EEE3097S PROJECT : PAPER DESIGN

1st VUYISA MPETSHENI
STUDENT NUMBER: KZWVUY001
DEPARTMENT: ECE
INSTITUTION: UCT

2nd S'THABISO LUSHABA
STUDENT NUMBER: LSHSTH002
DEPARTMENT: ECE
INSTITUTION: UCT

EEE3097S ENGINEERING DESIGN
COURSE LECTURE: PROF AMIT MISHRA
TUTOR: TRAVIS BARRET
DUE DATE: 22 AUGUST 2022

CONTENTS

TABLE I
WORK DIVISION

Task	Student assigned to the task
Introduction	Vuyisa Mpetsheni
Interpretation of the requirements	Both
Comparison of compression algorithms	Vuyisa Mpetsheni
Comparison of Encryption algorithms	S'thabiso Lushaba
Feasibility analysis	Vuyisa Mpetsheni
Possible bottlenecks	S'thabiso Lushaba
Encryption requirements	S'thabiso Lushaba
Encryption specifications	S'thabiso Lushaba
Compression requirements	Vuyisa Mpetsheni
Compression specifications	Vuyisa Mpetsheni
Subsystem Design	Both
Integration of subsystems	S'thabiso Lushaba
UML diagram	Both
Acceptance test procedure	Both
Development timeline	Both

I. Introduction

This report discusses the requirements and sub-systems of an ARM-based digital IP to be designed using the Raspberry Pi to encrypt and compress IMU sensor data. The department of electrical engineering is working on a flexible buoy that is to be installed on pancake ice to collect environmental data. They have asked our team to design a cheap but efficient ARM-based digital IP that will be used in transmitting the information from the IMU sensor. The objectives of this report are therefore to: analyze the requirements, develop subsystems, define requirements for each subsystem, derive specifications from the requirements, and also come up with an acceptance test procedure, and finally conclude with the development timeline for the project.

II. Interpretation of requirements

A. Technical requirements and derived specifications

1) requirements:

- The system shall collect environmental data using an IMU sensor.
- The Design must be ARM-based digital IP that compresses and encrypts data.

2) specifications:

- STM32F0dicovery board shall be used to communicate with the sensor
- IMU ICM20948 sensor shall be used to collect environmental data

B. Compression requirements and derived specifications

1) requirements:

- At least 25 percent Fourier coefficient of the data should be extracted.
- The data from the IMU sensor should be compressed without consuming too much power

2) specifications:

- Use a lossless Algorithm to compress the data.
- Use Zstd algorithm

C. Encryption requirements and derived specifications

1) requirements:

- The output data should be identical to the input data
- The Encryption/Decryption Method should use one key or passphrase.
- The Algorithm used should use as little power as possible
- The algorithm must not be heavy on the Processing unit.

2) specifications:

- Implement a symmetrical encryption algorithm to as they use one key.
- Implement an algorithm that uses lesser power, out of the symmetrical algorithms.
- symmetrical algorithm to minimize complexity for decoding, hence less strain on the processing unit.

III. Comparison of compression and encryption algorithms

1) Compression: For a better understanding of what characteristics are sought in the compression algorithm, let's first define the word compression. compression of data is the process of encoding data using fewer bits than the original data. Compression algorithms can be lossless (no data is lost) or lossy (some data might be lost). In this report, compression algorithms will be compared based on compression speed, decompression speed, and their compression ratios. A wide range of compression algorithms are available, but the report has narrowed them down to the most effective ones.

- **Delta-encoding** stores only the difference between a data object and one or more reference objects, thus reducing the amount of data required to represent that object. Repetitive information is handled well by the algorithm. it compresses integer data types.
- **Delta-of-delta encoding** This algorithm applies delta-encoding to already delta-encoded data. It also works with integers.
- **Simple-b8** a delta-of-delta algorithm reduces the number of integers stored but Simple-b8 finds a way to store them efficiently by storing a set of integers as a series of fixed-sized blocks. Each block's first bit indicates the minimum bit length.
- **Run-length encoding** RLE works best when there is a large number of repeats of the same value, this usually happens when a value does not change. Sequences are encoded to store only one value and its count.
- **XOR-based compression** XOR-based compression works better with floating point numbers than delta-encoding. This algorithm XORs successive floating point numbers together, storing only the different bits.
- **Dictionary compression** Stores an index into a dictionary containing the unique values after making a list of the possible values.

