Table of Contents

Introduction

Problem Statement: Design an ARM-based digital IP using the STM32 Micro-controller to encrypt and compress the IMU data. The ARM-based IP will be used by the SHARC buoy. Requirements are given below. These can be used to derive the exact specifications needed for the design.

The document outlines the design of the system that is going to solve the above problem statement. It contains the analysis of the requirements of the system, the breaking down of the system into subsystems and the tests used to validate the specifications of the system. The paper design also justifies how the proposed system will solve the problem statement.

Requirement Analysis

Interpretation of the requirements

The user requirements were derived from the design project scope as well as the thesis on the SHARC BUOY's. Each of these requirements will be expanded into functional requirements and hence design specifications in the sections to follow . They are listed in no particular order.

User Requirement ID	Description
UR001	User requires the lowest 25% of the Fourier Coefficients as usable.
UR002	The data must be encrypted.
UR003	The data must be compressed.
UR004	Minimize the power usage and maximize battery life longevity.
UR005	Data must be transferred as soon as processing is done.

Table 1: Interpretation of the User Requirements and their Description

Comparison of some available compression and encryption algorithms

Comparison of Compression Algorithms

Compression Algorithm	Compression Ratio	Compression Speed (Mbps)	Decompression Speed (Mbps)
gzip	2.10	444.69	2165.93
zstd	3.14	136.18	536.36
xz	4.31	2.37	62.97
lz4	2.10	444.69	2165.93

Table 2: Comparison of Compression Algorithms

As indicated in **Table 2** , there is a compromise between the compression speed and compression ratio. lz4 with the fastest speed has a lower compression ratio and xz with the largest compression ratio has the slowest speed.Zstd however has a good balance between speed and compression ratio, therefore, making it a more favorable algorithm.

Comparison of Encryption Algorithms

Encryption Algorithm	Processing Speed (MB/s)	Time required to crack all possible keys (50 billion keys per seconds)	Security Level
AES-128	61.010	5×10^{21} days	Extremely High
BlowFish	64.386	3200 days	Moderate
3DES	20.783	800 days	Not secure enough

Table 3 : Comparison of Encryption Algorithms

As indicated in **Table 3**, there is a compromise between the encryption speed and the level of security with AES-128 having the highest level of security needing a total of 5×10^{21} days to crack all possible security keys, however , with a second best encryption speed performance. With 3DES performing the worst, this leaves the BlowFish encryption offering the best encryption speed with a moderate security level(3200 days needed to crack all possible security keys). Since the project requirement did not specify the level of security needed, Blowfish would be the suitable choice as it has the fastest encryption performance, minimizing the number of computations done by the processor.

Feasibility analysis

Economic Feasibility

The project does not have a budget as most of the hardware requirements (STM32 Microcontroller, PC Laptop) and software requirements (MatLab,C) are available for free.

Operational Feasibility

The proposed system appears to solve the problems in the problem statement of the project. This system will allow for the easy, fast and cheap transmission of data from the IMU. Compression of data will make it faster and easier to transmit data from the onboard IMU to the user. Encrypting the data will ensure the security of the transmitted data.

Schedule Feasibility

Given the manpower and complexity of the project, that is what needs to be done as indicated in the development timeline , the project can be done to completion within the given time of the system.

Possible bottlenecks

There are multiple trade-offs that can be noticed from the comparison of algorithms shown above. All the trade-offs involve time in one way or another . The various trade-offs we expect to encounter are listed below:

- Compression level vs compression time
    - The more compressed the file (the lower the compression ratio), the longer it takes to run the corresponding compression algorithm .

- Choice of encryption algorithm vs data needing to be encrypted
    - Different encryption algorithms have different suitabilities. Some algorithms are better suited to finite length data, while some can handle different length data streams. This choice can affect the time it takes to perform the encryption.

- Processing time vs power used
    - The data needs to be pre-processed to pull out only the needed data, but this processing takes time and the more accurately it is done the longer it takes. There is a trade - off between time to process and depth of processing.

Subsystem Design

There are five main subsystems that have been identified and are relevant to the main STM32 micro-processor. Each of these systems are listed below along with their respective functional requirements and design specifications. The systems have been listed in the order in which the submodules interface with the external system as well as other subsystems. After the list of subsystem requirements,inter-subsystem interactions and UML diagram will provide an enhanced visual understanding of how each system and subsystem interface with each other.

