Journal of Computer Science, Information Technology and Telecommunication Engineering (JCoSITTE)
Vol. 6, No. 1, March 2025, pp. 800~806
ISSN: 2721-3838, DOI: 10.30596/jcositte.v6i1.22746

Advanced Encryption Standard (AES) Cryptography Application Design

Allwine[1], Sandi Badiwibowo Atim[2], Muhammad Afdhaluddin[3]
[1,2,3]Department of Computer Science, University of Lampung, Indonesia
[1,2,3]Faculty of Mathematics and Natural Sciences, University of Lampung, Indonesia

ABSTRACT

As technology advances, the need for secure data transmission and storage increases. Encryption and decryption are essential processes to ensure data confidentiality and integrity. Encryption transforms original data into unreadable form during transmission, while decryption restores it to its original state for the recipient. This guarantees that unauthorized parties cannot access the data. Cryptosystems have evolved over time, and with the rapid growth of communication technologies, stronger standards are needed. AES (Advanced Encryption Standard), based on the Rijndael algorithm, has become the current standard for encryption. AES can encrypt and decrypt 128-bit data blocks with key lengths of 128, 192, or 256 bits, addressing the limitations of older algorithms and providing enhanced data security to protect confidentiality in modern cryptosystems.

Keyword : AES; Encryption – Decryption; Data confidentiality; Cryptosystem.

Corresponding Author:
Allwine,
Department of Computer Science
University of Lampung
Jalan Prof. Dr. Ir. Sumantri Brojonegoro No.1, Kota Bandar Lampung, 35141, Indonesia.
Email : allwine@fmipa.unila.ac.id

Article history:
Received Jan 11, 2025
Revised Jan 15, 2025
Accepted Mar 10, 2025

1. INTRODUCTION

The advancements in computers and the swift processors that have developed over the last ten years mean that compromising any system is simply a matter of time. (Ekert, Artur K; Huttner, Bruno; Palma, 1994). A crucial element of communication through computers and networks is the protection of messages, data, or information throughout the data exchange process. This is one of the driving factors behind the emergence of cryptographic technology. Cryptography is based on data encoding algorithms that support two main aspects of information security: secrecy (protection of data confidentiality) and authenticity (protection against falsification and unauthorized modification of data).

Cryptography encompasses both the discipline and the skill of guaranteeing the safety of communications. The protection of a message is accomplished by converting it into a format that bears no understandable significance (Allwine, A., & Sitorus, J. H. P. 2019). Cryptography is a branch of mathematics related to information security aspects such as data integrity, entity authenticity, and data authenticity. Cryptography is the process of encoding or scrambling confidential messages so that their meaning remains unclear to those who attempt to intercept them (Vasanth & Dhikhi, 2016).

Digital steganography employs digital formats as a medium, including images, audio, written text, and video. The concealed secret information can be any type of file (Hutabarat, A., & Sawitri, R. 2024). Cryptography employs various techniques to secure data(Abood & Guirguis, 2018). Data transmission and storage via electronic media require a mechanism that ensures both the security and integrity of the transmitted data(Aldossary & Allen, 2016). This data must remain confidential during transmission and intact when received at the destination. To ensure this, encoding (encryption and decryption) processes are applied to the data being sent.

Encryption occurs during transmission by converting the original data into an unreadable form, while decryption occurs upon receipt by converting the encrypted data back into its original form(Vasanth & Dhikhi, 2016). Thus, the data sent during the transmission process is in an encrypted state, so unauthorized parties cannot understand the original data. Only the recipient with the secret key can decrypt the data and restore it to its original form.

Encryption can be understood as a code or cipher(Sarkar & Noel, 2020). A coding system uses a table or dictionary that has been defined for words of the information being transmitted, or parts of

the message. A cipher uses an algorithm to transform the entire stream of bits from the original message (plaintext) into an unintelligible cryptogram(Bucerzan & Cr, 2010). Since cipher systems are designed to be automated, this technique is widely used in computer network security systems.

In 1977, the National Institute of Standards and Technology (NIST) first announced the Data Encryption Standard (DES) as a data encoding standard(Rabah, 2005). The strength of DES lies in its key length of 56 bits. However, with the increasing speed of hardware and the widespread use of distributed computer networks, the use of DES proved to be inadequate, especially for data transmission over the internet. Special hardware that could determine the 56-bit DES key in a matter of hours was already being built. As a result, there was a need for a new algorithm with longer and more secure keys. Triple-DES emerged as an alternative solution to address these issues, although its encryption speed was considered too slow for some applications.

