School of Computer Science & Software Engineering
Bachelor of Computer Science (Digital System Security)

PROJECT REPORT

CSCI321 – Project – Krypto for iOS

Group: SS16-2C

Branata Kurniawan	4640949
Joshua Saputra	4641607
Yang Guang	4946054
Steven Lee	4643483

Abstract

This document is the Technical Manual for the Cryptanalysis project that contain the technical information of this application. This is a part of Final Year Project (CSCI321) for University of Wollongong

Team

NAME	STUDENT ID
Steven Lee • Project Leader • Lead Designer	4643483
Joshua Saputra • Programmer • System tester	4641607
Branata Kurniawan • Programmer • System tester	4640949
Yang Guang • Programmer • System tester	4946054

Supervisor : Dr. Ta Nguyen Binh Duong
Assessor : Mr. Tan Kheng Teck

Table of Contents

Introduction 5

System Overview 5

Application Features 5

Project Website 6

Stakeholder 6

Intended Readership 6

Installing and Open the Application 6

Uninstalling the Application 6

Technical Manual 7

- **Language** 8
- **Supported Device** 8
- **External Code** 8
- **Risk Assessments** 9
- **Development Methodology** 9
- **Wireframe** 10
- **Version Control Record** 13
- **Application UML** 17
- **Function Description** 18

User Manual 32

Test Case 48

Appendix 57

Document Status Sheet

Document Title	Project Report
Document Identification	(Directory for this technical manual)
Author(s)	Branata Kurniawan, Joshua Saputra, Yang Guang, Steven Lee
Version	0.2.0
Document Status	Draft / Internally Accepted / Conditionally Approved / Approved

Version	Date	Author(s)	Summary
0.0.1	25-06-2016	Branata Kurniawan	Document Creation
0.0.2	28-06-2016	Branata Kurniawan	Adding technical document
0.0.3	29-06-2016	Joshua Saputra	Adding user manual document
0.1.0	09-08-2016	Branata Kurniawan	Updating technical manual section
0.1.1	11-08-2016	Joshua Saputra	Updating user manual section
0.1.2	13-08-2016	Branata Kurniawan	Updating class diagram
0.1.3	14-08-2016	Yang Guang	Updating the wireframe section
0.1.4	16-08-2016	Branata Kurniawan	Updating Function Descriptions
0.1.5	17-08-2016	Branata Kurniawan	Updating the Risk Assessments
0.2.0	20-08-2016	Steven Lee	Adding the test case section

Document Change Report

Document Title	Prototype Project Report
Document Identification	(Directory for this technical manual)
Date of Changes	25-06-2016

Document Title	Project Report
Document Identification	(Directory for this technical manual)
Date of Changes	09-08-2016

Introduction

The technical manual provides information starting with the application features, user manual of the application, test case which used to test the system and lastly the technical manual of the application. Managing the project and understanding the process of this project's development can be achieved through this report. This document also provides the class diagram (UML) to understand the system architecture where MVC (Model – View – Controller) being used as the software architecture pattern in this project.

System Overview

Cryptanalysis Software (CS) is used to do cryptanalysis for any classical cryptogram. CS hides complexity in detecting hint when decrypting with the user interface which will make it easy to use. It also includes many functional tools to helps user to do analysis. Additionally, CS has Quiz feature that help the user to be better in understanding Cryptanalysis. CS use Tabbed Application architecture and build in XCode 7.2 using Swift 2.3 Language. This software is going into final stage of development. Since it is using Swift 2.3 language, CS need to be run in any recent iOS device with latest iOS version 9.3.

Application Features

- **Complete Mobile App**

  Means a complete computer program design to run on the mobile devices such as smartphones and tablet computers.

- **Analyse Tools**

  Provide analyse tools for Classical Cipher such as Shift, Substitution, Affine, and Polyalphabetic. The tools are Frequency Analysis function, which include Frequency Graph, Polygraph, Phase Frequency, and also tools used to analyse Transposition and Playfair.

- **Decryption Tools**

  Able to decrypt the cipher text into plaintext given the cipher text and key. Available for Affine, Substitution, Shift, Polyalphabetic, Transposition and Playfair cipher.

- **Auto-Decryption Tools**

  Able to auto decrypt the cipher text into plaintext without the key. Available for Affine, Polyalphabetic, and Shift cipher.

- **OCR**

  Make it possible for user to input text by taking photo of the cipher text, and then the application will translate the photo into text form.

- **Calculator**

  Provide some useful calculators function to help the student taking cryptography module. This tools help the user to calculate Inverse Mod, GCD, Fast Exponential, and IC values.

- **Cipher Text Generator**

  Generate random cipher text with random key for the user to practice their skill to find out about the key and solve the cipher.

Project Website

This project website states the purpose of this website, the aim on this project and promotes the application to its intended viewers. It also provides the project supported document such as technical manual, user manual, project proposal, and also the project diary under '*Documents*' section.

As for the web development platform, we used Wix to develop the website because it provides the drag and drop utilities which make it easier to use, in addition to that it gives us the flexibility in modifying the website to suit our needed.

And lastly, here is the link to our website: http://simfyp.wixsite.com/cryptanalysis

Stakeholder

The stakeholder involved in this project is all the four team-member who are building the Application from scratch. The supervisor, who is Dr. Ta Nguyen Binh Duong. He directs the progression of the work and also gives some suggestion on things that we can add to improve the application. Lastly the assessor will be Mr. Tan Kheng Teck who will be evaluate the project.

Intended Readership

We hope this document will address the needs of the following users of the Cryptanalysis Software:

- The cryptographer who want to break the classical cipher.
- The student who learn cryptography and need to understand how to break classical cipher.
- The data statistic provider which needed to get the statistical data.

Installing and Open the Application

At current state, they need to be built from source directly. Below is the description on how to build the source using XCode:
Requirement: You need at least 10.6 version of fully-functioning OS X

1) **Installing Xcode**. This can be done by going into App Store available in every iOS. Search for 'Xcode' apps and click download. Latest version of this app should be always available through App Store. After download is complete, install Xcode using the instructions suggested by the App.

2) **Opening Project.** After successful installation, *double click* on Xcode to run the apps. In the first menu, at the bottom-right end choose '*open another project'*. Choose '*Cryptanalysis.workspace'* (with white logo) and click Open.

3) **Run Project.** At the top-right, choose one emulator (it can be either iPhone 6 or iPhone 6 plus) and click *'Play'* button. Software is ready-to-go after it is loaded successfully. Note that it may be take some time to load the emulator.

Uninstalling the Application

Software can easily be removed by deleting it directly either from emulator or from the iPhone.

Technical Manual

Language

In Cryptanalysis (our application's name), language that being used is Objective-C and Swift 2.2.

Swift is a new language that Apple developed and some of the functionality still uses Objective-C. Therefore, around 80% of Swift and 20% of Objective-C is used to develop this application. The Objective-C is only used to integrate interactive user interface.

Supported Device

Swift 2.2 is supported for device with iOS 9 or above, and here are the products of Apple which are able to upgrade the OS to iOS 9, here is the list:

- iPhone 5
- iPhone 4s
- iPhone 5c
- iPhone 5s
- iPhone SE
- iPhone 6
- iPhone 6 Plus **(best)**
- iPod touch 5th gen
- iPad mini
- iPad Air
- iPad mini 2
- iPad 2
- [new] iPad
- iPad Air 2
- iPad mini 3
- iPad Pro

External Code

We use several external code to improve the experience of user when using the system. The external code is an open-source code which the author published it for other developer. Even though the code is complete the team still needed to edit some part of the code to fit in to the application and for the function working perfectly. Here is the list of External code that we use:

1. Tesseract-iOS-OCR
   Open-Source code for OCR function which transform a photo into the text form.
   Ref: https://github.com/gali8/Tesseract-OCR-iOS

2. GPUImage
   Open-Source code to help with compress the images inside the application user interface.
   Ref: https://github.com/BradLarson/GPUImage

3. Charts
   Open-Source code to help generate the graph inside the application.
   Ref: https://github.com/danielgindi/Charts

4. Holy View
   Open-Source code to create a first time used tutorial on the application.
   Ref: https://github.com/Frexas/HolyView

Risk Assessments

Description of Risk	Possible Consequences	Level of Risk	Contingency Plan
There is a small change on the UI if we run the application on other apple devices like iPhone 4s other than iPhone 6s	The button and text field will be smaller which will be hard for user to deal with	Low	Only on iPhone 4s, we decide to take off the logo to make the text field bigger
User's camera cannot capture the photo clearly when using OCR features	The OCR function cannot transform the photo into text correctly	Medium	User can help with the lighting when taking the photo of the text, or make sure that the paper is not crumpled
It’s hard for the user which haven't study cryptography yet to use the application	The user will not be able to operate the application correctly	Medium	By implementing the first time tutorial which explain the button, text field, and etc.
User's devices have a slow processor	The user will counter the problem with OCR and auto decrypt features which both features will be taking very long time to complete	High	We try to decreased the loop used in the auto decrypt and also train the OCR more to make the OCR be able to recognise the text in the photo faster and correctly

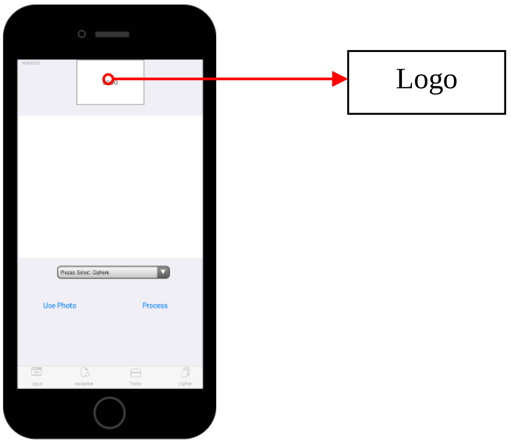
Development Methodology

As for previously, RUP framework was used as our development methodology. But as the time passes RUP framework didn’t suit us the best. So that we change the development process into extreme programming which to improve the quality of the software. Additionally, extreme programming involves a short development cycle. This helps with our productivity and be able to add additional features in a short time frame.

Wireframe

Wireframes are navigable prototype, which make it possible for us to roughly design the user interface of the application. And here are our Wireframes which we used as a base for UI (user interface) of our application.