Subsystem , Sub-subsystems Requirements and Specifications

Extraction & Storing of Data

This system primarily focuses on the extraction of the data generated by the IMU (a motion tracking device) and storing of it on the STM32 Microcontroller. We are constrained by the memory on the micro controller for the simulations that we will perform.

Functional Requirement ID	Description	User Requirement Addressed
FR001	Micro-controller must communicate with the IMU and extract relevant data	UR005
FR002	Micro-controller to have a storage location to collect raw data and modified subsystem data versions.	UR005

Table 4

Design Specification ID	Description	Functional Requirement Addressed
DS001	The IMU supports I2C communication and these comm ports will be enabled on the STM32. There exists a purchasable sd card module for the stm32 using the SPi comm channel and could be a solution to storage.	FR001 & FR002

Table 5

Data Filtering

A primary user requirement is that the lowest 25% Fourier Coefficients is filtered through. This is a fundamental subsystem as it extracts all relevant data from the IMU data output stored in STM32 microcontroller memory. Furthermore, this reduction in data allows for more efficient compression, encryption and transmission processes. The data IMU output is measured in the time domain and as such a Fourier transform needs to be performed on the dataset to convert it to the frequency domain. With the correct sampling time, this will allow the fast Fourier transform (fft) algorithm to be implemented along with the filtering process of the data's coefficient equivalent. Research has shown that the most appropriate choice in a programming language is the C language. It tops the list of all other languages when it comes to energy efficiency.

Functional Requirement ID	Description	User Requirement Addressed
FR003	Filter the data to remove potential aliasing prior to extracting the 25% of coefficients	UR001
FR004	C programming language is used	UR007

Table 6

Design Specification ID	Description	Functional Requirement Addressed
DS002	The code equivalent of a low pass filter will be supplied to the data output of the IMU to extract at least 25% of the Fourier coefficients of the data	FR003 and FR004

Table 7

Compression of Data

The compression of data is done so by using one of the many compression algorithms that are available on the internet.There have been companions made between the various algorithms in which the compression ratio and speed is used to determine the most efficient algorithm for the SHARC BUOY application.The compression of data precedes the encryption of the data for numerous reasons . The most prominent answer is that once data has been encrypted the file generates a random stream of data , which becomes near impossible to compress. Additionally , compression depends on finding a compressible pattern to reduce the over sized data.

The SHARC BUOY project requires the collected data, from the buoy, to be compressed and transmitted through the existing Antarctic transmission infrastructure. This transmission procedure is limited by the satellite network it uses. This network has high data transfer costs, inconsistent transmission reliability, bandwidth and data structure specifications which all therefore limit the aspect of transmission. Therefore, compression of the data will make it cheaper to transfer, as there is less to transfer, and will increase the likelihood of a complete transfer as the transfer can take place in a shorter period.

Functional Requirement ID	Description	User Requirement Addressed
FR005	Both data sent to the microcontroller and received from the micro controller must be compressed using an appropriate algorithm	UR003
FR006	The subsystem shall do minimum computations to save power.	UR004

Table 8

Design Specification ID	Description	Functional Requirement Addressed
DS003	We shall use the Zstd algorithm to compress our data from the IMU.	FR005
DS004	The adaptive mode of Zstandard is supposed to be set to the minimum required.	FR006
DS005	The microcontroller will communicate with IMU using the I2C interface in order to receive the data	FR007

Table 9

Encryption of Data

Encryption is a method of securing information in which only authorized individuals have access to its contents. The method of encryption involves converting the original data into an alternative, unreadable form, and it is only those who are authorized that can decipher the encrypted form using a security key to gain its contents

There exist many encryption algorithms with different levels of security and speed of execution, however having a high level of security and fast speed of execution are mutually exclusive. In the case of the SHARC BUOY Project,, the level of security is unspecified however a firm requirement is to reduce the amount of processing done within the processor, which means the project requirement is more in favor of the speed of encryption than a high level of security.

Functional Requirement ID	Description	User Requirement Addressed
FR007	The IMU's compressed contents to be sent to the encryption subsystem to generate an encrypted file version ready for transmission	UR002
FR008	The subsystem to limit the amount of processing done in the processor and to minimize the number of computations	UR004

Table 10

Design Specification ID	Description	Functional Requirement Addressed
DS006	A possible choice of encryption could be the use of the algorithm BlowFish which has a high level of speed and a moderate level of security.	FR007
DS007	BlowFish speed of execution is one of the fastest algorithms available today thus limiting the number of processes happening in the processor	FR008

Table 11

Inter-Subsystem and Inter-Sub-subsystems Interactions

- The compression subsystem gets raw data from the IMU and compresses it. The encryption subsystem gets the data from the compression subsystem and encrypts it before the data is transmitted.