	Ratio	Comp Speed MB/s	Decomp Speed MB/s
lz4	2.10	444.69	2165.93
zstd	3.14	136.18	536.36
zlib	3.11	23.21	281.52
xz	4.31	2.37	62.97

Fig. 1. Dictionary Compression Algorithms

- In this report LZ4 and Zstd algorithms shall be used since they have balanced compression ratio, compression speed, and decompression speed

2) Encryption: Encryption can be defined as the process of bringing means of security and privacy to a data set. This is achieved by converting readable data into an encoded format. The then encrypted data can be converted back to readable data

with a key/passphrase which only the designated receiver of the data has. There two main types of encryption are discussed below:

- **Symmetric Encryption** There is only one key used to encrypt and decry-pt the data. Any user or party with a matching key can decry-pt the data to readable data.
  - **Advantages:** Symmetric encryption is relatively simpler to implement. Less power consumption during computation and has faster performance. has keys of length size 128, 192, or 256 bits.
  - **Disadvantages:** The symmetric encryption is less secure as anyone with the matching key can convert the encoded data back to readable data.
- **Asymmetric Encryption** Encryption and decryption using matching key pairs. the matching keys are known as the private and public key. Every public key matches to only one private key.
  - **Advantages:** The readable data is only seen by the designated user, as opposed to being seen by anyone with a key, so it is more secure than symmetric encryption.
  - **Disadvantages:** Risky, If the private key is lost the data can never be decrypted. Relatively slow. Risk of a security breach if the key is discovered by someone else through other means.
- **Symmetric Encryption Algorithms:**

  1. Advanced Encryption Standard(AES), The algorithm takes in input as 8-bit arrays that create data blocks. The algorithm can operate in varying data blocks and makes use of key with length sizes 128, 192 or 256 bits depending on the data block.

  2. Data Encryption Standard(DES). The algorithm takes in plain text input and breaks it into chunks of data of size 64-bits and encrypts it using a cryptographic key. The cryptographic key used is of specific short length 56-bits.

  3. TwoFish Encryption. One of the fastest symmetric algorithms and makes use of one 256-bit key.The TwoFish encryption stands out from most symmetric algorithms as its implementation makes it more ideal for hardware and software environments.
- **Asymmetric Encryption Algorithms:**
  1. RSA algorithm(invented by Rivest, Shamir and Adleman) uses keys of length size ranging from 512 to 2048-bits. The initial procedure involves multiplying to large prime numbers and the integers used by this method are sufficiently large making it difficult to solve.

  2. Diffie–Hellman key exchange. This method allows two users having no prior knowledge of each other to have a shared key established between them over public channel. The established key is not being transmitted over the internet which makes increases the security it provides.

  3. Elliptic curve cryptography(ECC). This method is based on the structure of elliptic curves over infinite fields, It allows smaller keys compared to non-EC cryptography to provide equivalent security.

- Having discussed the encryption-decryption algorithms, with the project requirements in consideration, out of the three symmetric algorithms TowFish, EAS and DES the TwoFish algorithm is chosen for the following reasons:
  - The TwoFish algorithm is the fastest encryption algorithm, meaning a reduced processing time which favours the project requirement.
  - With faster performance comes reduced computational processing therefore reduced power consumption during data processing.
  - The TwoFish algorithm also has one key of length 256-bits which will be easier to deal with than multiple keys.

IV. Feasibility analysis

- Due to the availability of available hardware as well as compression and encryption algorithms, the project should be relatively simple to complete without costing a fortune.
- Using this system appears to reduce the cost of data transmission and also encrypts data very well.
- There is enough time to complete the project, and the group members have divided it so that it can be completed a week before the final submission.