Implementing AES in these operational modes brings a range of benefits and challenges related to the levels of data security.(Rabah, 2005) AES was designed to take over from DES and Triple-DES, which had been used for encrypting electronic data for a long time. After several rounds of selection, the Rijndael algorithm was selected as the cryptographic algorithm for AES in 2000.

AES is a symmetric cryptographic algorithm that works in block cipher mode, handling data blocks of 128 bits. It supports key lengths of 128 bits (AES-128), 192 bits (AES-192), or 256 bits (AES-256)(Alenezi et al., 2020). The AES block cipher algorithm can function in various modes, such as Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), and Output Feedback (OFB) (Bujari & Aribas, 2017). Using AES in these operational modes offers a range of advantages and challenges related to the security levels of data. (Stergiou et al., 2018).

2. Research Stages

2.1 AES

As per the authors El Adib and Raissouni, the states, input and output blocks of the AES algorithm to be 128 bits. This then means that Nb is catering for 4 bytes since the data size is also 128 bits(El Adib & Raissouni, 2012). Thus, the total number of potential keys is equal to 3.4 times 10^38 which in turn equals to 2^128 given that the key size is configured at 1228 bits. With the assumption in place that the fastest computers can look through a million keys per second, the total time to go through every possible key would equal 5.4 times 10^24. In the instance where the computers attempt to go through mllion keys after each millisecond, this would only increase to 5.4 times 10^18. In the instance where an AES algorithm is being used, special care must be taken when modyfing the input over blocks of size 128 bits. The AES algorithm alone has different configurations for the cipher key, the options are 128,192 or 256 bits. It is worth noting that the size of key for the AES Faram increases the complexity making it more harder for potential attackers (Fouque et al., 2014).

Table 1. Comparison of the Number of Rounds and Key Length

	Nk words	Nb words	Nr
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

2.2 System Design

The design of a cryptographic app that employs the Advanced Encryption Standard (AES) algorithm is intended to increase security in data transmission or exchange. (Kawle et al., 2014). This goes through a series of stages, beginning with the generation of hierarchical diagram and then evolves to State Transition Diagram (STD). Before this, I have elaborated a Flowchart and a System Development Life Cycle(SDLC) was run during the process Initially, the user interface design gets implemented and then to execution. Post system design, we move into the application development phase and finish up with testing to make sure that our program works as intended..

2.3 Hierarchical Diagram Design

It aims at bettering the design used to enhance the development of a cryptosystem application that uses AES for encryption. There is a hierarchical structure with four main submenus: Encrypt, Decrypt, About, and Help. The Encrypt submenu supports various options, including New, encryption process, Save, and Open. Similarly, the Decrypt submenu follows the same layout as Encrypt, including New, decryption process, Save, and Open.

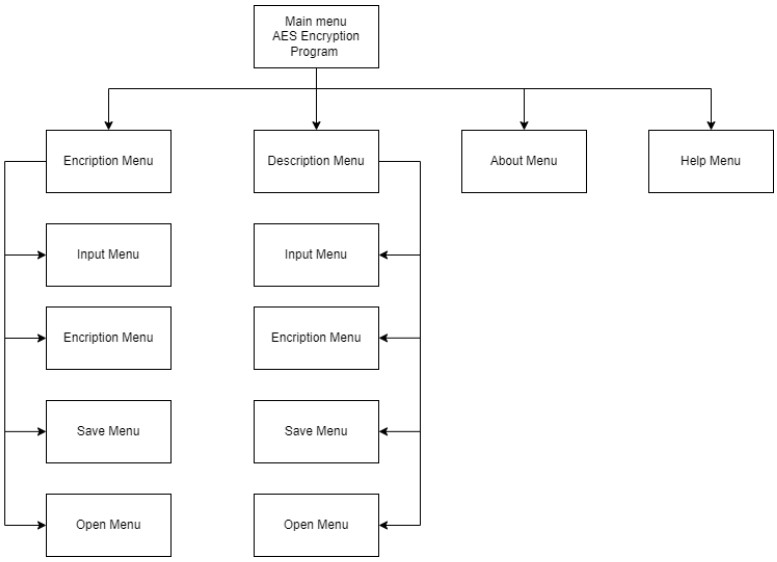

Fig 1. Hierarchical Diagram Design

2.4 State Transition Diagram Design

State diagrams effectively illustrate the various states of an interactive system and the transitions that occur between them, offering a clear and organized method to model user interactions (Parnas, 1969). This design is used to analyze what occurs in the system when transitions happen between states, what triggers these changes, and what the outcomes are as a result of those transitions.