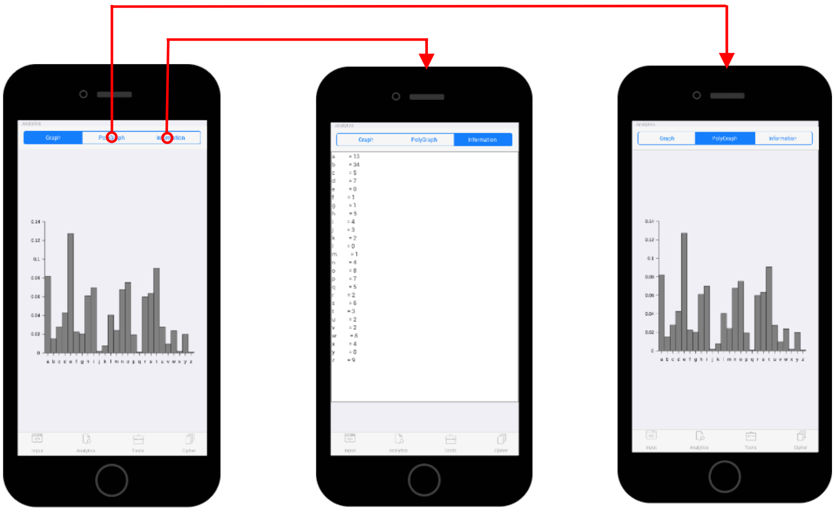
First Tab



Input View

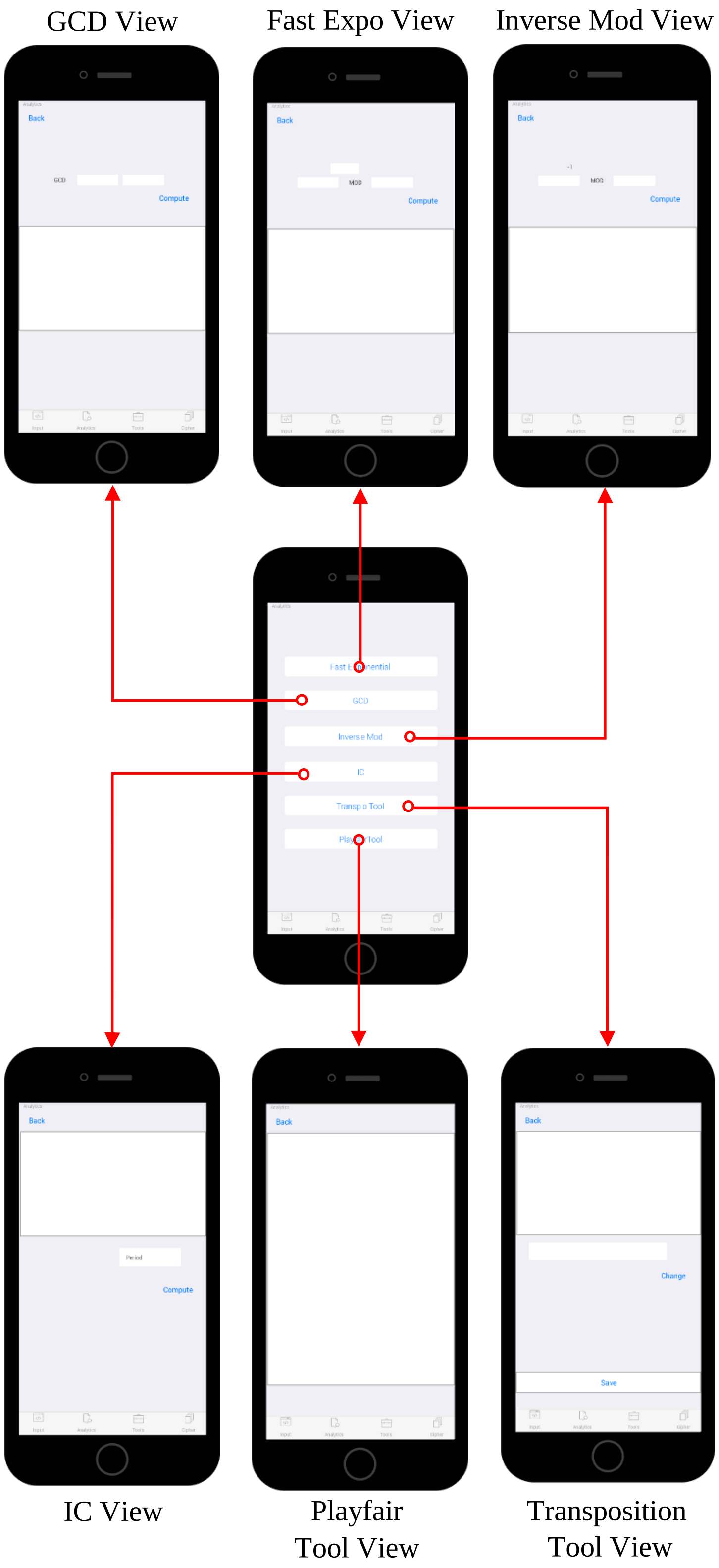
`

Second Tab

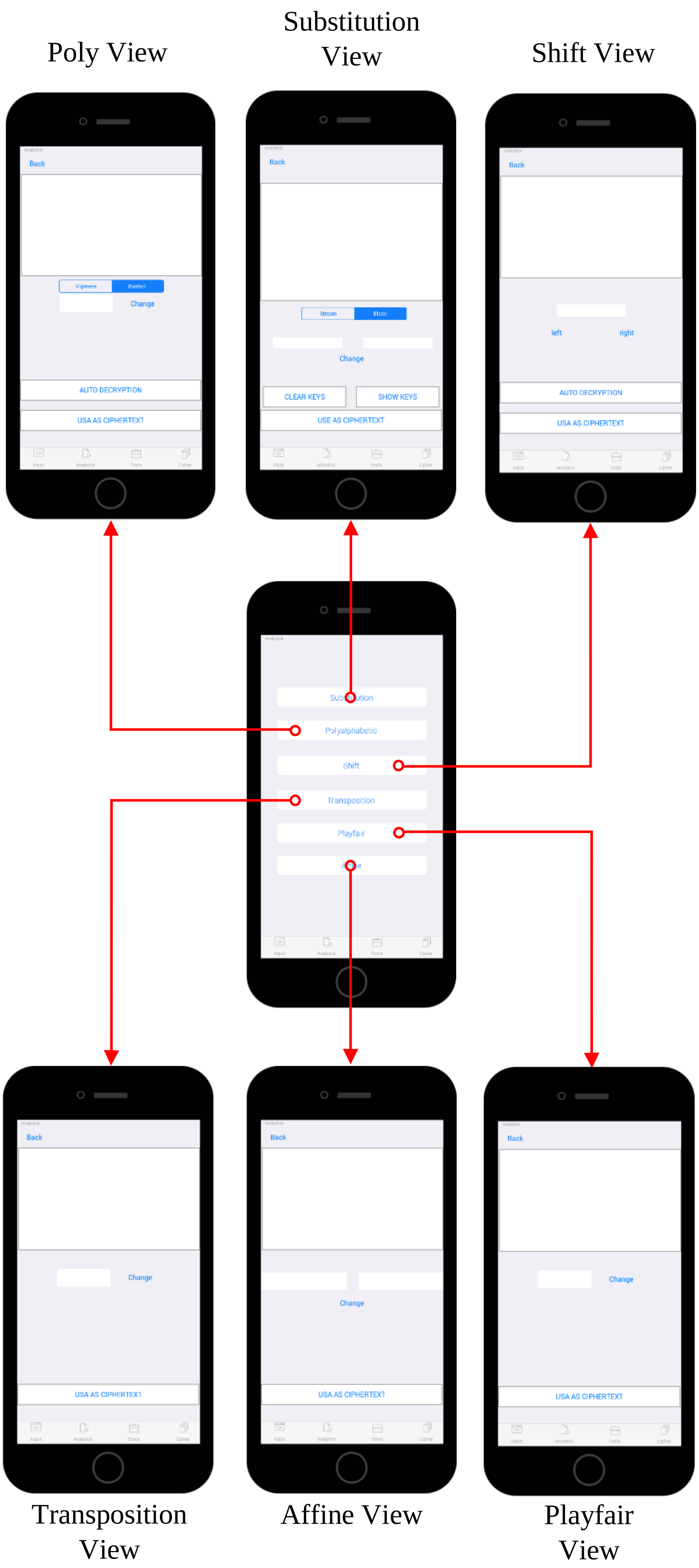


Graph View Info View Polygraph View

Third Tab


GCD View
Fast Expo View
Inverse Mod View
Back
GCD
MOD
-1
Compute
Fast Exponential
GCD
Inverse Mod
IC
Transpo Tool
Playfair Tool
Period
Change
Save
Input
Analytics
Tools
Cipher
IC View
Playfair Tool View
Transposition Tool View

Fourth Tab


Poly View
Substitution View
Shift View
Back
Vigenere
Baefort
Change
AUTO DECRYPTION
USA AS CIPHERTEXT
Back
Stream
Block
Change
CLEAR KEYS
SHOW KEYS
USE AS CIPHERTEXT
Back
left
right
AUTO DECRYPTION
USA AS CIPHERTEXT
Input
Analytics
Tools
Cipher
Substitution
Polyalphabetic
Shift
Transposition
Playfair
Affine
Back
Change
USA AS CIPHERTEXT
Back
Change
USA AS CIPHERTEXT
Back
Change
USA AS CIPHERTEXT
Transposition View
Affine View
Playfair View

Version Control Record

Version control record is based on each of the branch created via GitHub, here is the record:

1. *MonoDecryption*

Created at 19 Jun 2016. This branch contains all code about MonoAlphabetic implementation. Changes:

- 19 Jun 2016: Creation Date, Basic requirement code
- 4 Jul 2016: Introduction of Auto-fill to make user easily type key without leading alphabet
- 4 Jul 2016: Introduction of Segmentation Stream and Block cipher

Current Functionality:

- Basic replacing method of either letter or word.
- Auto-Fill feature that let user type only the key instead keys with leading alphabet.
- Stream and Block Cipher that allow user to replace word either with letter or word.

This branch is in responsibility of Joshua Saputra

2. *PolyDecryption*

Created at 11 Jun 2016. This branch contains all code about PolyAlphabetic implementation. Changes:

- 16 Jun 2016: First Iteration code, Fixed bug, Building of PolyDecryptionModel
- 17 Jun 2016: Fix conflict with global functions
- 18 Jun 2016: Merged with master branch

Current Functionality:

- Decryption of both Vigenere and Beaufort using specified key

This branch is in responsibility of Branata Kurniawan

3. *Graph*

Created at 11 Jun 2016. This branch contains all code about Graph implementation in Second Tab.
Changes:

- 12 Jun 2016: First iteration of Code, finish creating both graph and information.
- 13 Jun 2016: Change UI of StatisticalViewController. Combine two separate view inside a view
- 17 Jun 2016: Merged with master branch

Current Functionality:

- Displaying graph which represent frequency of given cipher-text
- Displaying information such as Monogram, bigram and Trigram
- Displaying graph which show frequency based on index and period

This branch is in responsibility of Steven Lee

4. *OCR*

Created at 11 Jun 2016. This branch contains all code maintaining OCR in Cryptanalysis tools. OCR allow user to get the cipher-text by directly taken photo or load from a library.
Changes:

- 17 Jun 2016: Created Initial pod for Facebook Pop and Tesseract
- 27 Jun 2016: Able to use photo from camera or library.

- 30 Jun 2016: Threshold and resize
- 24 Jul 2016: Remove Interface state, Remove cached file

Current Functionality:

- Enable user to get cipher-text shown in a photo. This photo can be taken directly from camera or library.

This branch is in responsibility of Steven Lee

5. *Calculator*

Created at 19 Jun 2016. This branch contains all code about all tools available in Third Tab of Cryptanalysis tools such as GCD, Fast Exponentiation, Inverse Mod.

Changes:

- 30 Jun 2016: First Iteration of Code. Calculator and RandomSelector implementation
- 1 Jul 2016: Adding GCD, inverse mod and fast exponential function.
- 7 Jul 2016: Fixing bug about UI.
- 13 Aug 2016: Fast exponential bug fixed.

Current Functionality:

- Introduction of Fast Exponential, Inverse Mod, GCD algorithm.

This branch is in responsibility of Yang Guang

6. *UI*

Created at 29 Jul 2016. This branch is where the latest UI in place.

Changes:

- 12 Jul 2016: MonoDecryptionImprovement and PolyDecrytpionImprovement merged into this branch. Initial UI of First Tab view is done here.
- 21 Jul 2016: Added Information UI
- 23 Jul 2016: Adding graph UI for Polygraph analysis
- 24 Jul 2016: Creating Alert for polyalphabetic graph. Make it only possible to add number to the graph

Current Functionality:

- Display relevant UI in First Tab and Second Tab
- Display alert as part of error handling

This branch is in responsibility of Steven Lee

7. *Random Selector*

Created at 7 Jul 2016. This code represent all random function used in Cryptanalysis Tools.

Changes:

- 20 Jul 2016: Creating RandomSentence, RandomWord and RandomNumber functions
- 21 Jul 2016: Adding minimum length for all three kind of randomization.

Current Functionality:

- Support all randomization functionality across Cryptanalysis Tools

This branch is in responsibility of Yang Guang

8. *MonoDecryption Improvement*

Created at 29 Jun 2016. This is an improvement branch of MonoDecryption since if we use old one, there will be a conflict at merging later. Along with improving MonoDecryption, this branch also contains of new type of decryption.

Changes:

- 12 Jul 2016: Introducing Auto-Fill
- 24 Jul 2016: Introducing Affine Cipher
- 27 Jul 2016: Implementing Quiz, Affine, Shift, Transposition, Playfair, Mono, Poly model class

Current Functionality:

- Auto-Fill feature for Mono Decryption
- New type of Cipher: Affine Cipher
- Implementing Quiz. Arrange all class model into its place

This branch is in responsibility of Joshua Saputra

9. *PolyDecryptionImprovement*

Created at 29 Jun 2016. This is an improvement branch of PolyDecryption since if we use old one, there will be a conflict at merging later. Along with improving PolyDecryption, this branch also contains of new type of decryption.

Changes:

- 2 Jul 2016: Adding Beaufort Decryption Type
- 12 Jul 2016: Adding Shift Decryption.
- 12 Jul 2016: Merging with UI Branch

Current Functionality:

- New branch to fix some bug in PolyDecryption model class
- Implementing Beaufort Decryption Type
- Implementing Shift Decryption Type

This branch is in responsibility of Branata Kurniawan

10. *newUIBranata*

Created at 27 Jul 2016. This branch contains improvement in UI stage for PolyDecryption. Additionally, there are some cipher type been added via this branch.

Changes:

- 29 Jul 2016: Introducing Transposition and Playfair Decryption
- 31 Jul 2016: Adding Affine and update button
- 31 Jul 2016: Introducing auto decryption for PolyAlphabetic

Current Functionality:

- Introducing Auto-Decryption for PolyDecryption. This will allow tools to find key directly for any given cipher-text.
- Introducing new type of Decryption called Transposition and Playfair

This branch is in responsibility of Branata Kurniawan

11. newUIJoshua

Created at 27 Jul 2016. This branch contains improvement in UI Stage for MonoDecryption.
Changes:

- 28 Jul 2016: Introducing Affine Button and Segmentation for Affine (Encrypt/Decrypt)

Current Functionality:

- Implementing UI for Affine Cipher.

