# Paper Design

*Oliver Shaw* *Mbasa Mguguma*
*SHWOLI002* *MGGMBA003*

| | |
|---|---|
| Oliver Shaw | Encryption Subsystem |
| Mguguma Mbasa | Compression Subsystem |

# 1. Introduction

The project is to design an ARM based digital IP using the STM32 to encrypt and compress IMU data that is from SCALE's flexible buoy system operating over the pancake ice of the Antarctic.

# 2 Requirement Analysis

| | |
|---|---|
| U1 | Extract 25% of Fourier Coefficients of the data |
| U2 | Minimize computation required |
| U3 | Encrypt and compress IMU data |

### 2.1 Encryption Subsystem

*Symmetric Encryption*

A symmetric encryption algorithm uses just a single cryptography key for encryption and decryption. Using symmetric encryption algorithm are significantly faster than asymmetric encryption algorithms. They also require less computational power than asymmetrical encryption algorithms. This means that symmetric encryption techniques are very useful for large data sets.

*Advanced Encryption System (AES)*

AES uses a family of block cypher that consists of different key lengths and block sizes. This encryption technique uses substitution and permutation. During the encryption process various sub-processes are used. This includes sub bytes, row shifts, column mixes and adding round keys. This process has many rounds depending on the size of the key.

Compared to other symmetric algorithms, such as DES, AES is a faster and safer algorithm. The option to use multiple key lengths are another big advantage to using AES.

*Asymmetric Encryption*

An asymmetric encryption algorithm uses multiple keys for encryption and decryption. There is a distinct key for the encryption and decryption stages. This type of encryption is more secure than symmetrical encryption. The use of a private decryption key means that a system is secure against man-in-the-middle attacks. This type of encryption also has the benefit of offering authentication. This is due to the private key that ensures the only entity able to see and decrypt the data is the entity its intended for.

*RSA Encryption Algorithm*

This system of encryption uses prime factorisation in which two large prime numbers are multiplied together. Figuring out the original prime numbers is the encryption puzzle. The full length algorithm is virtually impossible to crack.

RSA Encryption is extremely scalable and the level of protection is based directly on the key length.

*Feasibility Analysis*

One necessity for the project is to reduce the power consumption of the unit. The IMU gives information about the ice as well as the wave dynamics and this data needs as many data points as possible. Due to this system being a standalone (in terms of having limited power in the system. Therefore the smartest design decision would be to use symmetric encryption algorithm of AES. The other requirement of ensuring that at least 25% of the Fourier coefficients will also be ensured by using AES.

*Possible Bottlenecks*

The process of encryption is only bottlenecked by the size of the microcontroller. The microcontroller is only 32 bit which means the encryption is limited in terms of data transfer as well as key size. This means that the encryption is less secure.

**2.2 Compression Subsystem**

*Compression algorithms*

Since we will be using fixed point numbers, Integer compression algorithms will be efficient for the project.

*Run-length encoding*

This algorithm is lossless and is efficient when there is a lot of repeats in the data that is being compressed [1]. But it does not take into account the number of bytes used to represent that number.

*Delta*

This algorithm is also lossless and it reduces the number of bytes to store a number [1]. storing all the deltas takes up bits.

For the project we will use a combination of delta and Run-length encoding algorithms, to compress the data.

# 3 Subsystem Design

### 3.1 Encryption Subsystem

*Encryption Subsystem Requirements*

| | |
|---|---|
| U1.F2 | The encryption process must not lose any data |
| U2.F3 | The encryption algorithm must be high speed |
| U2.F4 | The encryption algorithm must be low power |

*Encryption Subsystem Specifications*

| | |
|---|---|
| U1.F2.S1 | The algorithm must maintain 100% integrity of the data |
| U2.F3.S1 | Encryption should take 1.5 seconds per Gb of data |
| U2.F3.S1 | Decryption should take 1.5 seconds per Gb of data |

### 3.2 Compression Subsystem

*Compression Subsystem Requirements*

| | |
|---|---|
| U1.F1 | The algorithm must not lose the data |
| U2.F1 | Use high speed algorithm |
| U2.F2 | Use low power |