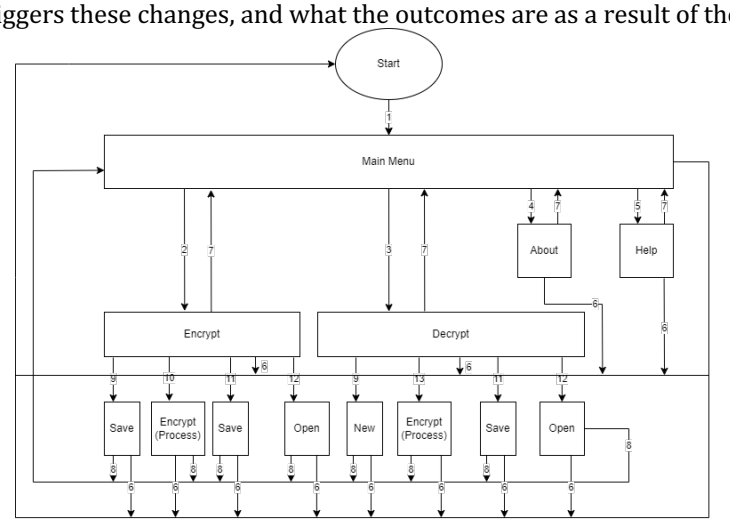

Fig 2. Transition Diagram Design

2.5 State Flowchart Design

State flowchart is a type of diagram that describes an embedded system process, showing the steps as boxes of various kinds and their orders by connecting them with arrows. It is popularly used to describe the embedded process before actual code is written. This allows programmers to see the path of their creations, discover important data, and gain control over the flow of inputs and outputs (Wang & Wayne, 2014). This design is used to design and represent the system's workflow in the form of a flowchart. Before the program is developed, the flowchart helps programmers plan and understand the logical flow of the system. After the program is completed, the flowchart serves as documentation that facilitates explaining the system's workflow to users or other involved parties.

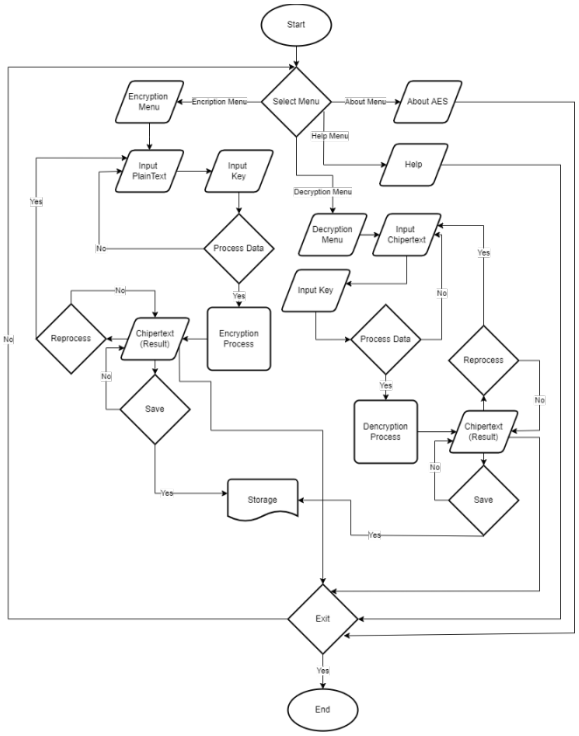

Fig 3. Flowchart Design

2.6 Encryption Module Design

This design includes a frame message to hold the text before encryption. The module is located in the toolbar at the top of the application, containing buttons for key functions. When the Encrypt toolbar is activated, the buttons you can use include New, Process Encrypt, Open, Save, and Back to Menu.