This branch is in responsibility of Joshua Saputra

12. OCRUI

Created at 24 Jul 2016. This is upgraded version of UI Branch which contains latest update for UI in Cryptanalysis tools.
Changes:

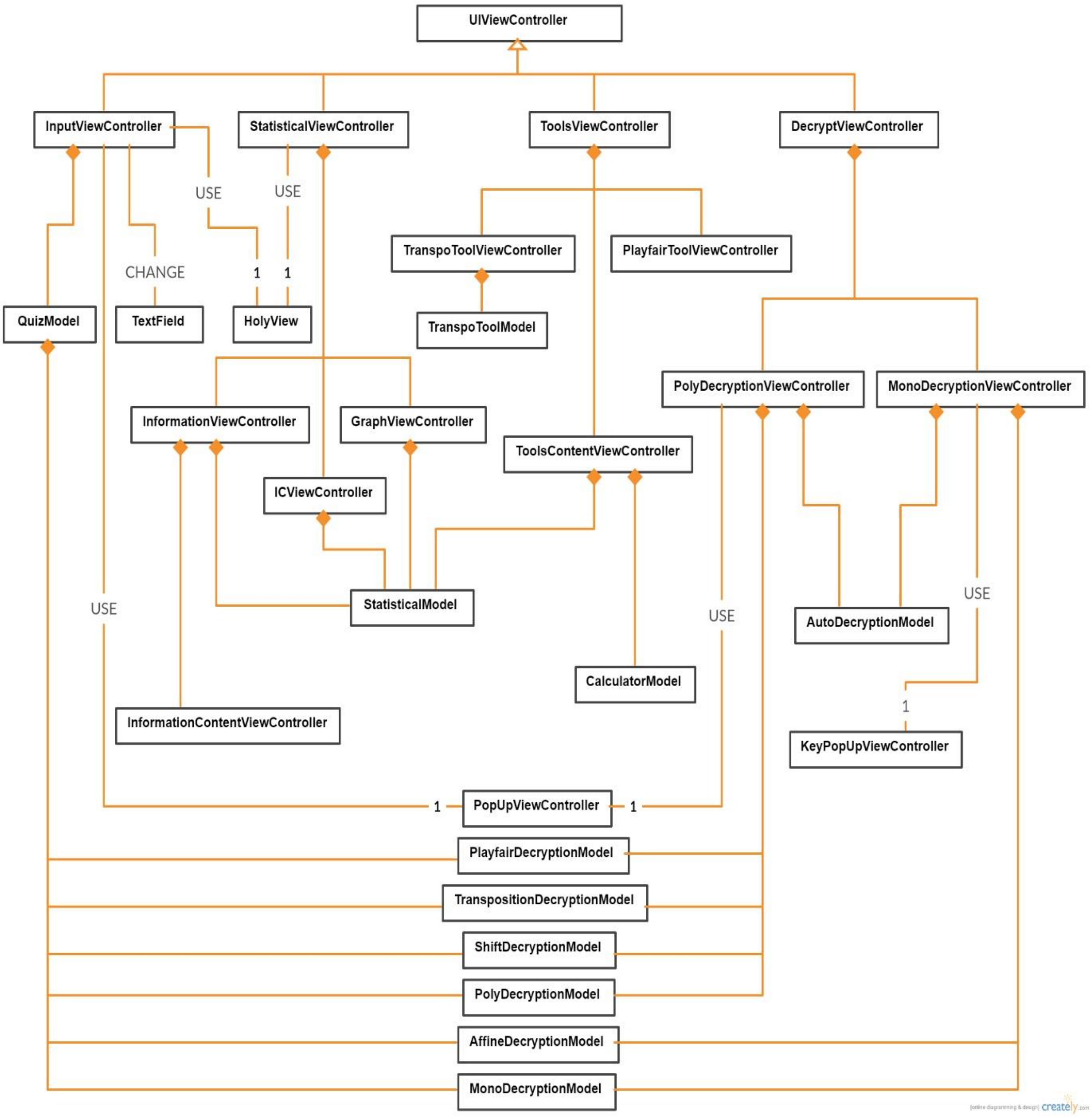
- 24 Jul 2016: Removed Cached File
- 25 Jul 2016: Creating interactive UI when photo is being processed, Adding GPU Image framework
- 27 Jul 2016: Adding 'Creating Quiz' button for Cipher-text in First Tab
- 29 Jul 2016: Putting key to pop view
- 31 Jul 2016: Improve UI for all Cipher-text
- 3 Aug 2016: Added auto-decryption UI, Fixed bug in auto-decryption for Shift
- 7 Aug 2016: Add UI for Decryption View
- 8 Aug 2016: Updating overall layout
- 9 Aug 2016: Finish overall UI, adding stack view
- 10 Aug 2016: Fix some issue and add test unit
- 11 Aug 2016: Added unit test and improvement to UI
- 12 Aug 2016: Added another unit test and change background colour on the UI, Change proper class name
- 13 Aug 2016: Put tesseract framework to the class
- 14 Aug 2016: Fixing calculator model, adding constraint to Polyalphabetic tools

Current Functionality:

- Represent all latest UI update of all four tab of the tools
- Added Unit Test

This branch is in responsibility of Steven Lee

Application UML



Function Description

InputViewController			
Function Name	Return Type	Parameter	Description
dismissKeyboard	-	-	This function will dismiss keyboard when the user tap on the screen other than the input boxes.
textViewDidChange	-	textView (UITextView)	This function will function will reassign the global variables when the text input is changed.
disableInitButtons	-	-	This functions will disable few buttons that are not supposed to be pressed before some action is done.
textField	Boolean	textField (UITextField), Range (NSRange), String (String)	This function will filter allowed characters which in this function is none so the picker view cannot be inputted.
numberOfComponentsInPickerView	Int	pickerView (UIPickerView)	Return number of component that can be chosen in picker view.
pickerView	Int	pickerView (UIPickerView), component (Int)	Return total number of choice for the picker view.
pickerView	String	pickerView (UIPickerView), row (Int), component (Int)	Return the options available at picker view.
pickerView	-	pickerView (UIPickerView), row (Int), component (Int)	Initialisation of the picker view.
showInfoPopup	-	sender (UIButton)	To create overlay to show the key.
onGetCipherText	-	sender (AnyObject)	To generate random cipher text based on cipher type that has been chosen at the picker view.
clearText	-	sender (AnyObject)	Clear the text in the UITextView after the button pressed.

showTutorial	-	-	This function will the tutorial when the app is open for the first time.
viewInitialization	-	-	This function will initialize all the component for the view.

StatisticalViewController			
Function Name	Return Type	Parameter	Description
changeSegment	-	sender (UISegmentControl)	Hide the graph segment if the info segment selected and vice versa.

GraphViewController			
Function Name	Return Type	Parameter	Description
viewWillAppear	-	Animated (Bool)	When a user moves to this interface, before the interface appear, the inside code will be executed first which is refreshing the result of the result text.
frequencyLengthDidChange	-	sender (UISlider)	Change the frequency of the graph based on the UISlider moved by the user.
onSwitch	-	sender (UISwitch)	This function will determine which switch is being toggled and call either remove symbols or not case sensitive.
setCharts	-	xAxisLabels ([String]), values ([Double])	This function will generate data to be shown to the chart.
showTutorial	-	-	This function will the tutorial when the app is open for the first time.
viewInitialization	-	-	This function will initialize all the component for the view.

StasticalModel			
Function Name	Return Type	Parameter	Description
getTrimmedText	String	-	Returned the Trimmed Text to the user.
removeSpecialCharsFromString	String	text (String)	Remove Special Characters from the text such as symbol, number and space.
generateChart	-	lengthOfCharacter (Int)	Generate the chart data and stored in into variable Data.
getXAxisLabel	String	-	This will return array of String of x axis label.
getXAxisData	Double	xAxisLabels (String)	This will return the data corresponds to the x axis data.
getStaticalInformation	String	-	This will return text based information of the static.
generatePolyGraph	-	index (Int), period (Int)	Generate the data needed to form the graph using the index and period.
getPolyGraphXLabel	[String]	-	Return the array if String of x axis label for poly graph.
getPolyXAxisData	[Double]	-	Return the data corresponds to the x axis data for poly graph.
calIC	[Double]	text (String), period (Int)	Return array of float number which the result of calculating the IC of the text based on the period.
getAverageIC	Double	text (String), period (Int)	Return a single float number which is the average from all of the IC calculated for the text.
estimatedKeyLength	Int	text (String)	Estimate the key length of the text by calculating its IC.

ICViewController			
Function Name	Return Type	Parameter	Description
registerForKeyboardNotifications	-	-	To add observer to the keyboard when the keyboard is shown or dismiss.
deregisterFromKeyboardNotifications	-	-	To remove observer to the keyboard.
keyboardWasShown	-	notification (NSNotification)	To scroll up the UI when the keyboard is shown.
keyboardWillBeHidden	-	notification (NSNotification)	To roll down the view to the default position when the keyboard is hidden.
setCharts	-	xAxisLabels ([String]), values ([Double])	To generate the data to be shown at the graph.
onGenerateGraph	-	sender (AnyObject)	This function validates the input and show alert when the input is invalid.

InformationViewController			
Function Name	Return Type	Parameter	Description
putToPageView	-	-	Put information to the paged view.
viewControllerAtIndex	InformationContentViewController	index (Int)	Return the index of current page view.
pageViewController	UIViewController	pageViewController (UIPageViewController), viewController (UIViewController)	Initialization of the page view.
presentationCountForPageViewController	Int	pageViewController (UIPageViewController)	Return the total number of page view.
presentationIndexForPageViewController	Int	pageViewController (UIPageViewController)	Return the minimum of the page view.

InformationContentViewController			
Function Name	Return Type	Parameter	Description
viewDidLoad	-	-	Initialize the view.

MonoDecryptViewController			
Function Name	Return Type	Parameter	Description
changeButtonAction	-	sender (AnyObject)	It triggers an action when the change button is pressed which will change from a text to a different text
dismissKeyboard	-	-	This function will dismiss keyboard when the user tap on the screen other than the input boxes
viewWillAppear	-	Animated (Bool)	When a user moves to this interface, before the interface appear, the inside code will be executed first which is refreshing the result of the result text
autoFillButtonAction	-	sender (AnyObject)	This function will fill remaining characters that are not in the input box.
changeButtonAction	-	sender (AnyObject)	This function will be triggered when change button is pressed which will call encryption function.
changeMono	-	-	This function will encrypt or decrypt using mono encryption.
changeAffine	-	-	This function will encrypt or decrypt using affine encryption.
addActivityIndicator	-	-	This function will add loading screen.
removeActivityIndicator	-	-	This function will remove loading screen.
isSubstitution	-	-	This function will initialize buttons for substitution cipher.

isAffine	-	-	This function will initialize buttons for affine cipher.
isAffineOrSubstitution	-	cipherType (String)	This function will call either is substitution or is affine.
onAutoDecryption	-	sender (AnyObject)	This function is triggered when auto fill is pressed and call autofill button action.
onUseAsCipherText	-	sender (AnyObject)	This function will make the result text into the input text.

MonoDecryptionModel			
Function Name	Return Type	Parameter	Description
init	-	-	Initialise the dictionary by putting the alphabet onto it.
clearStreamDictionary	-	-	Clear the stream type's dictionary before using it to decrypt or encrypt the text.
clearBlockDictionary	-	-	Clear the block type's dictionary before using it to decrypt or encrypt the text.
insertKeyToDictionaryBlock	-	userKey (String), userValue (String)	Insert the prepared key into the Dictionary for block cipher.
applyReplaceUsingDictionaryBlock	-	globalText (String)	The encryption or decryption process using the dictionary for block cipher.
insertKeyToDictionaryStream	-	userKey (String), userValue (String)	Insert the prepared key into the Dictionary for stream cipher.
applyReplaceUsingDictionaryStream	-	globalText (String)	The encryption or decryption process using the dictionary for stream cipher.
removeDuplicateLetterFromString	String	string (String)	Remove the duplicate characters on a string.
autoFillKeyString	String	keyString (String)	Return the new key which containing all 26 alphabets used to prepared the key.
resetStreamDictionary	-	-	Reset the dictionary into the initialize state.
resetBlockDictionary	-	-	Reset the dictionary into the initialize state.

getStreamDictionary	String	-	Return the dictionary used in Stream cipher.
getBlockDictionary	String	-	Return the dictionary used in Block cipher.