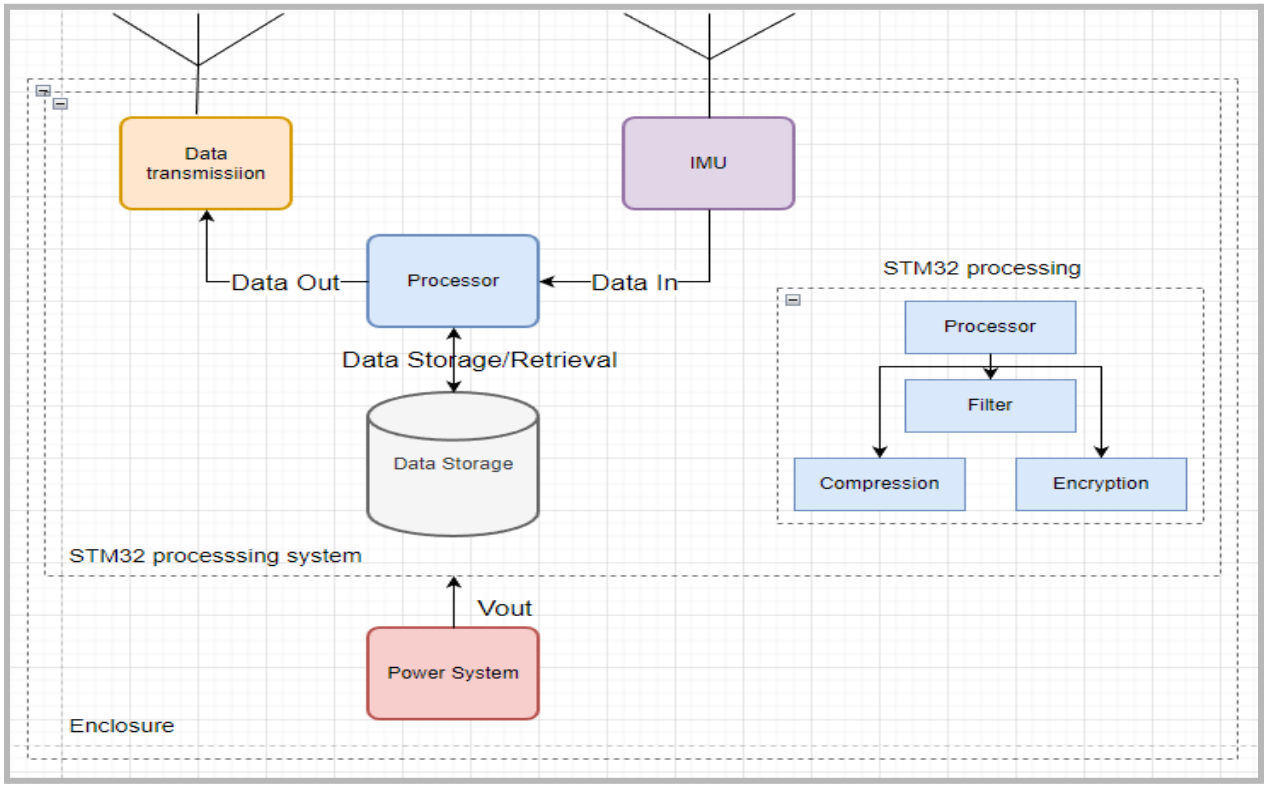

Figure 1: UML Diagram showing the Different Subsystems

UML Diagram

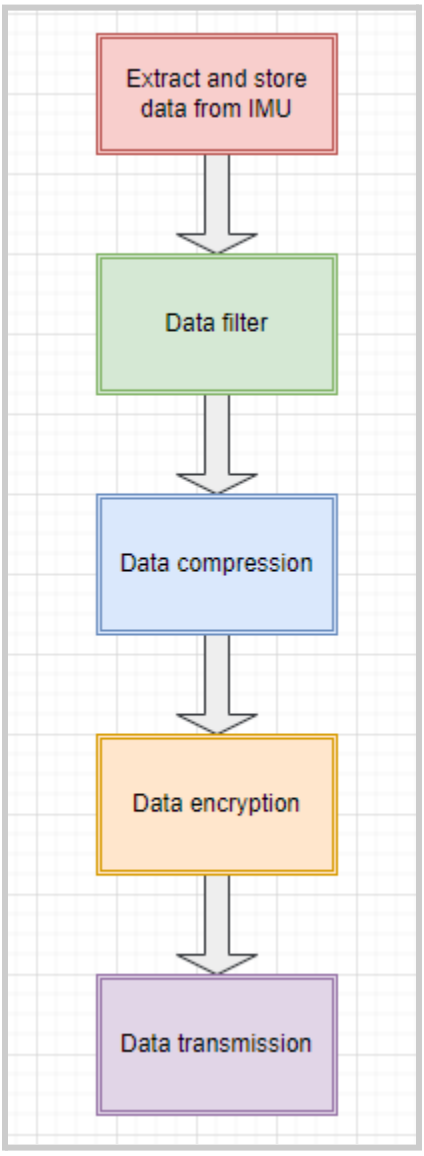

Figure 2: UML Diagram showing the Different Subsystems

Acceptance Test Procedure

Figures of merits, which we will use to validate our final design

Specification(s) ID	Specification(s)	Figure of Merits
DS003	We shall use Zstd algorithm to compress our data from the IMU	The algorithm is supposed to compress the data from the simulations so that we will be able to retrieve at least 25% of the data
DS004	The adaptive mode of Zstandard is supposed to be set to the minimum required	Fastest time of execution
DS005	The microcontroller will communicate with the IMU in order to receive the data	Print out data from the IMU to prove retrieval
DS006	The use of the algorithm BlowFish to encrypt the compressed IMU data	Sending an unencrypted file to the microcontroller and retrieving it, seeing if the input data matches the yield data. The data should not match