2.7 Dencryption Module Design

This design features a frame message along with a toolbar that includes buttons for New, Open, Save, Process Decrypt, and Back to Menu. The New button allows users to create a blank frame message, and if there is already text present, it changes to Clear, enabling the deletion of the message. The Process Decrypt button is used to restore the encrypted message by entering the key agreed upon by the two parties. If the key entered is incorrect, the message would remain obscured; if it is correct, the original message would be revealed. The Save button is used for the purpose of storing the decrypted message, while the Open button is used by the user to access a previously saved file.

2.8 About Module Design

The Whereabouts module integrates a brief overview of the cryptosystem program and its producers. It is accessible to all users through the About option in the primary menu present at the top of the application. This becomes a pinnacle of information for users as to what the program mainly strives to achieve, how it would function, and why it has become a necessity.

2.9 Help Module Design

The Help module is where the detailed explanation of how to work with the software is and what functions are included with the cryptosystem application. The module comes to screen when users click the Help toolbar from the main menu at the top of the application. It gives thorough guidance in enabling them to easily and efficiently understand how to use the program as well as its features.

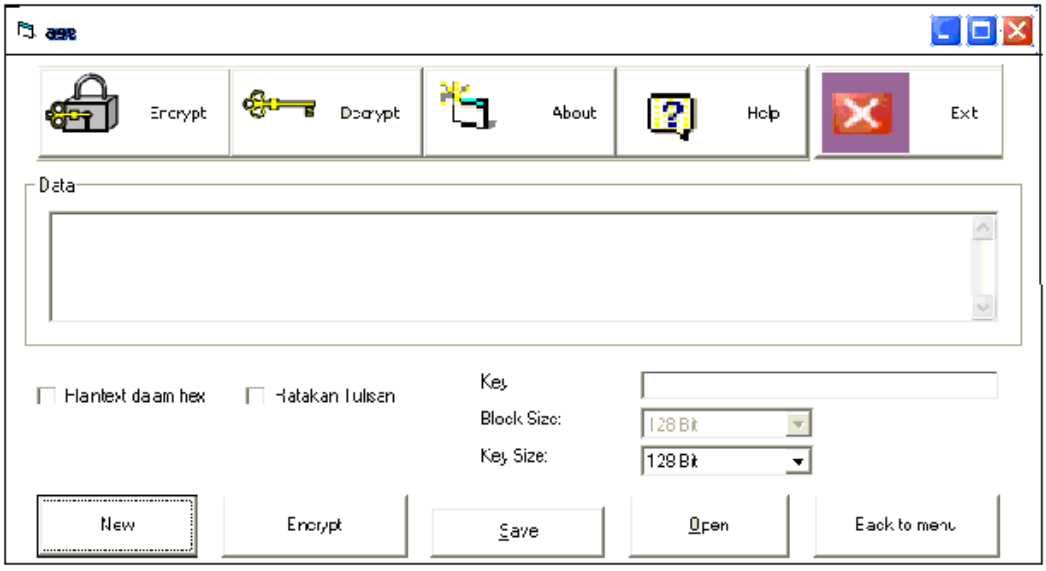

Fig 4. Interface Design

3. RESULTS AND DISCUSSION

The testing method applied is black-box testing, meaning the program takes input and gives an output. This output is then examined against the design specifications and user requirements. If the output is aligned with the user requirements and specifications, then the program is deemed to be correct, and thus no changes are needed. If the output does not match the specifications, the program contains errors and requires modifications. The process is repeated until the program produces the correct output as per the design specifications and user needs.

3.1 Encryption Module Testing

Encryption module serves to scramble messages or information, transforming them into an unreadable format for others, thereby creating a secret message. Function of encryption module in this program operates according to the design specifications.