PolyDecryptionViewController

Function Name	Return Type	Parameter	Description
hideElements	-	-	This function will hide elements depending on which button call the view.
buttonPressed	-	sender (UIButton)	This function will be triggered when change button is pressed which will call poly, Transpo, or Playfair depending which view they are on.
onShift	-	sender (UIButton)	This function will be triggered when change on is shift which will call poly, transpo, or playfair depending which view they are on.
changePoly	-	-	This function will be triggered when either left or right button is pressed to do shift encryption.
changeTranspo	-	-	This function is encryption function for transposition.
changePlayfair	-	-	This function is encryption function for playfair cipher.
onAutoDecryption	-	sender (AnyObject)	This function will do auto encryption.

PolyDecryptionModel

Function Name	Return Type	Parameter	Description
vigenereEncryption	String	text (String), key (String)	Return cipher text encrypted using Vigenere cipher given the text and key.

beaufortEncryption	String	text (String), key (String)	Return cipher text encrypted using Beaufort cipher given the text and key.
decryptionButton	String	ctext (String), key (String), type (Int)	Decrypt the ctext (cipher text) using the key and return the plain text based on the type of cipher.
checkKey	Bool	key (String)	Check if whether the key is valid or not and return the bool based on the result of the check.
autoDecryptPoly	String	str (String), isBeaufort (Bool)	Automatically find the best possible key to decrypt the str (plaintext) and return the key.
findKeyPoly	String	bArray ([String]), fsArray ([UInt64]), keyLength (Int)	Function used in autoDecryptPoly to find the key given the length of the key.
getFitnessScore	UInt64	text (String), bigram (String), isBeaufort (Bool)	Function used in autoDecryptPoly to find the decrypted text's Fitness Score (how close the text to English word) using possible key.
getPlainText	Character	chr (Character), key (Character), isBeaufort (Bool)	Return the text after using the key to decrypt it.

QuizModel			
Function Name	Return Type	Parameter	Description
generateMonoalphabetic	-	-	Generate the text encrypted using Substitution cipher.
generateVigenereCipher	-	-	Generate the text encrypted using Vigenere cipher.
generateBeaufortCipher	-	-	Generate the text encrypted using Beaufort cipher.
generateAffineCipher	-	-	Generate the text encrypted using Affine cipher.
generateShiftLeftCipher	-	-	Generate the text encrypted using Shift Left cipher.
generateShiftRightCipher	-	-	Generate the text encrypted using Shift Right cipher.

generateTransposition Cipher	-	-	Generate the text encrypted using Transposition cipher.
generatePlayfairCipher	-	-	Generate the text encrypted using Playfair cipher.
getPlainText	String	-	Return the plain text generated randomly by the program
getKeyWord	String	-	Return the key word used to encrypt the plaintext to produce the cipher text
getCipherText	String	-	Return the cipher text
randomSentence	String	-	Generate a random sentence used for plaintext
randomWords	String	sentence (String)	Generate random word to form a sentence
randomNum	Int	max (Int)	Generate a random number given the maximal value for the number

TextField			
Function Name	Return Type	Parameter	Description
canPerformAction	Bool	action (Selector), sender (AnyObject)	Use to disable paste and select by returning the bool.

PopUpViewController			
Function Name	Return Type	Parameter	Description
onClose	-	sender (UIButton)	This function will close the overlay view when the close button is pressed.
showAnimate	-	-	This function will give animation when the view is called.
removeAnimate	-	-	This function will show remove animation when the view is being closed.

ShiftDecryptionModel			
Function Name	Return Type	Parameter	Description
shiftLeftEncryption	String	text (String), key (String)	Return cipher text encrypted using Shift Left cipher given the text and key.
shiftRightEncryption	String	text (String), key (String)	Return cipher text encrypted using Shift Right cipher given the text and key.
decryptionButton	String	ctext (String), offset (String), type (String)	Decrypt the ctext (cipher text) using the key and return the plain text based on the type of cipher.
shiftRightDecrypt	String	text (String), offset (Int)	Return the plaintext given the cipher text and offset, decrypted using Shift Right cipher.
autoDecryptShift	String	str (String)	Function to automatically return the best possible key to decrypt the text using Shift Right and Shift Left cipher.

PlayfairDecryptionModel			
Function Name	Return Type	Parameter	Description
playFairEncryption	String	text (String), key (String)	Return cipher text encrypted using Playfair cipher given the plain text and key.
prepareKey	[[String]]	key (String)	Transform the key into a table (2D array) used to encrypt and decrypt the text later on.
autoFillKeyString	String	keyString (String)	Return the new key which containing all 26 alphabets used to form the table.
removeJLetter	String	text (String)	Return the text back after removing the J from it.

checkRepetition	String	text (String)	Check if there is 2 characters is the same and located side by side, if it is true than the program adds another character to distinguish it.
replaceJWithI	String	text (String)	Return the text back after substitute the J with I.
groupingTheText	[String]	text (String)	Return an array which each element containing the pair of characters from the text.
getEncryptionValue	String	text [String], keyTable ([[String]])	Return the cipher text after encrypted using the table.
getDecryptionValue	String	text [String], keyTable ([[String]])	Return the plaintext after decrypted using the table.
decryptionButton	String	text (String), key (String)	Decrypt the ctext (cipher text) using the key and return the plain text.
analyzePlayfair	String	text (String)	Return the text back to make it for user to analyse it.

TranspositionDecryptionModel

Function Name	Return Type	Parameter	Description
transpositionEncryption	String	text (String), key (String)	Return cipher text encrypted using Transposition cipher given the plain text and key.
randomChar	Character	-	Generate a random char.
getKeyTable	[Int]	key (String)	Return an array of key in form of Int (number).
decryptionButton	String	text (String), key (String)	Decrypt the ctext (cipher text) using the key and return the plain text.

AffineDecryptionModel

Function Name	Return Type	Parameter	Description
getAlphaKey	Int	-	Return the alpha key used to encrypt or decrypt the text.

getBetaKey	Int	-	Return the beta key used to encrypt or decrypt the text.
applyAffineEncryptionUsingKey	String	globalText (String), alphaKey (Int), betaKey (Int)	Apply Affine encryption manually given the alpha key, beta key and the text.
applyAffineDecryptionUsingKey	String	globalText (String), alphaKey (Int), betaKey (Int)	Apply Affine decryption manually given the alpha key, beta key and the text.
autoDecryptAffine	String	str (String)	Return the best possible key used to encrypt the current text.

ToolsViewController			
Function Name	Return Type	Parameter	Description
onButton	-	sender (UIButton)	This function will set what string to be sent to the next view.
prepareForSegue	-	segue (UIStoryboardSegue), sender (AnyObject)	This function will send the string that has been set on the function onButton to the next view.

TranspoToolViewController			
Function Name	Return Type	Parameter	Description
onChange	-	sender (AnyObject)	This function will do transformation for transposition.
onSave	-	sender (AnyObject)	This function will save the changed.

TranspoToolModel			
Function Name	Return Type	Parameter	Description
analyzeByPeriod	String	text (String), period (Int)	Analyse the text by processing the text based on the period inputted by the user.
addingTab	String	text (String)	Adding the tab to the text for each of 5 characters

getTable	[String]	text (String), keyTable ([Int])	Get the table of the text after processed using the key table.
getKeyTable	[Int]	keyLength (Int)	Prepare the key table for the first time.
swapTheKey	[Int]	keyTable ([Int]), int1 (Int), int2 (Int)	Swap the key table value based on the user inputted.

ToolsContentViewController			
Function Name	Return Type	Parameter	Description
onCompute	-	sender (UIButton)	This function will be triggered when the compute button is pressed which will call computation function.
compute	-	-	Will call one of the function below depending on which button was pressed on previous view.
computeGCD	-	-	Calculate GCD number.
fastExpo	-	-	Calculate fact expo.
inverseMod	-	-	Calculate inverse mod.
IC	-	-	To calculate the IC of the text.
showAlert	-	-	To show alert box if the input is invalid

CalculatorModel			
Function Name	Return Type	Parameter	Description
gcd	String	number_1 (Int), number_2 (Int)	Calculate the gcd.
fastExpo	String	base (Int), modulus (Int), exponent (Int)	Calculate the fast expo.
gcdR	String	dividend (Int), divisor (Int)	Calculate the inverse mod.

DecryptViewController			
Function Name	Return Type	Parameter	Description
onButton	-	sender (UIButton)	This function will set what function to be sent to the next view.
prepareForSegue	-	segue (UIStoryboardSegue), sender (AnyObject)	This function will send the string to the next view to determine what button is pressed.

AutoDecryptionModel			
Function Name	Return Type	Parameter	Description
generateAutoDecryptAffine	-	text (String)	Generate the additional class and function used to auto decrypt the cipher text.
generateAutoDecryptPoly	-	text (String), isBeaufort (Bool)	Generate the additional class and function used to auto decrypt the cipher text.
generateAutoDecryptShift	-	text (String)	Generate the additional class and function used to auto decrypt the cipher text.

KeyPopUpViewController			
Function Name	Return Type	Parameter	Description
onClose	-	sender (UIButton)	This function will close the overlay view.
showAnimate	-	-	This function will show animation when the view is being shown.
removeAnimate	-	-	This function will show animation when the view is being closed.

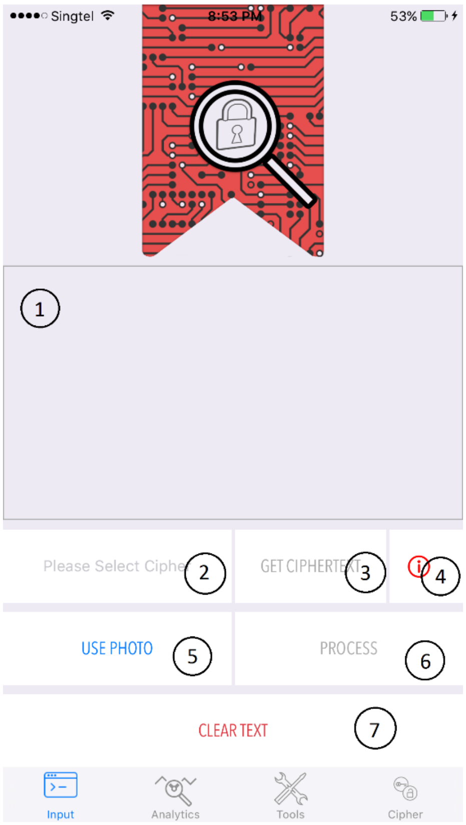
User Manual

User Manual

Cryptanalysis Software use 4 main tab for easier to use. Below is the list of the function of those tabs:

First Tab

This tab is used to entering cryptogram that need to be analysed. Additionally, cipher-text generator also been included to generate any chosen type of cipher-text that user want to use.



1. **Input Box**
   - Text field where the cipher-text is entered.

2. **Cipher Selection**
   - Scroll box of cipher-text type (Quiz)

3. **Get Cipher-Text Button (Disabled)**
   - Button to get chosen cipher-text (Quiz). It will become **Enabled** when user choose cipher type

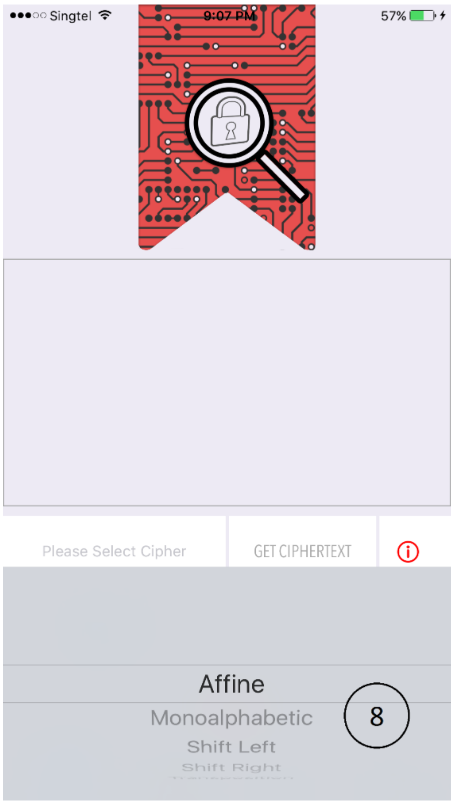
4. **Hint Button (Red)**
   - Button to get what key of chosen cipher-text (Quiz). It will become **Green** when cipher-text is generated.

