iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35329

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Golang Library/CLI Tool to Provide Data Analysis and Conversion Utilities

Tyagraj Desigar[1], Jyoti Chavan[2], Karthik Shriniketan[3], Prof. Jagruti Wagh[4]
[1, 2, 3]*Department of Computer Engineering, MMCOE, Karve-Nagar, Pune, India*
[4]*Professor, Department of Computer Engineering, MMCOE, Karve-Nagar, Pune, India*

Abstract: ***When developing or testing any complex software that deals with data or networks or low-level system software, or when doing regular administration of any development or production environments, or while learning about security or networks people often need tools to convert, parse, analyse and use data in various ways. These can sometimes be difficult to find, scattered and sometimes much more in-depth than they need. In this paper, we aim to propose a system that will get aggregate such operations into a library that will be accessible using the command line interface. The goal of this project is to create a versatile library/CLI tool that takes a simple text input and performs the user's desired operations on it and produces an output.***

Keywords: ***CLI tool, Golang, library, stack, operations.***

I. INTRODUCTION

GO is a modern, low-level, statically typed, concurrency focused language built for scalability and high performance. It provides vast programming functionalities and is easy to learn and understand because of its syntactic similarities to C. Due to the unavailability of a preloaded CLI tool in many software users resort to online applications for the same. This introduces various challenges including network, data security, and performance. The proposed system aims at solving the issues encountered by the user and provides a convenient way to perform tasks and operations provided by our system.

The proposed system is an in-built CLI tool that will allow the user to perform set operations without the aforementioned problems. The goal achieved is such that the end-user can successfully do tasks as per the library. The library could be used for programmatic usage of this functionality in scripts or other tools. The library would provide a simple interface that the user can use to create a 'stack' of operations to be performed on input or simply use any of the operations standalone. Each operation would have some customizable parameters with some default behaviour and in the end, return the output in the desired format.

II. MOTIVATION

The unavailability of an in-built CLI tool with a versatile library is a problem faced by many people. The alternative methods available are useful and are successful in solving the problems, however, they pose new risks and threats to the work of the user.

With the help of the library, operations can be aggregated and create a system that will solve the need for searching of individual operations and tools.

III. LITERATURE SURVEY

[1] The Go Programming Language was created at Google by Rob Pike, Ken Thompson and Robert Griesemer. Before introducing Go, Google had a development process that was not scaling well due to slow builds, uncontrolled dependencies, hard to read code, poor documentation. The purpose of this paper was to review the current state of the language. The entire software development process of Go has been analysed and also studies the compilers and cross-compilation support in Go. Go is an easy language with a simple type system. It has a concise language specification and lacks complicated constructs. A reduced build time and easy detection of mistakes is the aim of Go. [2] Go is an object-oriented programming language and has a syntax similar to C. It has multiple features like an extensive library system, fast compilation and concurrency primitives. Static and dynamic typing is supported by Go and is designed to be safe, efficient and comprehensible. The author discusses implementation differences between the other programming languages with regards to Go. [3] In this paper, the authors have explored the character and capabilities of the language. They have implemented various functions and packages within GO and worked on multiple methods, interfaces, slices, maps and types. Concurrency and communication of Golang have been discussed and emphasized upon. Go-routine is one of the main features of Go which is based on concurrent programming. It is a function executing in parallel with other go-routines and requires very little address space. Golang has been used as a system programming language and can be used for problems that require text processing and script programs.

©IJRASET: All Rights are Reserved

IV. PROBLEM STATEMENT

We propose to design a GO Library/CLI tool which provides data analysis and conversion utilities to the user. The library will include a host of operations which will convert, encrypt and decrypt the data fed by the user.