DS007	Choosing an encryption algorithm that limits the number of processors done on the microcontroller.	Time how long it takes for the file to be encrypted and returned from the microcontroller, generating an average speed.

Table 10

Experiment design to test these figures of merit

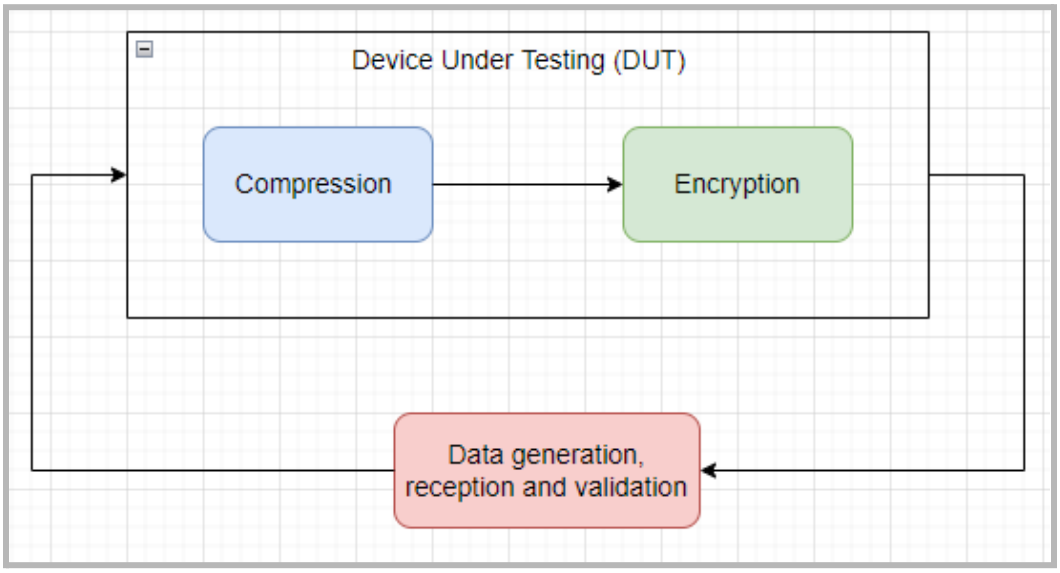

Figure 3: Experiment set up for Loop-In Testing

Set up the apparatus, as shown above , If we are testing for Compression only the DUT will only test the compression block and likewise for Encryption. Analysis can be done on the output results to see if they meet the figures of merit we suggested.

Acceptable performance definition

For the compression test with the input being Fast Fourier Transform (fft) data , an acceptable performance is for us to be able to extract at least 25% compressed Fourier transformation coefficients from the compressed data, Also the compression speed must be less than or equal to 136.18 Mbps.