5. **Use Photo**
   - Button to scan/choose picture for cipher-text

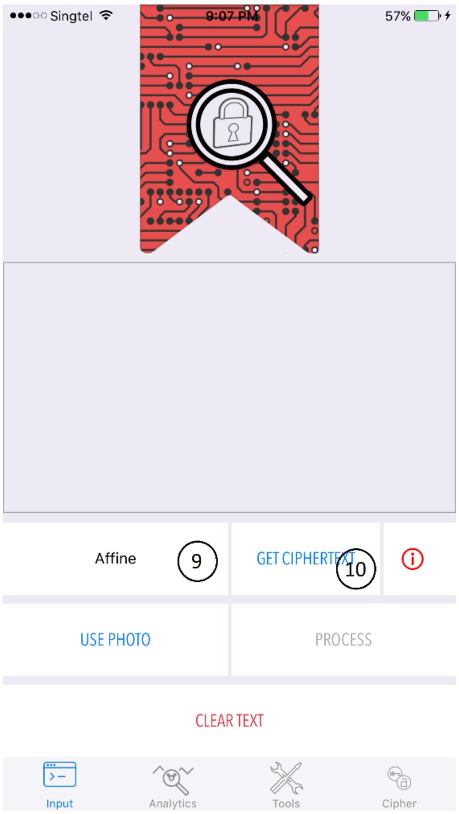
6. **Process**
   - Button to enable scanning if **Use Photo** has been chosen

7. **Clear Text**
   - Button to clear all the text in **Input Box**

***note**: User need not to always choose cipher type to use the tool. (Quiz) is used to generate cipher-text example of chosen type for user to learning

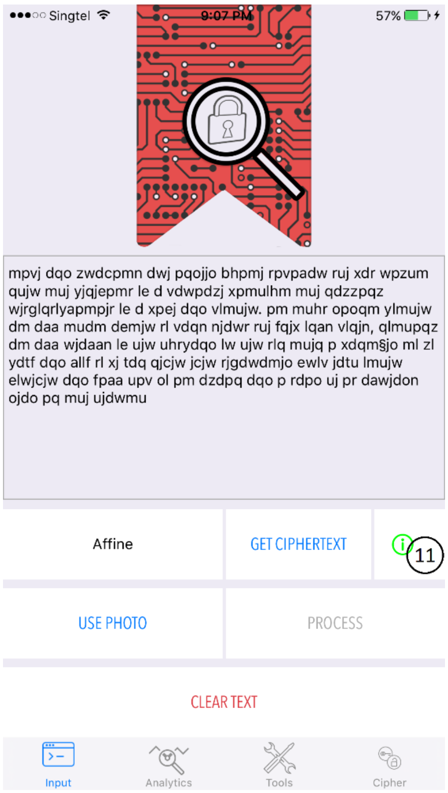


8. **Cipher Selection**
   - User can scroll down to choose what type of cipher. This will enable **Get Cipher-Text Button**.



9. **Cipher Selection (Affine)**
   - User select "**Affine**" type for example

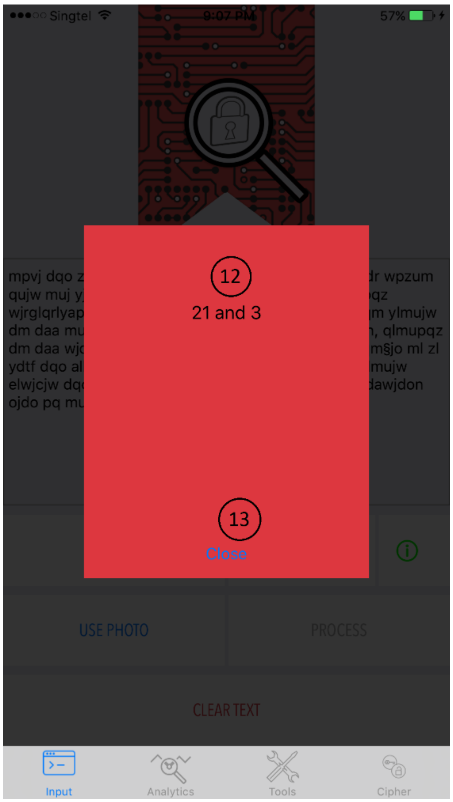
10. **Get Cipher-Text Button (Enabled)**
   - User can press this button to get random cipher-text based on type in **Cipher Selection**



mpvj dqo zwdcpmn dwj pqojjo bhpmj rpvpadw ruj xdr wpzum
qujw muj yjqjepmr le d vdwpdzj xpmulhm muj qdzzpqz
wjrglqrlyapmpjr le d xpej dqo vlmujw. pm muhr opoqm ylmujw
dm daa mudm demjw rl vdqn njdwr ruj fqjx lqan vlqjn, qlmupqz
dm daa wjdaan le ujw uhrydqo lw ujw rlq mujq p xdqm§jo ml zl
ydtf dqo allf rl xj tdq qjcjw jcjw rjgdwdmjo ewlv jdtu lmujw
elwjcjw dqo fpaa upv ol pm dzdpq dqo p rdpo uj pr dawjdon
ojdo pq muj ujdwmu

11. **Hint Button (Green)**

- Button will become green. This indicates that key can be accessed by pressing **Hint Button (Green)**.



21 and 3

12. **Hint Key (Affine)**

- This is the key for cipher-text generated. This key generated based on cipher type and cipher-text chosen.

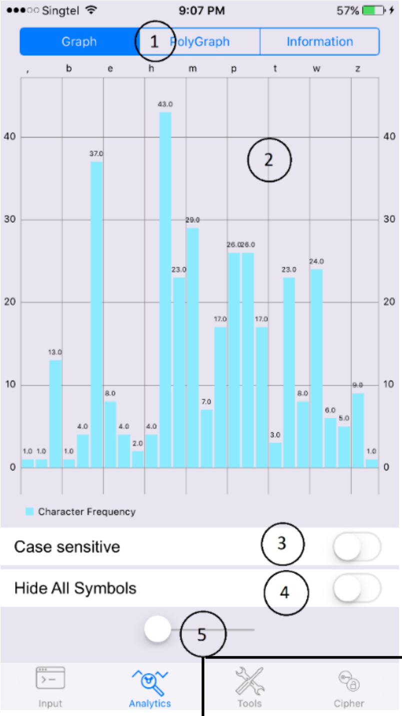
13. **Close Button**

- Close hint box.

Second Tab

This is where the graph of frequency showed for user and also Information about n-gram is presented here. 3 Sub-Tab is showed below:

1. Frequency



1. Segmentation Bar

- Bar to choose categories

2. **Graph**

- Graph which show frequency of letter from **Input Box** in First Tab

3. **Case-Sensitive Option Bar**

- Enable/disable Case-Sensitive mode

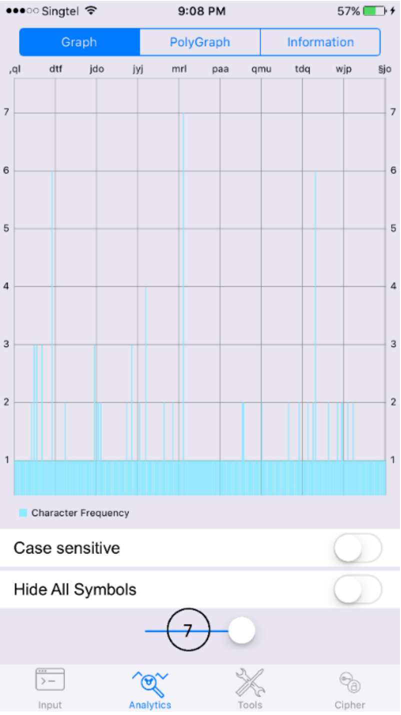
4. **Hide All Symbol Option Bar**

- Enable/disable Hide All Symbol mode

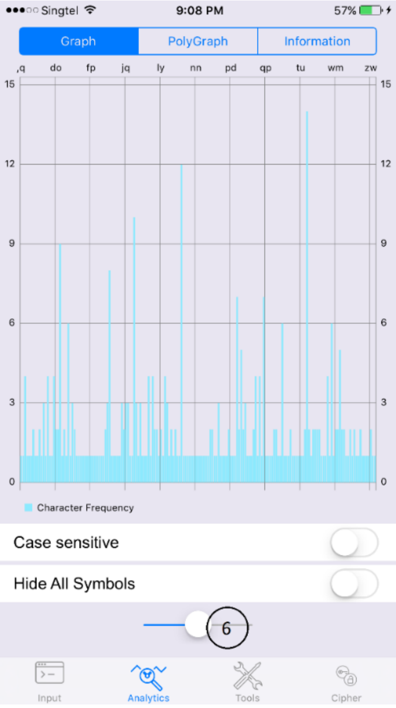
5. **n-gram Slide Bar (Monogram)**

- Bar to go from monogram up to trigram (currently on monogram)

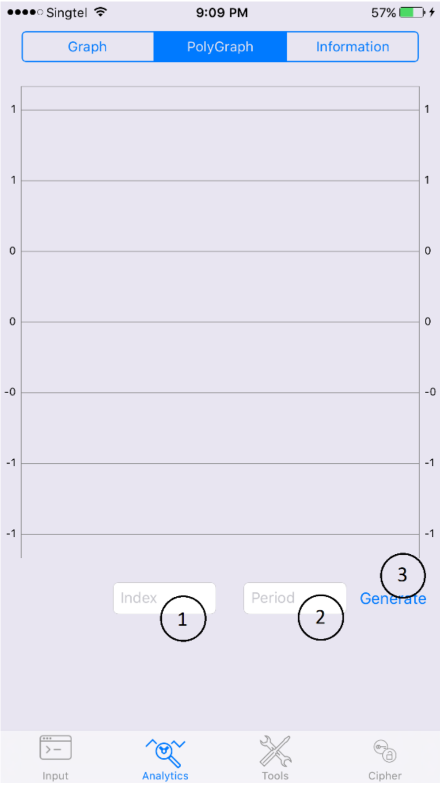
Bigram



Trigram



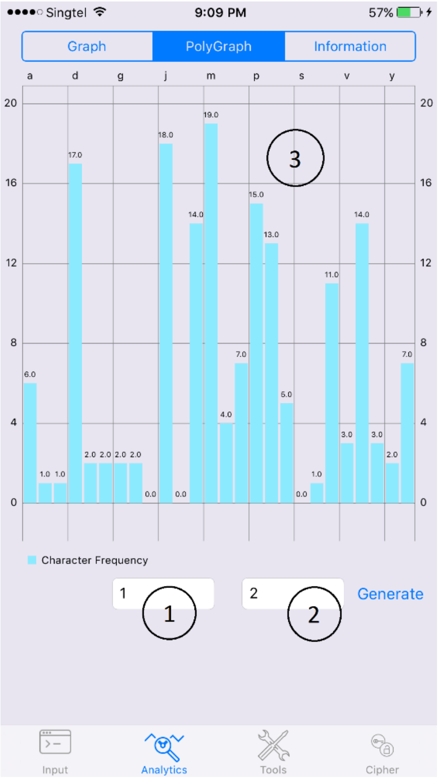
2. *Polygraph*



1. **Index Number Field**
   - Number field where user inputs index of cipher-text

2. **Period Number Field**
   - Number field where user inputs period of cipher-text

3. **Generate Button**
   - Button to generate both index and period and show graph



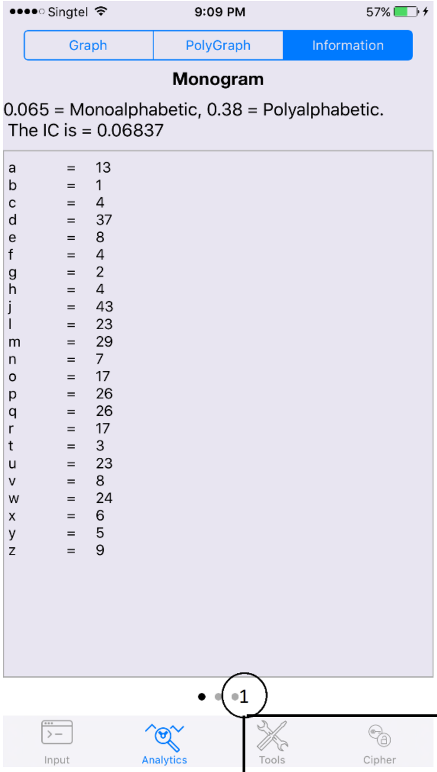
1. **Index**
   - Represents what index that the user wants to analyse. "1" means that it will take every 1st character.

2. **Period**
   - Represents the length of the key.

3. **Graph**
   - This will show the frequency analysis based on Index and Period

***note:** index of 1 and period of 2 means that Polygraph will take 1st character for every word of length 2. This used for the Polyalphabetic cipher's analytic.

