High Level Design (HLD)

Hybrid Techniques for Data Encryption

Last date of revision: 3 March 2023

Sudhanshu Meshram

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 18/01/2022 | 1.0 | Added Introduction and General description | Sudhanshu Meshram |
| 21/01/2022 | 1.1 | Added Design details and abstract | Sudhanshu Meshram |
| 22/01/2022 | 1.2 | Created and organized whole document | Sudhanshu Meshram |
| 25/01/2022 | 1.3 | Updated Design Details | Sudhanshu Meshram |
| 27/01/2022 | 1.4 | Checked and added Performance | Sudhanshu Meshram |
| 28/01/2022 | 1.5 | Added Conclusion and References | Sudhanshu Meshram |
| 31/01/2022 | 2.0 | Updated diagrams in Design Details | Sudhanshu Meshram |

**Table of Content**

Document Version Control ……………………………………………………………. 2

Abstract ………………………………………………………………………………………… 4

Introduction ………………………………………………………………………………….. 5

1. Why this High-Level Document? ……………………………………………. 5

2. Scope …………………………….………………………………………………………. 5

3. Definition ………………………………………………………………………………. 5

General Description ………………………………………………………………………. 6

1. Product Perspective ……………………………………………………….………. 6

2. Problem Statement ………………………………………………………..………. 6

3. Problem Solution ……………………………………………………………………. 6

4. Proposed Methodology …………………………………………………….……. 6

5. Further Improvements ……………………………………………………………. 6

6. Data Required ……………………………………………………...…………………. 6

7. Tools Used ………………………………………………………………………………. 7

8. Constraints …………………………………………………………………..…………. 7

9. Assumptions ……………………………………………………………………………. 7

Design Details ……………………………………………………………..…………………. 8

1. Process Workflow ………………………………………………………..…………. 8

2. Error Handling ………………………………………………………………..………. 9

Performance …………………………………………………………………………….……. 10

1. Reusability ……………………………………………………………..………………. 10

2. Application compatibility ………………………………………..………………. 10

3. Resources Utilization ………………………………………………………….……. 10

4. Deployment ……………………………………………………………………..…..…. 10

Conclusion ………………………………………………………………………….……..……. 11

Reference ………………………………………………………………………………..………. 12

**Abstract**

Nowadays, Data is in hype, So many of concepts are running around in our virtual environment on the basis of real world aspects. Humans are working on large amount of data, generating it in a huge scale also working on it and predicting for future possibilities. In account to share this data big data, data is passed to several users maybe confidential in order to protect this data files, messages, docs, numbers, etc. We are using Encryption and Decryption data algorithm techniques to protect data from attackers. This lead to protect of data sent and receive in a convenient way successfully.

**Introduction**

**1. Why this High-Level Design Document?**

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the architecture of the project

**2. Scope**

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

**3. Definition**

The terms used in the projects are:

Python libraries :- Cryptography :- It is a package in python used for providing encryption decryption related data

Fernet :- Fernet guarantees that a message encrypted using it cannot be manipulated or read without the key. [Fernet](https://github.com/fernet/spec/) is an implementation of symmetric (also known as “secret key”) authenticated cryptography.

**General Description**

**1. Product Perspective**

The Data Encryption techniques are used to encrypt data with some logical ciphers generated by algorithms in order to protect them from unwanted user only receiver should receive it without any interruption in communication. Encryption of data contains the Key for particular Encryption algorithm. They cannot known by attackers as they are set randomly and only can be known by Sender and receiver.

**2. Problem Statement**

Human world nowadays is working on virtual world with a large amount of resources provided in order to protect them we need security for confidentiality.

**3. Problem Solution**

Developing different encryption techniques for data protection and Encryption, Decryption using information technology based architectural functionalities to operate flow of system nicely without fail.

**4. Further Improvement**

The project can be extended by using Blockchain based concept in future if studied deeply as it provide good security that cannot be broken . Also current encryption technology can be improved with more ideas like different ciphers or key changing at both ends. I know you are expecting some AI related stuff but I m know AI is far more dangerous than any of our models.

**5. Data Required**

For training the model we need the data of attacks done and most common generated ciphers and techniques commonly used attackers to decrypt the data, the way they thinks which things they target first.

**6. Tools Used**

* Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn are used to build the whole model.
* PyCharm and Visual Studio Code is used as IDE.
* For visualization of the plots, Matplotlib and Seaborn are used.
* Front end development is done using Stream lit.
* GitHub is used as version control system.

**7. Constraints**

The model should protect users data without letting it known by the user no problems should be faced by user while using the platform.

**8. Assumptions**

We are assuming that the project should use different cipher keys for each message encryption without fail and check for all possibilities of data loss.

All the code the web page or application should be ensuring that the data which is been Encrypted should be decrypted by the Receiver only at any cost data confidentiality should be maintained.

**Design Details**

1. **Process Workflow**

**Cryptography**

key

key

Encryption

Algorithm

Decryption Algorithm

Plain text at Receiver

end

Plain Text by Sender

**Encryption Process**

**Deployment Process**

**Data Box**

**Front End page**

**Start**

**2. Error Handling**

Initially we got error when connecting Heroku with GitHub which we solved. We then got error displaying the graphs and multiple text were not solving. All of the above errors have been resolved.

**Performance**

**1. Reusability**

The techniques follows the aspects of reusability as it python if I have modify or reuse it for any other cipher text conversion processes then it will be useful for my work of any kind of security based protocol.

**2. Application compatibility**

I’m using python to code this process and python is a compatible language also html then sent to python code page direct access.

**3. Resource utilization**

Python provides a lot of functionalities also code is small so if a user is using it. It will not going to lag , python provides a lot of database related module so it is easy to operate with less power consumption as well.

**Conclusion**

The techniques is to protect the data sent so code will generate the random cipher text on the given plaintext , so the user entering the plaintext and then it will be encrypted and sent to receivers end with the key known by both of them and data will be decrypted at receivers end.

The whole techniques is about mixture of different data protection techniques for the security purpose.

**References**

1. https://www.researchgate.net/publication/363325970\_Application\_Research\_of\_Data\_Encryption\_Technology\_in\_Computer\_Network\_Information\_Security
2. https://scikit-learn.org
3. https://numpy.org
4. https://seaborn.pydata.org/examples/regression
5. https://seaborn.pydata.org
6. https://matplotlib.org/
7. https://pandas.pydata.org/docs