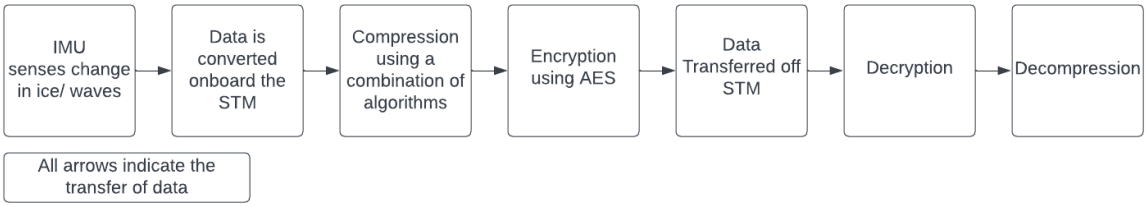
*Compression Subsystem Specifications*

| | |
|---|---|
| U1.F1.S1 | 25% of Fourier Coeffients of the data must be retained |
| U2.F1.S1 | Compression and decompression must take not more than 2 seconds |
| U2.F1.S1 | Compression ratio of the algorithm should be less than 0.25 |

### 3.3 Subsystem Interaction

*Interaction protocol*

For the sending of the data from the IMU to the STM we will use SPI protocol since it is faster than I2C [2]. We are prioritizing speed to minimize power usage in order to stay within the brief as well as be able to have the system running for as long as possible on a limited power supply.




# 4 Acceptance Test Procedure

### 4.1 Encryption Subsystem

| | |
|---|---|
| U1.F2.S1.A1 | Take a data set and compare it at encryption vs decryption to ensure 100% integrity in the data |
| U2.F3.S1 | Set a timer system up for the encryption process |
| U2.F3.S1 | Set a timer system up for the decryption process |

The timing testing procedures will be very straight forward as they rely simply on code based timings that can be easily implemented into the code in order to ensure they are running efficiently and quickly. This will allow us to check for bottlenecks slowing down the encryption or decryption process.

A defined data set must be input into the encryption system in order to test the integrity of the code. This way we will be able to compare the output of the decryption and compare it directly to the defined data set.

The figures of merit for the encryption subsystem are that the encryption and decryption each take 1.5 seconds (maximum) to execute. As well as 100% integrity of the data after decryption.

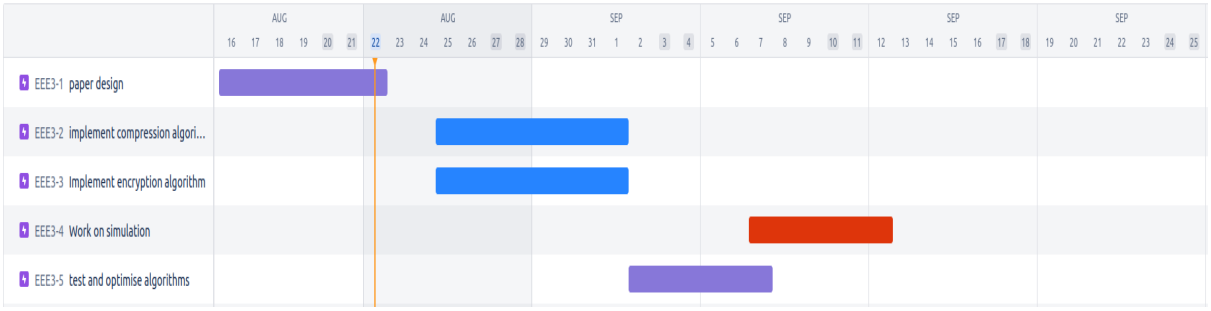
### 4.2 Encryption Subsystem

To test the compression subsystem, we will compress the data from the IMU in the stm and send it to a laptop then decompress it. From the data and the process, we will use the following to test if we meet the specifications:

| | |
|---|---|
| U1.F1.S1.A1 | Check if 25% of Fourier Coeffients is in the data decompressed. |
| U2.F1.S1 | Time the process to check if it is under 2 seconds |

| | |
|---|---|
| U2.F1.S1 | Calculate compression ratio and check if it is less than 0.25 |

# 4 Timing Diagram




# 5 References

[1] https://www.timescale.com/blog/time-series-compression-algorithms-explained/
[2] https://prodigytechno.com/i2c-vs-spi/
[3] https://www.thesslstore.com/blog/types-of-encryption-encryption-algorithms-how-to-choose-the-right-one/