3. *Information*



Monogram

0.065 = Monoalphabetic, 0.38 = Polyalphabetic.
The IC is = 0.06837

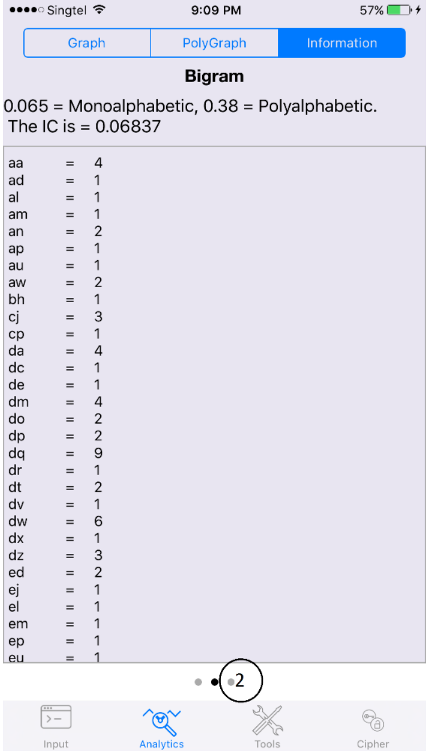
```
a    =  13
b    =  1
c    =  4
d    =  37
e    =  8
f    =  4
g    =  2
h    =  4
j    =  43
l    =  23
m    =  29
n    =  7
o    =  17
p    =  26
q    =  26
r    =  17
t    =  3
u    =  23
v    =  8
w    =  24
x    =  6
y    =  5
z    =  9
```

1. **n-gram Swipe Bar (Monogram)**

- Swipe to change n-gram from monogram to bigram and trigram. It will show all possible letter and its frequency.

Bigram

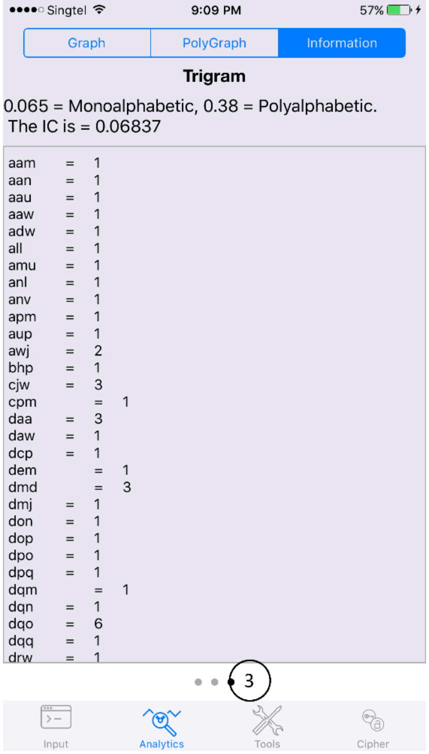
Trigram



Bigram

0.065 = Monoalphabetic, 0.38 = Polyalphabetic.
The IC is = 0.06837

```
aa   =  4
ad   =  1
al   =  1
am   =  1
an   =  2
ap   =  1
au   =  1
aw   =  2
bh   =  1
cj   =  3
cp   =  1
da   =  4
dc   =  1
de   =  1
dm   =  4
do   =  2
dp   =  2
dq   =  9
dr   =  1
dt   =  2
dv   =  1
dw   =  6
dx   =  1
dz   =  3
ed   =  2
ej   =  1
el   =  1
em   =  1
ep   =  1
eu   =  1
```



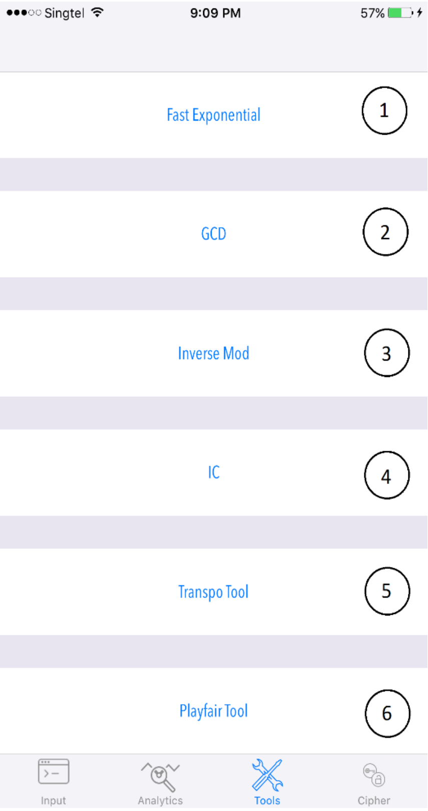
Trigram

0.065 = Monoalphabetic, 0.38 = Polyalphabetic.
The IC is = 0.06837

```
aam  =  1
aan  =  1
aau  =  1
aaw  =  1
adw  =  1
all  =  1
amu  =  1
anl  =  1
anv  =  1
apm  =  1
aup  =  1
awj  =  2
bhp  =  1
cjw  =  3
cpm      =  1
daa  =  3
daw  =  1
dcp  =  1
dem      =  1
dmd      =  3
dmj  =  1
don  =  1
dop  =  1
dpo  =  1
dpq  =  1
dqm      =  1
dqn  =  1
dqo  =  6
dqq  =  1
drw  =  1
```

Third Tab

Third tab contains tools that can be used to help user break the cipher-text and also some calculator features to help user find IC, GCD, Fast exponential and inverse modulo.



1. **Fast Exponential**
   - Fast Exponential is an algorithm that can calculate A^B mod C quickly. It is useful to calculate very big number without doing too much complex calculation.

2. **GCD (Greatest Common Divisor)**
   - GCD of A and B is the largest positive integer that divided both A and B without the remainder. This algorithm is useful to determine if A and B is co-prime or not.

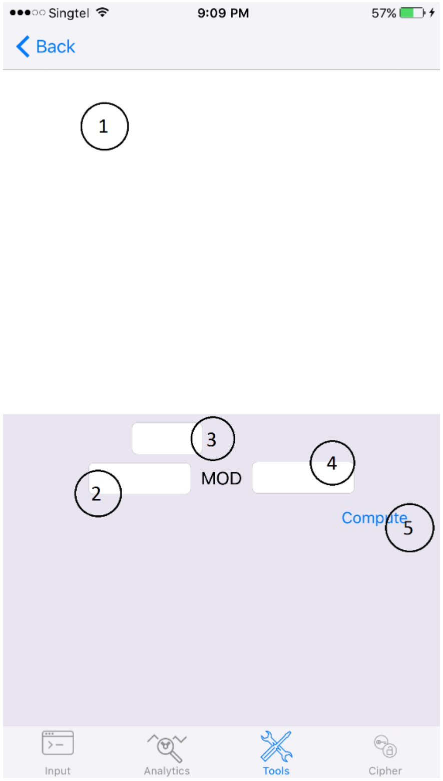
3. **Inverse Modulo**
   - Inverse Mod (or Extended Euclidean Algorithm) of a modulo m is integer x such that a * x mod m is equivalent to 1 mod m. This is useful to determine inverse of key in modern cryptography.

4. **IC (Index of Coincidence)**
   - IC is a ratio of the total or normalized by dividing by the expected amount for a random source model. In another word, IC is a way to determine the length of key of statistical-based cipher (Vigenere or Beaufort).

5. **Transposition Tool**
   - This is tool to help user to decrypt Transposition Cipher. This will show ‘pre-decrypted’ phase of a cipher-text.

6. **Playfair Tool**
   - This is tool to help user to analyse the cipher text encrypted using the playfair cipher.

1. *Fast Exponential*



1. **Output Text Field**
   - All output from calculation is displayed here

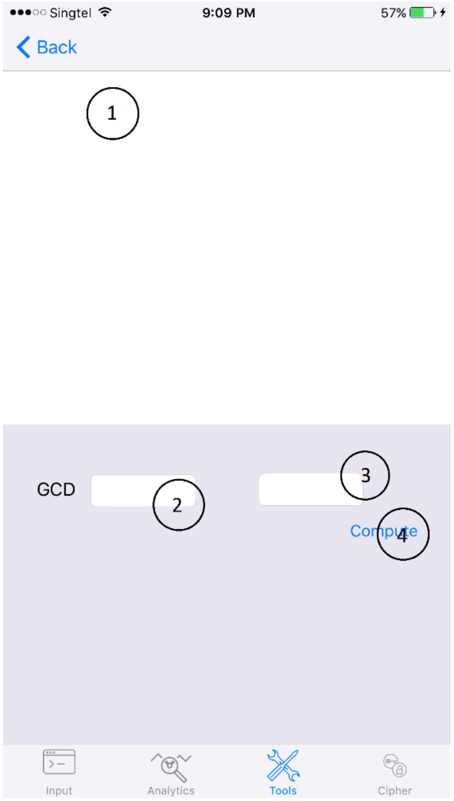
2. **Dividend Modulus**
   - Number to be divided by divisor. Can be a very big number

3. **Power**
   - Exponential Number

4. **Divisor Modulus**
   - Number that divide modulus dividend.

5. **Generate Button**
   - Button to calculate Fast Exponential

2. *GCD*



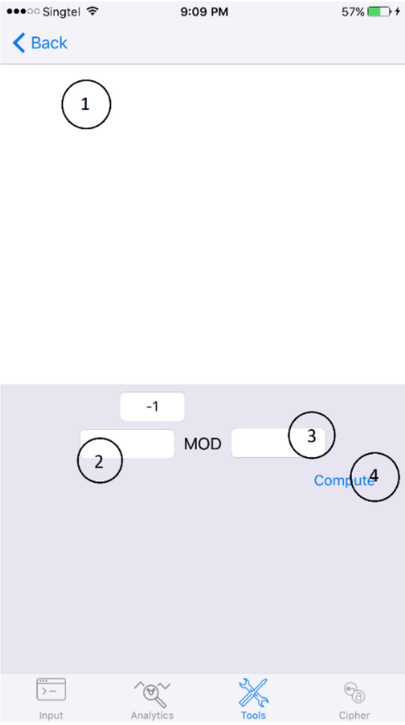
1. **Output Text Field**
   - All output from calculation is displayed here

2. **First Number**
   - First Number that the GCD want to be found.

3. **Second Number**
   - Second Number that the GCD want to be found.

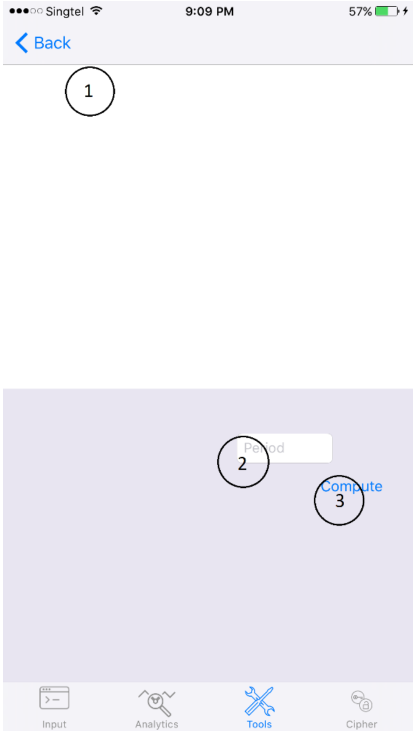
4. **Generate Button**
   - Button to calculate GCD

3. Inverse Modulo



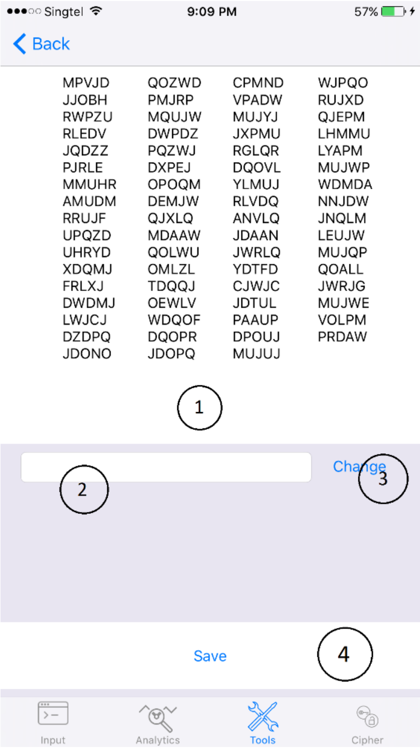
1. **Output Text Field**
   - All output from calculation is displayed here
2. **Dividend Modulus**
   - Number to be divided by divisor. Can be a very big number
3. **Divisor Modulus**
   - Number that divide modulus dividend.
4. **Generate Button**
   - Button to calculate Inverse Mod

4. IC



1. **Output Text Field**
   - All output from calculation is displayed here
2. **Period Number Field**
   - Period (or key length) that user want to guess.
3. **Compute Button**
   - Button to calculate Index of Coincidence