V. Possible bottlenecks

- Our processor differs from the one in the Buoy project in a few ways, one of them being that the Buoy's processor is a floating point, whereas ours is a fixed point.
- This project will use the ICM20948, a 9-axis motion tracking device, whereas the buoy uses the ICM20689, a 6-axis motion tracking device. As a result, some data from the ICM20689 sensor will need to be discarded.
- As the project is aimed at minimizing the processor computation while maximizing the storage saved, this might be a problem to overcome without compromising data accuracy.
- There data that will be used will be slightly different from the data that is usually collected from the southern ocean due to the difference in environmental conditions.

VI. UML Diagram

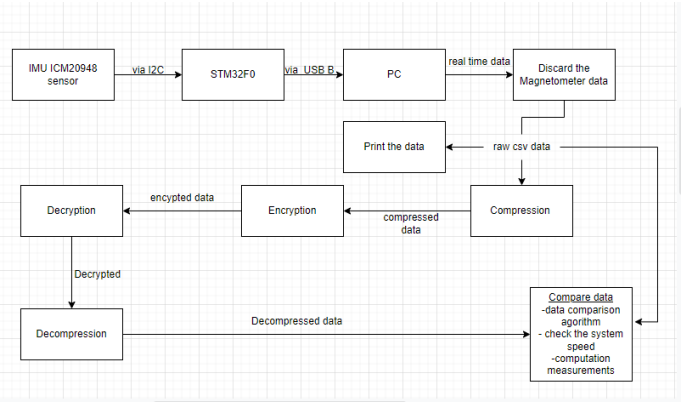

Fig. 2. Subsystem Interaction UML diagram

VII. Subsytem Design

A. Subsystems and Sub-subsystems Requirements

1) Hardware Subsystem (STM32F0dicovery):

- Must sufficiently perform Encryption.
- Must sufficiently perform Compression.
- must connect to the IMU: ICM-20649

2) Software Subsystem:

- **Encryption : Subsystem Encoding**
  - Must Encode all the data set/No data leaks
  - Must Not be a complex algorithm.
- **Encryption : Subsystem Decoding**
  - The Data decoded must be identical to the encoded data.
  - Must Not be a complex algorithm.
  - Algorithm must be fast and efficient.
- **Encryption : Subsystem Key/Passphrase Generation**
  - Only a one key/single key must be produced.

B. Subsystems and Sub-subsystems Specifications

1) Hardware Subsystem (STM32F0dicovery):

- 3v3 Power Supply
- ARM Cortex M0 Processor

2) Software Subsystem:

- **Encryption : Subsystem Encoding**
  - TwoFish Encryption Algorithm
- **Encryption : Subsystem Decoding**
  - TwoFish Encryption Algorithm
- **Encryption : Subsystem Key/Passphrase Generation**
  - 256-bit key length size.

C. Subsystems Interactions

- The Data is retrieved from the IMU ICM20948 sensor via I2C com line to the STM32F0).
- The STM is connected to a PC via USB type B debugger.
- The PC performs the necessary data analysis to cut out the Magnetometer data.
- The then processed data is passed to the compression algorithm
- After the data has been efficiently compressed it is passed on to the next step for encryption.
- The data is encoded in the encryption subsystem and a key is generated
- The data is then passed on to decryption , the data is decoded given the right key is used.
- The decrypted data is then Decompressed.
- The Decompressed data is passed onto a data comparison algorithm against the original data to check if the process was successful.

VIII. Acceptance Test procedure

A. Figure of merit

- Data from the sensor can be printed to prove communication.
- At least 25 percent Fourier coefficient of the data can be extracted.
- The system can minimize the consumption of power.
- The data before the system is the same as the data after the system.
- the data takes up less storage after compression.
- the data should be encoded to unreadable format after encryption.

B. Experiment Design to Test Figures of Merit

- Connect the IMU sensor to the STM32F0 and PC, print out data from the IMU sensor to prove communication and if the sensor and STM32F0 work.
- Run the system codes(compression, encryption, decryption, and decompression) and record the execution time.
- Run the compression algorithms and record the execution time. Check if the compressed data takes up less space than the original.
- Run the encryption algorithms, record the execution time, and print out the encrypted data to check if the data is encoded into unreadable data.
- Run a code to compare the raw data and the data after going through the system to see if that at least 25 percent Fourier coefficient of the data is extracted.
- Run a code to compare raw data and data after going through the system to check if the data is the same.