V. SYSTEM ARCHITECTURE

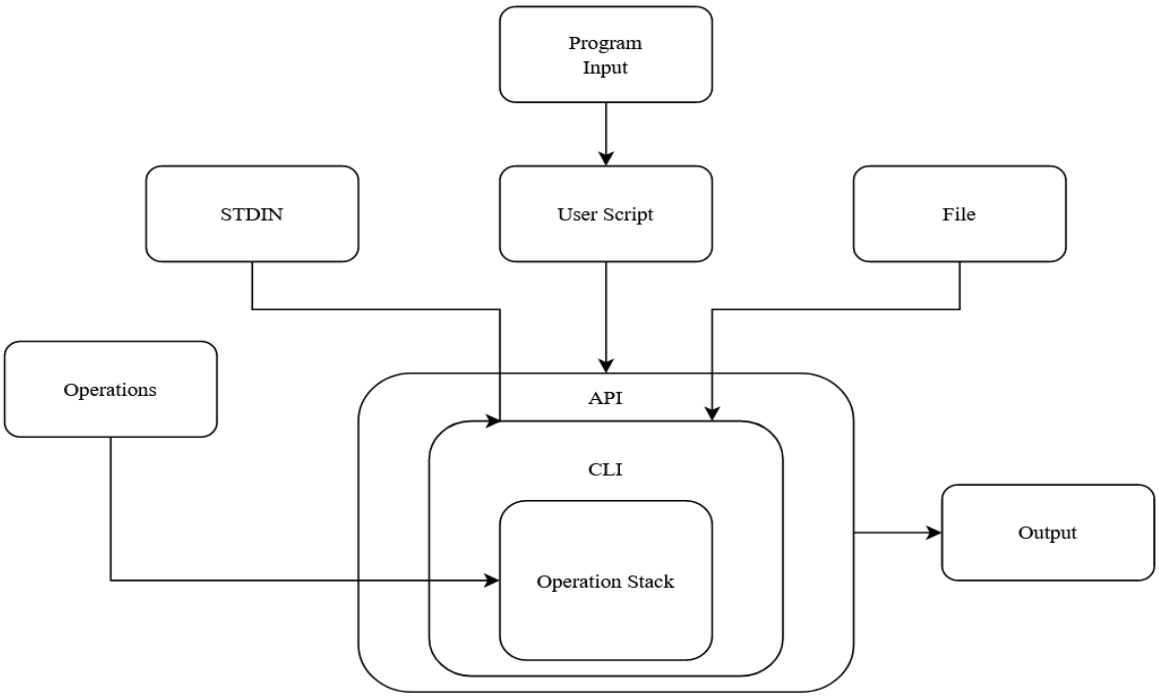


Fig. 1 System Architecture

The user uses the CLI tool to input the file from his device. They can use any type of data of choice on which the operations are to be performed. The user script is a script that contains the necessary classes and functions required for the operations to communicate with the data and CLI. The operation stack contains a list of operations that is automatically accessed by the API. The program performs the operations as per the user requirements and displays the output in the required format.

VI. SYSTEM IMPLEMENTATION

The system has the following methods of implementation:

1) *Input:* The user will be able to upload the data of his choice and can be in any format. This data will serve as the object to be analysed or converted as per user requirements.
2) *Processing:* The data in the file is processed according to the operation requested by the user. The operation stack contains the list of operations present. The CLI tool is used to access this stack. A list of operations that can be performed may include decoding a Base64-encoded string, conversion of data from hex dump followed by decompression and AES decryption.
3) *Output:* The output will be in the desired format.

VII. APPLICATIONS

1) *Forensics:* Computer forensics is a new field that makes use of this extended library. Steganography is the technique of hiding secret data inside an ordinary file or message. This tool can be used to extract such information.
2) *Testing:* It provides software developers with a way to test which hashing algorithms or encryption methods will be useful and check the overall strength of the web application.
3) *Data Compression:* LZW data compression can be performed using the application. Fast execution and lossless output are a feature of this compression method.

©IJRASET: All Rights are Reserved

VIII. FUTURE WORK

Golang has come a far long way since its invention and the language is still evolving and expanding. However, the support and feature set has scaled up and offers many more choices to software developers and programmers. Since Go is simple and efficient, scalability is possible without compromising the performance.

Programmers and developers will be able to design new functionalities which will allow the user to perform more operations and explore features built into the system. Tools for encoding and decoding can be implemented for text as well as ciphers. Compression and decompression of data are possible using deflate algorithm. Utility operations like whitespace removal, alphabet count, unit conversion and colour code parsing can be added for general purpose use.