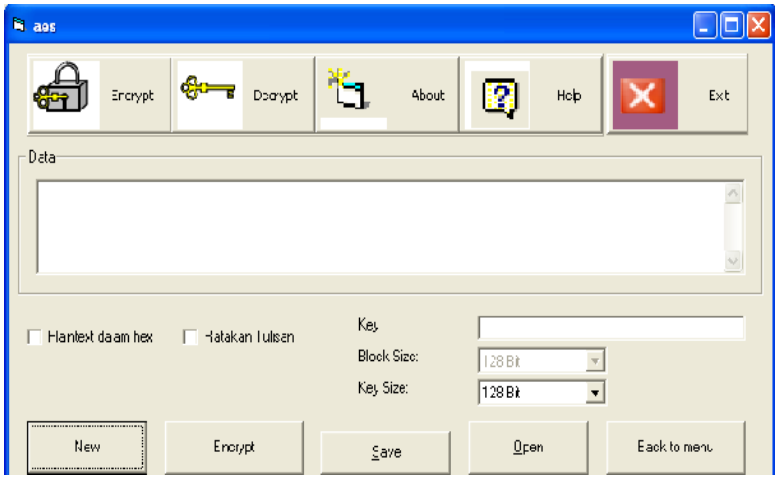

Fig 5. Encryption Module

3.2 Decryption Module Testing

Decryption module works to convert scrambled messages back into a readable format, allowing the intended recipient or authorized parties to access the content. The decryption module in this program functions in accordance with the design specifications.

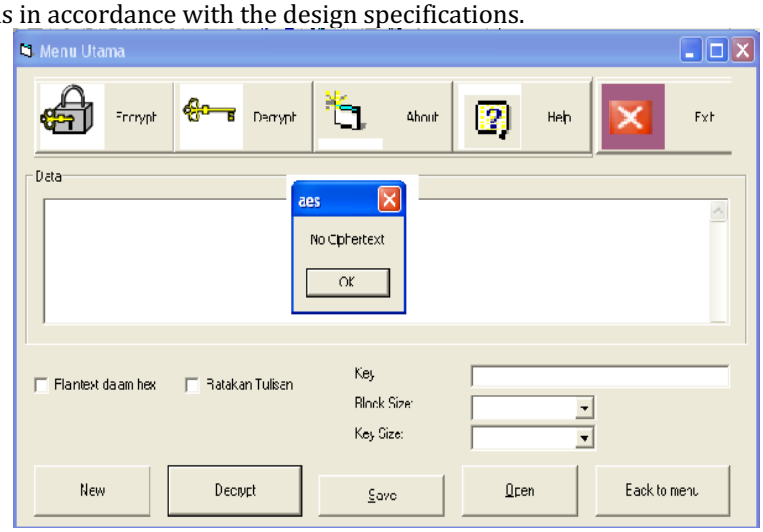

Fig 6. Decryption Module

3.3 About Module Testing

This module provides brief information about the application and its developers. The function of the About module in this application works properly and aligns with the specifications outlined in the design.

Fig 7. About Module

3.4 Help Module Testing

The Help module functions as a comprehensive guide to assist users in operating this application program. Each of its features works properly and aligns with the specifications set out in the program design.

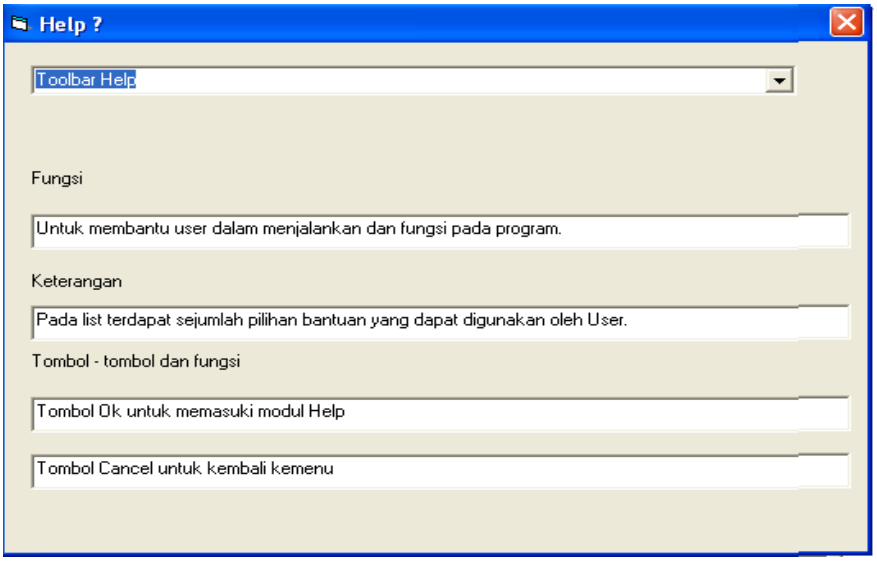

Fig 8. Help Module

4. CONCLUSION

From the design and creation of the cryptosystem application utilizing the Advanced Encryption Standard (AES) algorithm, the subsequent conclusions may be inferred:

1. The application functions as expected according to the technical specifications designed.
2. This cryptosystem application restricts unauthorized individuals from accessing the sender's information or data, as the message has been encrypted.
3. The application ensures the confidentiality of messages or information, as well as the files stored on the computer.