For the encryption test with an input of sample csv IMU data, an acceptable performance would be to determine if the contents of the csv file are scrambled after being processed by the encryption submodule. We also need to determine if the encryption is reversible by decrypting the file to determine if the original contents are present. Lastly, the speed of encryption should roughly achieve a rate of 64MB/s..

Development timeline

The project is planned to be completed within the course timeline. The paper design, progress report 1, progress report 2 , final report and presentation of the demo video must all be completed within the allotted time frame.

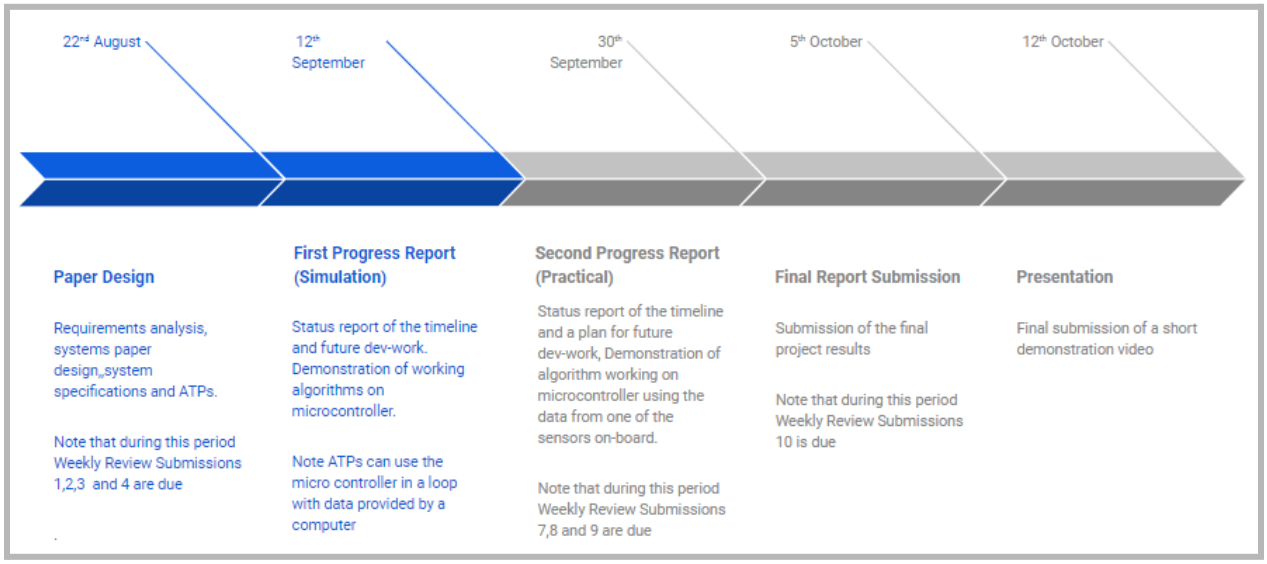

Figure 3: Our Project Development Timeline

Project Management

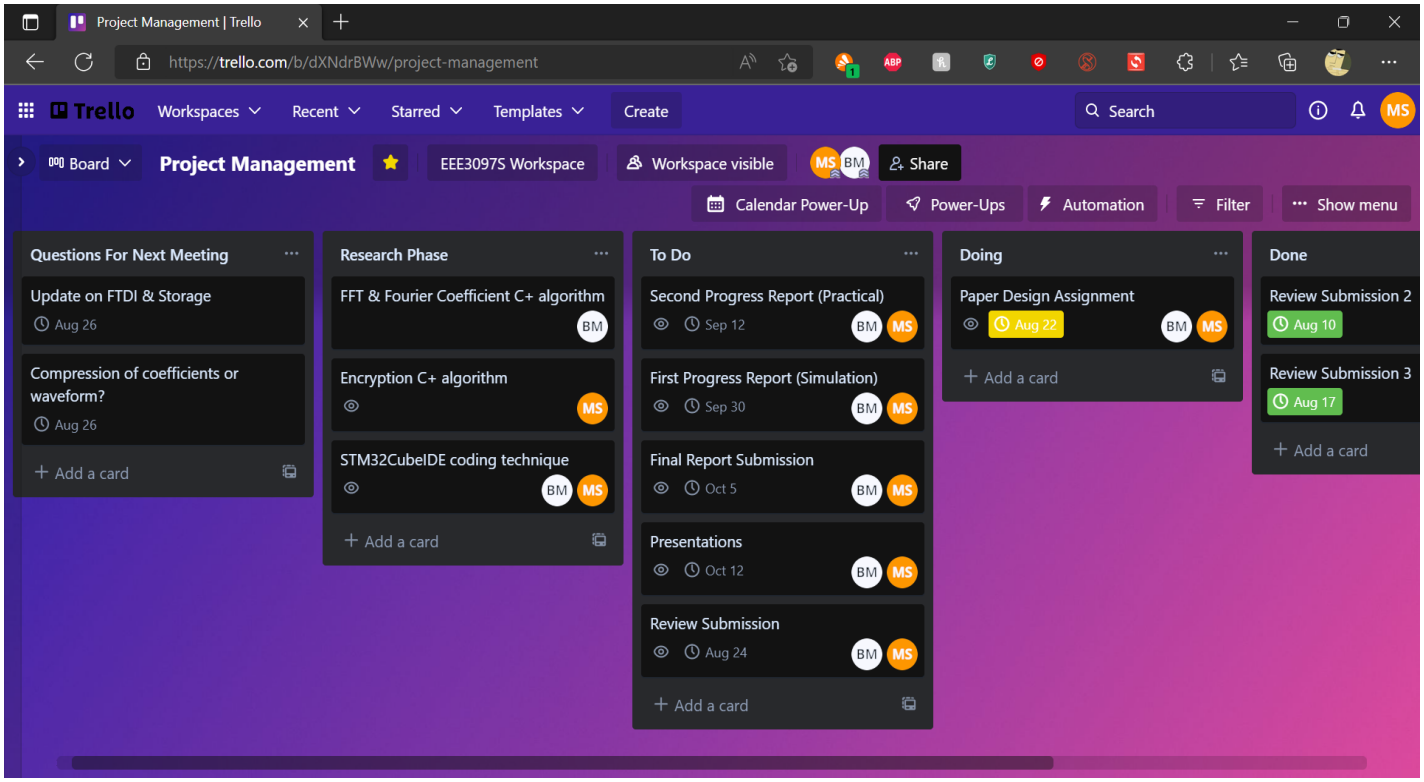

Figure 4: Our Project Management Tool

- Here is the link to our github repository here

Distribution of Work

Work	Member
Interpretation of the requirements	MTMBRE002
Comparison of some available compression and encryption	MTMBRE002 - Compression & STLMOU001 - Encryption
Feasibility analysis	MTMBRE002
Possible bottlenecks	STLMOU001
Subsystem and Subsystem Requirements	MTMBRE002 & STLMOU001
Subsystem and Subsystem Specifications	MTMBRE002 & STLMOU001
Inter-Subsystem and Inter Sub-subsystem interaction	STLMOU001
UML or OP Diagrams	MTMBRE002 & STLMOU001
Figure of merits	MTMBRE002 & STLMOU001