5. *Transposition Tool*



```
MPVJD   QOZWD   CPMND   WJPQO
JJOBH   PMJRP   VPADW   RUJXD
RWPZU   MQUJW   MUJYJ   QJEPM
RLEDV   DWPDZ   JXPMU   LHMMU
JQDZZ   PQZWJ   RGLQR   LYAPM
PJRLE   DXPEJ   DQOVL   MUJWP
MMUHR   OPOQM   YLMUJ   WDMDA
AMUDM   DEMJW   RLVDQ   NNJDW
RRUJF   QJXLQ   ANVLQ   JNQLM
UPQZD   MDAAW   JDAAN   LEUJW
UHRYD   QOLWU   JWRLQ   MUJQP
XDQMJ   OMLZL   YDTFD   QOALL
FRLXJ   TDQQJ   CJWJC   JWRJG
DWDMJ   OEWLV   JDTUL   MUJWE
LWJCJ   WDQOF   PAAUP   VOLPM
DZDPQ   DQOPR   DPOUJ   PRDAW
JDONO   JDOPQ   MUJUJ
```

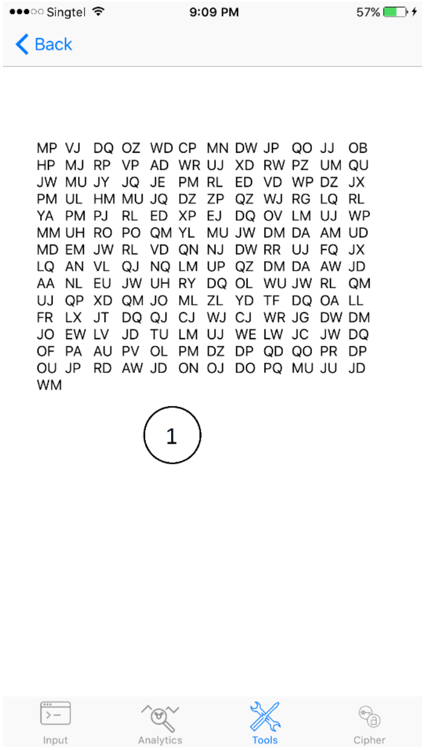
1. **Display Text Field**
    - Display the cipher text in group of 5 letter to make it easier for user while analyse it

2. **Period Number Field**
    - Look for transposition of period number that taken from user input.

3. **Change Button**
    - Button to change text based on the period inputted into the **Period Number Field**

4. **Save Button**
    - Save all change that has been made to real cipher-text

6. *Playfair Tool*

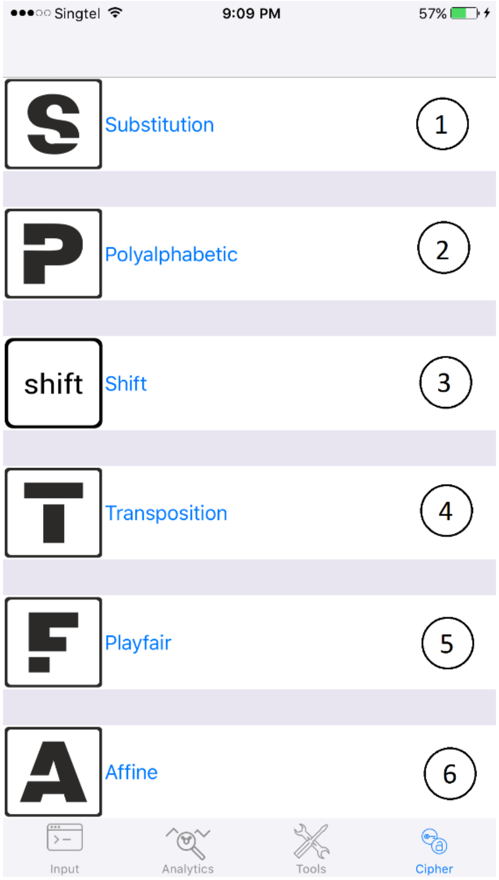


```
MP VJ DQ OZ WD CP MN DW JP QO JJ OB
HP MJ RP VP AD WR UJ XD RW PZ UM QU
JW MU JY JQ JE PM RL ED VD WP DZ JX
PM UL HM MU JQ DZ ZP QZ WJ RG LQ RL
YA PM PJ RL ED XP EJ DQ OV LM UJ WP
MM UH RO PO QM YL MU JW DM DA AM UD
MD EM JW RL VD QN NJ DW RR UJ FQ JX
LQ AN VL QJ NQ LM UP QZ DM DA AW JD
AA NL EU JW UH RY DQ OL WU JW RL QM
UJ QP XD QM JO ML ZL YD TF DQ OA LL
FR LX JT DQ QJ CJ WJ CJ WR JG DW DM
JO EW LV JD TU LM UJ WE LW JC JW DQ
OF PA AU PV OL PM DZ DP QD QO PR DP
OU JP RD AW JD ON OJ DO PQ MU JU JD
WM
```

1. **Display Text Field**
    - Display the cipher text in a group of 2 letter to make it easier for user to analyse it

Fourth Tab

This tab is use to decrypt cryptogram presented in First Tab. Many choice of decryption tools can be chosen by the user. Some tools contain auto-decryption that can find the key automatically for cipher-text.



1. **Substitution Cipher**
   - Substitution Cipher is a method of decrypting plaintext to cipher-text according to fixed system. The change can be letter by letter or word by word.

2. **Polyalphabetic Cipher**
   - Polyalphabetic Cipher is a method of decrypting plaintext to cipher-text according to non-fixed system. It means that one letter can be substituted into multiple possible letters.

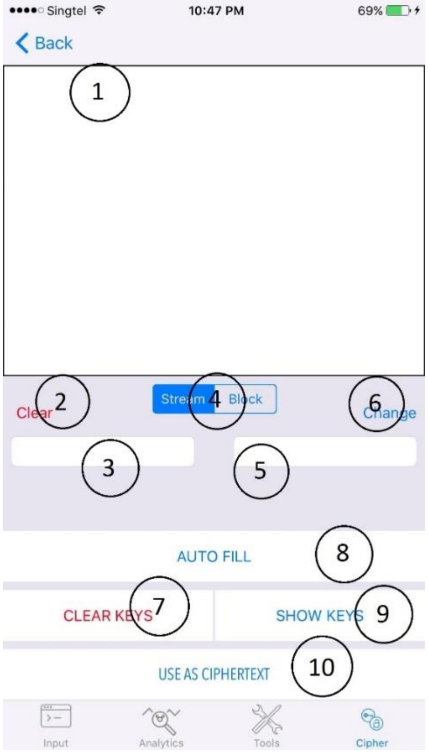
3. **Shift Cipher**
   - Shift Cipher is a method of decrypting by shifting every letter of plaintext by a fixed number.

4. **Transposition Cipher**
   - Transposition Cipher is a method of decrypting based on shifting position of letter according to a regular system. The plaintext is still same but it is in disorganized condition.

5. **Playfair Cipher**
   - Playfair cipher is a method of decrypting based on substitute the pair of cipher text into plaintext based on the table provided from the key

6. **Affine Cipher**
   - Affine cipher is a method of decrypting where all letter is encoded into its numerical form, encrypted using simple mathematical operation (Addition and Multiplication) and converted back to its alphabet form.

1. *Substitution Cipher*



1. **Output Text Field**
   - Cipher-text from **First Tab** is displayed here. Any change that user do also changed directly in this box.

2. **Clear Button**
   - Clear all text input in **First Input** and **Second Input**

3. **First Input**
   - Box contains what letter that the user wants to change.

4. **Stream/Block Segmentation**
   - Allow user to change from stream decryption into block decryption or vice versa

5. **Second Input**
   - Box contains what letter that the **First Input** want to change into.

6. **Change Button**
   - Commit button to change letter.

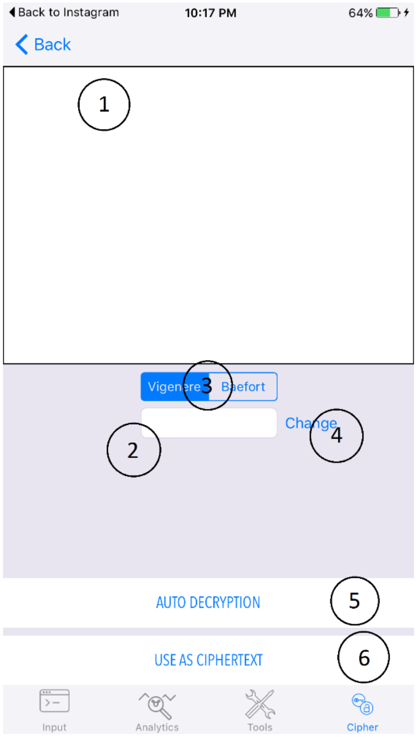
7. **Clear Keys**
   - Clear all change made by the user.

8. **Auto Fill**
   - Transform word (or letter) in **First Input** into its 26-length key equivalent.

9. **Show Keys**
   - Show all currently changed letter in the cipher-text

10. **Use as Cipher-text**
    - Commit all change made to Cipher-text

2. Polyalphabetic Cipher



1. **Output Text Field**
   - Cipher-text from **First Tab** is displayed here. Any change that user do also changed directly in this box.

2. **Key Input Box**
   - Box that contains key specified by user.

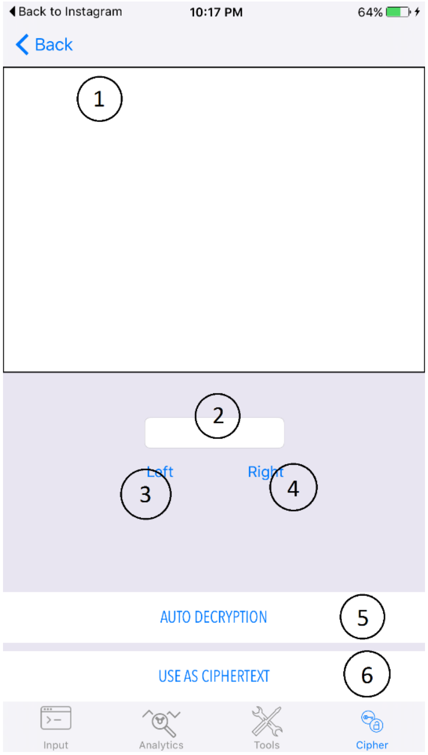
3. **Vigenere/Beaufort Segmentation**
   - Allow user to change from Vigenere type of decryption into Beaufort type or vice versa

4. **Change Input Box**
   - Commit button to apply the key.

5. **Auto Decryption**
   - Allow the cryptanalysis key to find the key automatically and show it to the user.

6. **Use as Cipher-text**
   - Commit all change made to Cipher-text

3. Shift Cipher



1. **Output Text Field**
   - Cipher-text from **First Tab** is displayed here. Any change that user do also changed directly in this box.

2. **Key Input Field**
   - Box that contains key specified by user.

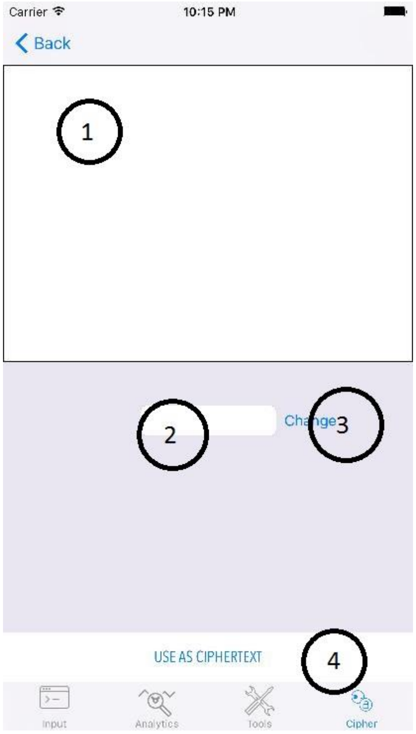
3. **Left Button**
   - Apply Shift Left decryption using key specified in **Key Input Field**

4. **Right Button**
   - Apply Shift Right decryption using key specified in **Key Input Field**

5. **Auto Decryption**
   - Allow Cryptanalysis tools to perform auto decryption that finds the key automatically and display it to the user.

6. **Use as Cipher-text**
   - Commit all change made to the cipher-text

4. Transposition Cipher



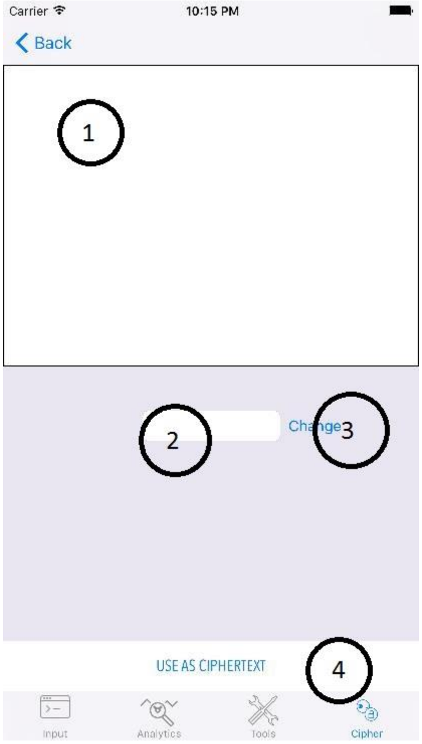
1. **Output Text Field**
   - Cipher-text from **First Tab** is displayed here. Any change that user do also changed directly in this box.