REFERENCES

Research Publications (IJSRP), *8*(7), 495–516. https://doi.org/10.29322/ijsrp.8.7.2018.p7978

Aldossary, S., & Allen, W. (2016). Data Security, Privacy, Availability and Integrity in Cloud Computing: Issues and Current Solutions. *International Journal of Advanced Computer Science and Applications*, *7*(4). https://doi.org/10.14569/ijacsa.2016.070464

Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2020). Symmetric encryption algorithms: Review and evaluation study. *International Journal of Communication Networks and Information Security*, *12*(2), 256–272.

Allwine, A., & Sitorus, J. H. P. . (2019). KEAMANAN DATA DENGAN SKEMA MULTI-KEY HIERARCHICAL IDENTITY-BASED SIGNATURE. *Jurnal Bisantara Informatika*, *3*(1), 17. https://doi.org/10.21131/jbi.v3i1.7

Bucerzan, D., & Cr, M. (2010). *1 Introduction Design of a Stream Cipher*. *V*(4), 483–489.

Bujari, D., & Aribas, E. (2017). Comparative Analysis Of Block Cipher Modes Of Operation. *International Advanced Researches & Engineering Congress*, *November 2017*, 2–5. https://www.researchgate.net/publication/322294203_Comparative_Analysis_of_Block_Cipher_Modes_of_Operation

Ekert, Artur K; Huttner, Bruno; Palma, M. P. A. (1994). Eavesdropping-on-quantum-cryptographical-systems.pdf. *Physical Review A*, *50 (2)*, 1047–1056. https://doi.org/https://doi.org/10.1103/physreva

El Adib, S., & Raissouni, N. (2012). AES Encryption Algorithm Hardware Implementation: Throughput and Area Comparison of 128, 192 and 256-bits Key. *International Journal of Reconfigurable and Embedded Systems (IJRES)*, *1*(2). https://doi.org/10.11591/ijres.v1i2.551

Fouque, P., Jean, J., Peyrin, T., Fouque, P., Jean, J., Peyrin, T., & Evaluation, S. (2014). *Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128 To cite this version : HAL Id : hal-01094302 Structural Evaluation of AES and Chosen-Key Distinguisher of*.

Hutabarat, A., & Sawitri, R. (2024). Text Data Embedding into Images Using Chaotic Least Significant Bit Encod-ing Steganography. *Jurnal Pepadun*, *5*(3), 286–298. https://doi.org/10.23960/pepadun.v5i3.246

Kawle, P., Hiwase, A., Bagde, G., Tekam, E., & Kalbande, R. (2014). Modified Advanced Encryption Standard. *International Journal of Soft Computing and Engineering (IJSCE)*, *4*(1), 21–23.

Parnas, D. L. (1969). On the use of transition diagrams in the design of a user interface for an interactive computer system. *Proceedings of the 1969 24th National Conference, ACM 1969*, 379–385. https://doi.org/10.1145/800195.805945

Rabah, K. (2005). Theory and Implementation of Data Encryption Standard: A Review. *Information Technology Journal*, *4*(4), 307–325. https://doi.org/10.3923/itj.2005.307.325

Sarkar, P., & Noel, M. S. (2020). Cipher: Encryption & Decryption. *International Research Journal of Engineering and Technology*, 731–737. www.irjet.net

Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of IoT and Cloud Computing. *Future Generation Computer Systems*, *78*(December), 964–975. https://doi.org/10.1016/j.future.2016.11.031

Vasanth, S., & Dhikhi, T. (2016). Secure data transmission using steganography and encryption techniques. *International Journal of Pharmacy and Technology*, *8*(4), 21130–21139. https://doi.org/10.5121/ijcis.2012.2314

Wang, G., & Wayne, F. (2014). Embedded System Software Design With State Diagrams. *International Journal of Embedded Systems Volume*, *1*, 28–34.