Experiment design to test these figures of merit	MTMBRE002 & STLMOU001
Acceptance performance definition	MTMBRE002 & STLMOU001
Development timeline	MTMBRE002

References

1. GitHub. 2021. GitHub - lz4/lz4: Extremely Fast Compression algorithm. [online] Available at: https://github.com/lz4/lz4 [Accessed 03 September 2021].
2. Ameer, I., 2020. The Effect of Re-Use of Lossy JPEG Compression Algorithm on the Quality of Satellite Image. Neuroquantology, 18(5), pp.17-25.
3. Rao, K., 2021. Discrete cosine transform : algorithms, advantages, applications in SearchWorks catalog.
4. [online] Searchworks.stanford.edu. Available at: <https://searchworks.stanford.edu/view/155762> [Accessed 01 September 2021]
5. Al Tamimi, A., 2022. *Performance Analysis of Data Encryption Algorithms*. [online] Cse.wustl.edu. Available at: <https://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/index.html> [Accessed 22 August 2022].
6. Jeevalatha, E. and SenthilMurugan, S., 2018. *Evolution of AES, Blowfish and Two fish Encryption Algorithm*. 9th ed. [ebook] Vaniyambadi, pp.1-4. Available at: <https://www.ijser.org/researchpaper/Evolution-of-AES-Blowfish-and-Two-fish-Encryption-Algorithm.pdf> [Accessed 22 August 2022].
7. Sari, S., 2022. *DES vs 3DES vs Blowfish vs AES*. [online] Baeldung. Available at: <https://www.baeldung.com/cs/des-vs-3des-vs-blowfish-vs-aes> [Accessed 22 August 2022].