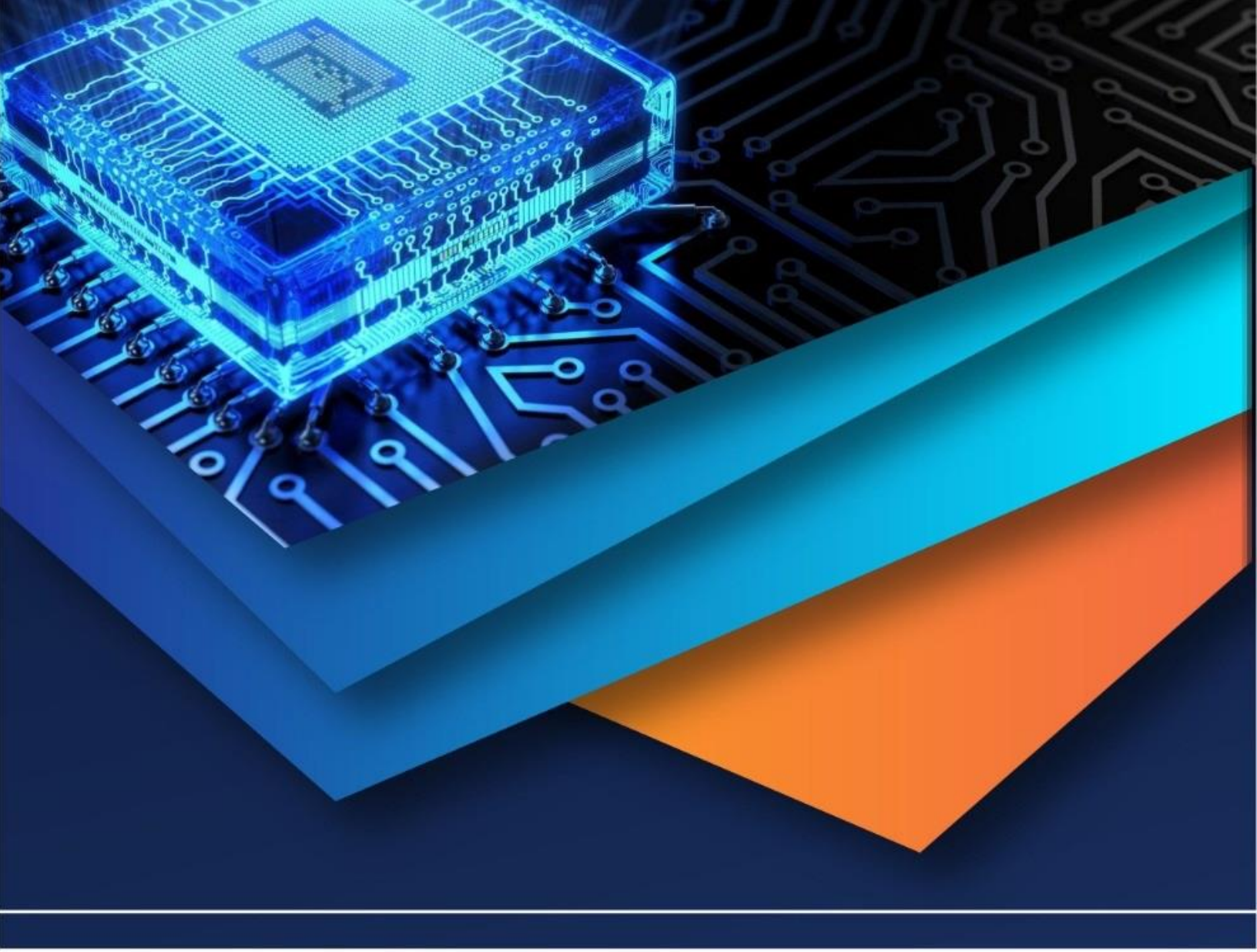
IX. CONCLUSION

In this paper, we have proposed a system that integrates Go and CLI and provide us with an opportunity to build one for various systems. Since Go is similar to C and has multiple advantages and features, it allows us to create a lightweight yet vast library. The system will also be efficient and fast due to the simplicity of Golang. The future scope of Golang and CLI is vast with the possibility of the addition of multiple tools and operations.

REFERENCES

[1] Westrup, Erik and Fredrik Pettersson, Using the Go Programming Language in Practice, 2014.
[2] Shaikh, Mrs. Bareen & Borde, Sangeeta, Quantitative Evaluation by applying metrics to existing "GO" with other programming language, International Conference on Ongoing Research in Management And IT, Jan. 2013.
[3] Magdalina Todorova, Maria Nisheva-Pavlova, Georgi Penchev, Trifon Trifonov, Petar Armyanov and Atanas Semerdzhiev, The GO programming language – characteristics and capabilities, Annual of “Informatics” Section Union of Scientists in Bulgaria, Volume 6, 2013, 76–85.
[4] Rob Pike, The Go Programming Language, GO Talk, Oct 2009.
[5] I. L. Taylor, The go frontend for gcc, 2010. [Online]. Available: http://talks.golang.org/2010/gofrontend-gcc-summit-2010.pdf
[6] Gccgo in gcc 4.7.1, 2012. [Online]. Available: http://blog. golang.org/gccgo-in-gcc-471

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)