C. Acceptable Performance Definition

- Printing the data from the sensor on the PC, if the data printed out is a 9-axis motion tracking data then the performance is acceptable otherwise if there is no data printed out, the performance is unacceptable

- After running the compression algorithm if there compressed data takes less storage than the original, then the performance is acceptable but if the storage taken by compressed data is more or the same as the original data then the performance is unacceptable
- After running an encryption algorithm if the data is encoded into unreadable data then the performance is acceptable but if the data is readable, the performance is unacceptable.
- after running the whole system, if At least 25 percent Fourier coefficient of the data is extracted, then performance is acceptable but if less than 25 percent Fourier coefficient of the data is not extracted, then the performance is unacceptable.
- after running a compare function if the original data is the same as the data after running the whole system, then the performance is acceptable but if the data are not the same, the performance is unacceptable.

IX. Project Tracking

A. Development Timeline Diagrams

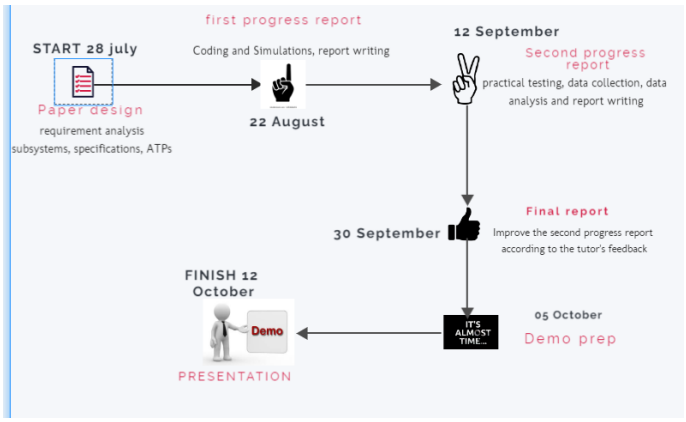

Fig. 3. A diagram showing development of the project

-

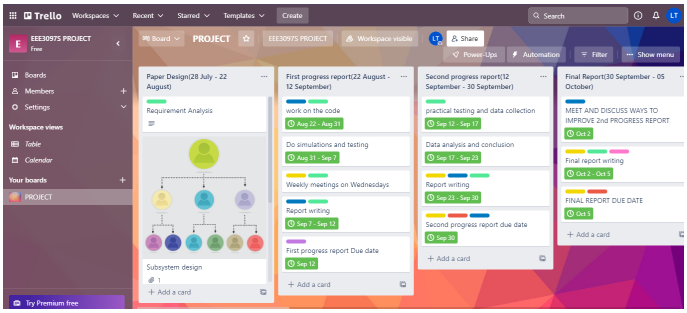

Fig. 4. Trello Project Management page

-

X. References

[1]PreVail.(2021,January,12).Public private key pairs and how they work[online].Available:https://www.preveil.com/blog/public-and-private-key/.[Accessed: 17 Aug 2022]

[2]GeeksforGeeks(2022,June,13). RSA Algorithm in Cryptography[Online].Available:https://www.geeksforgeeks.org/rsa-algorithm-cryptography.[Accessed: 15 Aug 2022]

[3] J. Thakkar, Types of Encryption: 5 Encryption Algorithms How to Choose the Right One, Hashed Out by The SSL Store™, 2021. [Online]. Available: https://www.thesslstore.com/blog/types-of-encryptionencryption-algorithms-how-to-choose-the-right-one/. [Accessed: 22 Aug 2022].

[4]"Time-series compression algorithms, explained", Timescale Blog, 2022. [Online]. Available: https://www.timescale.com/blog/time-series-compression-algorithms-explained/. [Accessed: 22- Aug- 2022]

[5]Y. Collet and C. Turner, "Smaller and faster data compression with Zstandard", Engineering at Meta, 2022. [Online]. Available: https://engineering.fb.com/2016/08/31/core-data/smaller-and-faster-data-compression-with-zstandard/. [Accessed: 22- Aug- 2022]