2. **Key Input Field**
   - Box that contains key specified by user

3. **Change Button**
   - Apply decryption using key specified in **Key Input Field**

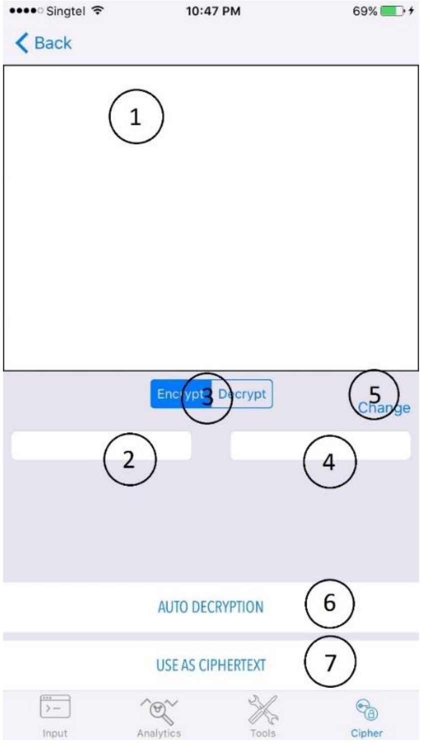
6. **Use as Cipher-text**
   - Commit all change made to the cipher-text

5. Playfair Cipher



1. **Output Text Field**
   - Cipher-text from **First Tab** is displayed here. Any change that user do also changed directly in this box.
2. **Key Input Field**
   - Box that contains key specified by user
3. **Change Button**
   - Apply decryption using key specified in **Key Input Field**
6. **Use as Cipher-text**
   - Commit all change made to the cipher-text

6. Affine Cipher



1. **Output Text Field**
   - Cipher-text from **First Tab** is displayed here. Any change that user do also changed directly in this box.

2. **Alpha Key (Addition)**
   - Box that contains alpha key specified by user

3. **Encrypt/Decrypt Segmentation**
   - Allow user to change from Encryption to Decryption method or vice versa

4. **Beta Key (Multiplication)**
   - Box that contains beta key specified by user

5. **Change Button**
   - Apply decryption using key specified in **Alpha Key** and **Beta Key** field

6. **Auto Decryption**
   - Allow Cryptanalysis tools to perform Auto decryption to find the key automatically and display it to user.

7. **Use as Cipher-text**
   - Commit all change made to the cipher-text

Test Case

System Test Case	
Test ID	1.1
Title	Substitution Decryption
Description	Decrypt the cipher text given the key using the Substitution cipher to produce a valid plaintext

No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Valid cipher text without any space, symbol or number and a key to decrypt it using the Stream type cipher	The cipher text decrypted into a valid plaintext with no error	As per expected	Pass
2	Invalid cipher text with spaces, symbols and numbers on it, and also a key to decrypt it using the Stream type cipher	The cipher text decrypted into a valid plaintext, ignore the symbols, spaces and number on it with no error	As per expected	Pass
3	Valid cipher text without any space, symbol or number and a key to decrypt it using the Block type cipher	The cipher text decrypted into a valid plaintext with no error	As per expected	Pass
4	Invalid cipher text with spaces, symbols and numbers on it, and also a key to decrypt it using the Block type cipher	The cipher text decrypted into a valid plaintext, ignore the symbols, spaces and number on it with no error	As per expected	Pass

System Test Case	
Test ID	1.2
Title	Polyalphabetic Decryption
Description	Decrypt the cipher text given the key using the Polyalphabetic cipher to produce a valid plaintext

No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Valid cipher text without any space, symbol or number and a key to decrypt it using the Vigenere type cipher	The cipher text decrypted into a valid plaintext with no error	As per expected	Pass
2	Invalid cipher text with spaces, symbols and numbers on it, and also a key to decrypt it using the Vigenere type cipher	The cipher text decrypted into a valid plaintext, ignore the symbols, spaces and number on it with no error	As per expected	Pass
3	Valid cipher text without any space, symbol or number and a key to decrypt it using the Beaufort type cipher	The cipher text decrypted into a valid plaintext with no error	As per expected	Pass
4	Invalid cipher text with spaces, symbols and numbers on it, and also a key to decrypt it using the Beaufort type cipher	The cipher text decrypted into a valid plaintext, ignore the symbols, spaces and number on it with no error	As per expected	Pass
5	Auto decrypt the cipher text for Vigenere type cipher	The cipher text decrypted into a valid plaintext with no error	As per expected	Pass
6	Auto decrypt the cipher text for Beaufort type cipher	The cipher text decrypted into a valid plaintext with no error	As per expected	Pass

<table>
<tr><td colspan="5"><b>System Test Case</b></td></tr>
<tr><td colspan="2"><b>Test ID</b></td><td colspan="3">1.3</td></tr>
<tr><td colspan="2"><b>Title</b></td><td colspan="3">Shift Decryption</td></tr>
<tr><td colspan="2"><b>Description</b></td><td colspan="3">Decrypt the cipher text given the key using the Shift cipher to produce a valid plaintext</td></tr>
<tr><td><b>No</b></td><td><b>Test input</b></td><td><b>Expected Result</b></td><td><b>Actual Result</b></td><td><b>Status (Pass/Fail)</b></td></tr>
<tr><td><b>1</b></td><td>Valid cipher text without any space, symbol or number and a key to decrypt it using the Shift Left type cipher</td><td>The cipher text decrypted into a valid plaintext with no error</td><td>As per expected</td><td>Pass</td></tr>
<tr><td><b>2</b></td><td>Invalid cipher text with spaces, symbols and numbers on it, and also a key to decrypt it using the Shift Left type cipher</td><td>The cipher text decrypted into a valid plaintext, ignore the symbols, spaces and number on it with no error</td><td>As per expected</td><td>Pass</td></tr>
<tr><td><b>3</b></td><td>Valid cipher text without any space, symbol or number and a key to decrypt it using the Shift Right type cipher</td><td>The cipher text decrypted into a valid plaintext with no error</td><td>As per expected</td><td>Pass</td></tr>
<tr><td><b>4</b></td><td>Invalid cipher text with spaces, symbols and numbers on it, and also a key to decrypt it using the Shift Right type cipher</td><td>The cipher text decrypted into a valid plaintext, ignore the symbols, spaces and number on it with no error</td><td>As per expected</td><td>Pass</td></tr>
<tr><td><b>5</b></td><td>Auto decrypt the cipher text for Shift Left type cipher</td><td>The cipher text decrypted into a valid plaintext with no error</td><td>As per expected</td><td>Pass</td></tr>
<tr><td><b>6</b></td><td>Auto decrypt the cipher text for Shift Right type cipher</td><td>The cipher text decrypted into a valid plaintext with no error</td><td>As per expected</td><td>Pass</td></tr>
</table>

System Test Case	
Test ID	1.4
Title	Transposition Decryption
Description	Decrypt the cipher text given the key using the Transposition cipher to produce a valid plaintext

No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Valid cipher text without any space, symbol or number and a key to decrypt it using the Transposition type cipher	The cipher text decrypted into a valid plaintext with no error	As per expected	Pass
2	Invalid cipher text with spaces, symbols and numbers on it, and also a key to decrypt it using the Transposition type cipher	The cipher text decrypted into a valid plaintext, delete the symbols, spaces and number on it with no error	As per expected	Pass

System Test Case	
Test ID	1.5
Title	Playfair Decryption
Description	Decrypt the cipher text given the key using the Playfair cipher to produce a valid plaintext

No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Valid cipher text without any space, symbol or number and a key to decrypt it using the Playfair type cipher	The cipher text decrypted into a valid plaintext with no error	As per expected	Pass
2	Invalid cipher text with spaces, symbols and numbers on it, and also a key to decrypt it using the Playfair type cipher	The cipher text decrypted into a valid plaintext, delete the symbols, spaces and number on it with no error	As per expected	Pass

<table>
<tr><td colspan="5"><b>System Test Case</b></td></tr>
<tr><td colspan="2"><b>Test ID</b></td><td colspan="3">1.6</td></tr>
<tr><td colspan="2"><b>Title</b></td><td colspan="3">Affine Decryption</td></tr>
<tr><td colspan="2"><b>Description</b></td><td colspan="3">Decrypt the cipher text given the key using the Affine cipher to produce a valid plaintext</td></tr>
<tr><td><b>No</b></td><td><b>Test input</b></td><td><b>Expected Result</b></td><td><b>Actual Result</b></td><td><b>Status (Pass/Fail)</b></td></tr>
<tr><td><b>1</b></td><td>Valid cipher text without any space, symbol or number and a key to decrypt it using the Affine type cipher</td><td>The cipher text decrypted into a valid plaintext with no error</td><td>As per expected</td><td>Pass</td></tr>
<tr><td><b>2</b></td><td>Invalid cipher text with spaces, symbols and numbers on it, and also a key to decrypt it using the Affine type cipher</td><td>The cipher text decrypted into a valid plaintext, ignore the symbols, spaces and number on it with no error</td><td>As per expected</td><td>Pass</td></tr>
<tr><td><b>3</b></td><td>Auto decrypt the cipher text for Affine type cipher</td><td>The cipher text decrypted into a valid plaintext with no error</td><td>As per expected</td><td>Pass</td></tr>
</table>

System Test Case				
Test ID	2.1			
Title	Fast Exponential Calculator			
Description	Calculate A^B mod C quickly without doing too much complex calculation			
No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	A valid number for each of dividend modulus, power, and divisor modulus	The calculator generates a result of Fast exponential calculation from the given number	As per expected	Pass
2	An invalid number for each of dividend modulus, power, and divisor modulus such that the number is too big to calculate	Show an alert box to inform the user that the number is too big	As per expected	Pass

System Test Case				
Test ID	2.2			
Title	GCD Calculator			
Description	GCD of A and B is the largest positive integer that divided both A and B without the remainder used to determine whether A and B is co-prime or not			
No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	A valid number for each of first number and second number	The calculator generates a result of GCD calculation from the given number	As per expected	Pass
2	An invalid number for each of first number and second number such that the number is too big to calculate	Show an alert box to inform the user that the number is too big	As per expected	Pass

System Test Case				
Test ID	2.3			
Title	Inverse Modulo Calculator			
Description	Inverse Mod (or Extended Euclidean Algorithm) of a modulo m is integer x such that a * x mod m is equivalent to 1 mod m used to determine inverse of key in modern cryptography.			
No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	A valid number for each of dividend modulus and divisor modulus	The calculator generates a result of Inverse Modulo calculation from the given number	As per expected	Pass
2	An invalid number for each of dividend modulus and divisor modulus such that the number is too big to calculate	Show an alert box to inform the user that the number is too big	As per expected	Pass

System Test Case				
Test ID	2.4			
Title	IC Calculator			
Description	IC is a ratio of the total or normalized by dividing by the expected amount for a random source model used to determine the length of key of statistical-based cipher (Vigenere or Beaufort)			
No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	A valid number for period number	The calculator generates a result of IC calculation from the given number	As per expected	Pass
2	An invalid number for period number such that the number is bigger than the length of the cipher text	Show an alert box to inform the user that the number is too big	As per expected	Pass

System Test Case				
Test ID	3.1			
Title	Transposition Tool			
Description	This is tool to help user to decrypt Transposition Cipher. This will show 'pre-decrypted' phase of a cipher-text			
No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	A valid number for period number	The tool change the cipher text based on the given period number	As per expected	Pass
2	An invalid number for period number such that the number is bigger than the length of the cipher text	Show an alert box to inform the user that the number is too big	As per expected	Pass

System Test Case				
Test ID	3.2			
Title	Frequency Graph Tool			
Description	Showed the graph of frequency for user and also Information about n-gram			
No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	A text contains some upper-case, lower-case characters and symbols for case-sensitive option	The tool changes the graph and differentiate the upper and lower-case characters	As per expected	Pass
2	A text contains some upper-case, lower-case characters and symbols for hide all symbol option	The tool changes the graph and exclude all the symbols on the text	As per expected	Pass

System Test Case	
Test ID	4
Title	Input Test
Description	Some test on the input from the user

No	**Test input**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	A very long text inputed by the user	The application limit how long the inputted text can be and make the user cannot add the text anymore	As per expected	Pass
2	An empty text, means there is no a single string inside the input box	The application do not crash upon receiving empty input	As per expected	Pass

Appendix

Not applicable