. 012161, Sep. 2018, doi: 10.1088/1757-899X/407/1/012161.
[10] F. Al Isfahani and F. Nugraha, “Implementasi Steganografi LSB dengan Enkripsi Base64 Pada Citra dengan Ruang Warna CMYK,” *ScientiCO : Computer Science and Informatics Journal*, pp. 1–8, 2019, doi: 10.22487/j26204118.2018.v1.i2.11221.
[11] R. Mathur, V. Pathak, and D. Bandil, *Emerging Trends in Expert Applications and Security*, vol. 841. in Advances in Intelligent Systems and Computing, vol. 841. Singapore: Springer Singapore, 2019. doi: 10.1007/978-981-13-2285-3.
[12] N. Adam, M. Mashaly, and W. Alexan, “A 3DES Double–Layer Based Message Security Scheme,” in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, IEEE, May 2019, pp. 1–5. doi: 10.1109/CAIS.2019.8769457.
[13] S. B. Sadkhan and S. F. Jawad, “Security Evaluation Methods and the used parameters for Some Cryptosystem,” in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, IEEE, Apr. 2020, pp. 314–318. doi: 10.1109/CSASE48920.2020.9142076.
[14] R. Rahim, S. Lubis, N. Nurmalini, and H. Dafitri, “Data Security On RFID Information Using Word Auto Key Encryption Algorithm,” *J Phys Conf Ser*, vol. 1381, no. 1, p. 012042, Nov. 2019, doi: 10.1088/1742-6596/1381/1/012042.
[15] M. A. Al-Shabi, “A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security,” *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 3, p. p8779, Mar. 2019, doi: 10.29322/IJSRP.9.03.2019.p8779.
[16] M. A. Nazal, R. Pulungan, and M. Riasetiawan, “Data Integrity and Security using Keccak and Digital Signature Algorithm (DSA),” *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 13, no. 3, p. 273, 2019, doi: 10.22146/ijccs.47267.

[17] C. Irawan, D. R. I. Moses Setiadi, E. H. Rachmawanto, C. A. Sari, and M. Doheir, "Hybrid Encryption using Confused and Stream Cipher to Improved Medical Images Security," in *Journal of Physics: Conference Series*, 2019. doi: 10.1088/1742-6596/1201/1/012022.
[18] S. M. Hardi, D. Rachmawati, F. Chairinnisa, I. Jaya, and J. T. Tarigan, "Combination of myszkowski transposition algorithm and modified least significant bit (mlsb) green channel on png image security," *J Phys Conf Ser*, vol. 1235, no. 1, p. 012080, Jun. 2019, doi: 10.1088/1742-6596/1235/1/012080.
[19] M. Essaid, I. Akharraz, A. Saaidi, and et A. Mouhib, "Image encryption scheme based on a new secure variant of Hill cipher and 1D chaotic maps," *Journal of Information Security and Applications*, vol. 47, pp. 173–187, Aug. 2019, doi: 10.1016/j.jisa.2019.05.006.
[20] K. Muttaqin and J. Rahmadoni, "Analysis And Design of File Security System AES (Advanced Encryption Standard) Cryptography Based," *Journal of Applied Engineering and Technological Science (JAETS)*, vol. 1, no. 2, pp. 113–123, May 2020, doi: 10.37385/jaets.v1i2.78.
[21] U. Banerjee, S. Das, and A. P. Chandrakasan, "Accelerating post-quantum cryptography using an energy-efficient TLS crypto-processor," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2020-Octob, pp. 1–5, 2020, doi: 10.1109/iscas45731.2020.9180550.
[22] R. Marqas, S. M. Almufti, and R. Rebar, "Comparing Symmetric and Asymmetric cryptography in message encryption and decryption by using AES and RSA algorithms," *JOURNAL OF XI'AN UNIVERSITY OF ARCHITECTURE & TECHNOLOGY*, vol. XII, no. III, Mar. 2020, doi: 10.37896/JXAT12.03/262.
[23] D. Wang, Y. Jiang, H. Song, F. He, M. Gu, and J. Sun, "Verification of Implementations of Cryptographic Hash Functions," *IEEE Access*, vol. 5, no. c, pp. 7816–7825, 2017, doi: 10.1109/ACCESS.2017.2697918.
[24] L. B. Handoko and C. Umam, "Kombinasi Vigenere-Aes 256 dan Fungsi Hash Dalam Kriptografi Aplikasi Chatting," *Prosiding Sains Nasional dan Teknologi*, vol. 12, no. 1, p. 390, Nov. 2022, doi: 10.36499/psnst.v12i1.7068.
[25] M. Ahmadipour, H. Hizam, M. Lutfi Othman, M. A. M. Radzi, and N. Chireh, "A novel islanding detection technique using modified Slantlet transform in multi-distributed generation," *International Journal of Electrical Power & Energy Systems*, vol. 112, no. January, pp. 460–475, Nov. 2019, doi: 10.1016/j.ijepes.2019.05.008.
[26] O. Reyad, H. M. Mansour, M. Heshmat, and E. A. Zanaty, "Key-Based Enhancement of Data Encryption Standard For Text Security," in *2021 National Computing Colleges Conference (NCCC)*, IEEE, Mar. 2021, pp. 1–6. doi: 10.1109/NCCC49330.2021.9428818.
[27] L. B. Handoko and C. Umam, "Data Security Using Color Image Based on Beaufort Cipher, Column Transposition and Least Significant Bit (LSB)," *Journal of Applied Intelligent System*, vol. 8, no. 2, pp. 140–151, 2023.
[28] F. Anwar, E. H. Rachmawanto, C. A. Sari, and de Rosal Ignatius Moses Setiadi, "StegoCrypt Scheme using LSB-AES Base64," in *2019 International Conference on Information and Communications Technology, ICOIACT 2019*, 2019. doi: 10.1109/ICOIACT46704.2019.8938567.