

SAS-STE

(Secure Text Encryption)

Official Documentation

SAS

Updated: 1/9/2023

Contents

Contents.....	2
Introduction To SAS-STE.....	3
Keys.....	3
Static Keys	3
Dynamic Keys	4
The Algorithm.....	5
Layers.....	6
Supported Characters.....	7
Security.....	8
User side security improvements	8
SAS-STE Releases.....	9
SAS-STE-Wizard	9
Setting Up SAS-STE-Wizard [Windows].....	9
Setting Up SAS-STE-Wizard [Linux]	10
Options in SAS-STE-Wizard	10
Function-Key option:.....	11
Input Method option:	12
Output Method option:.....	13
Usage Sample	15
SAS-STE-Mobile.....	16
Setting Up SAS-STE-Mobile [Android]	16
Pages In SAS-STE-Mobile.....	17
Main page:	17
Key Generator page:.....	17
Input page:	18
Input Key page:.....	19
Output page:	19
Static Key Generator page:.....	20
Static Key Display page:	20
File Selector page:.....	20
Text Recognizer page:	21
Enlarge page:.....	22

Introduction To SAS-STE

SAS-STE: SAS-Secure Text Encryption is a very secure and unique opensource text encryption algorithm developed and maintained by Saaiq Abdulla Saeed (SAS). The algorithm uses high level characters to convert plain text into cypher text instead of encrypting at binary level. More specifically it substitutes a plain text character with another character at a much more advanced and randomized manner (not like the 'Caesar cipher' which shifts characters along the alphabet based on single a number acting as a key). Furthermore the encryption can be utilized to encrypt text within text based files (e.g. .txt , .java, .py, .html, etc.). This utilization is done in all the official releases of SAS-STE allowing users to encrypt both text-based files and plain text.

This encryption algorithm uses two types of keys, Static Keys and Dynamic Keys (previously referred to as 'Public Keys'). Static Keys are keys which are made to be unique to each copy of a SAS-STE release while Dynamic Keys are meant to be used as general keys which is to be unique for each encryption done. These keys are fixed format keys which can be generated using an SAS-STE release (Do note that SAS-STE allows users to re-use previously generated keys). Though these keys are fixed format, the number of unique keys that can be generated are quite high and the chances of duplicate keys are low. This will be proved once you get an understanding of the algorithm and how it uses the two types of keys, which is explained in the upcoming sections 'Keys' and 'The Algorithm'.

Keys

As mentioned before SAS-STE uses two types of keys, Static Keys and Dynamic Keys. Below are the details of both the keys.

Static Keys

Static Keys are a set of keys which are meant to be unique per SAS-STE release (By default all releases will have the same 'default' Static Keys) and can be changed within a release. A Static Key itself is an array of scrambled numbers of fixed length (95 – array length is equal to the number of supported characters mentioned in the 'Supported Characters' section) which is used as a reference when encrypting plain text using a Dynamic Key (How it is referred will be explained in the 'The Algorithm' section).

In the current version of SAS-STE (v4) there are a total of four Static Keys, each used by a separate layer of the encryption algorithm. All of these Static Keys are stored in a single file named 'StaticKeys.stkey'. Every release will contain a Static Keys file with the default Static Keys, and on changing the Static Keys, this file will be overridden. Though this file will be overridden by newly generated Static Keys, SAS-STE releases provides the function to restore defaults. If you are to take a backup of Static Keys, this 'StaticKeys.stkey' file is the file you want to take. Then the Static Keys file of another release can be replaced with the backup in order to process data

using the backed up Static Keys (Some releases such as the SAS-STE-Mobile has an inbuilt feature to replace Static Keys with the backed up Static Keys without having to do it manually).

Dynamic Keys

Dynamic Keys are the keys which are to be used generally and varied upon per encryption. Like Static Keys, Dynamic Keys are also fixed format keys. Dynamic Keys are currently 112 characters long (896 bits). There are four parts to the Dynamic Key. (Fig.1.1 is a sample Dynamic Key. More specifically the hardcoded basic Dynamic Key in SAS-STE releases that will be used when processing data using the 'No Key' function, where the only variation in your encrypted data is of which it gets from unique Static Keys).

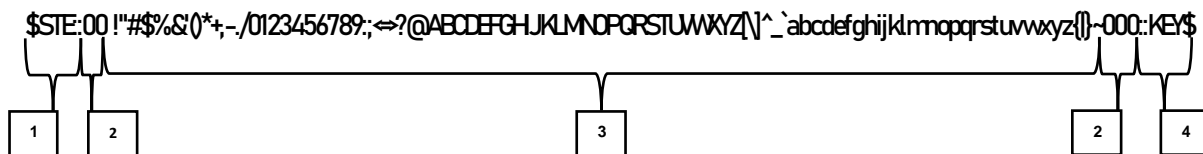


Fig.1.1

1- Header: These six characters at the beginning of the key can be changed to any character. However, the length should be same.

2- Function Integers: These are randomly generated integers which are used to decide which functions to use at certain layers where there are a number of functions available within the algorithm, i.e., how many times should the plain text be processed at layer1 or where in the plain text should the extra random characters (salting) be added.

3- Character Set: This is a randomly scrambled character set with the length of 95 characters (Equal to the number of supported characters mentioned in section 'Supported Characters') which will be used in processing data while referring to the Static Keys.

4- Footer: Like the header these six characters at the end of the key can also be changed to any character while maintaining its fixed length.

Do also note that whenever 'Key' is mentioned in a SAS-STE release, it is referring to Dynamic Keys and not Static Keys (Static Keys will be referred to as it is).

The Algorithm

Before attempting to read this section, it is advised to have covered the 'Keys' section. To recall the 'Keys' section, there are two types of keys in SAS-STE, Static Keys and Dynamic Keys. Static Keys are an array of scrambled up numbers while Dynamic Keys are a fixed length key contain multiple parts, most importantly a *character set* - KEY_CHAR_SET which will be converted into an array by the algorithm.

As for how the SAS-STE algorithm works at a basic level, when a plain text is inputted, the plain text will go through one of the many salting functions based on one of the function integers in the Dynamic Key, and after that the salted text will be divided into a *character array* - CHAR_ARR. Then for *each character* – ECHAR of CHAR_ARR, its place value in the KEY_CHAR_SET will be noted - PLACE_VALUE. Then the *integer* in PLACE_VALUE of the *Static Keys array* will be noted - STKEY_INT. After that the ECHAR *character* will be substituted with the *character* at the STKEY_INT place value of the KEY_CHAR_SET encrypting the ECHAR *character*. This process is repeated for each character of the input plain text until all are encrypted. Below is an example of encryption via the SAS-STE algorithm.

Plain Text: abc

Salted Text: tabjc

Salted Text Array: CHAR_ARR[5] = t, a, b, j, c

Dynamic Key Character Set: KEY_CHAR_SET[10] = o, b, t, p, j, c, q, a, f, z

Static Keys Array: STKEY_ARR[10] = 5, 2, 8, 0, 1, 7, 4, 9, 3, 6

1. The first character in the Salted Text Array is 't'. CHAR_ARR[0] is 't'
2. 't' is at the place value 2 of the Dynamic Key Character Set. KEY_CHAR_SET[2] is 't'
3. Now in the Static Keys Array's place value 2 there is the integer 8. STKEY_ARR[2] is 8
4. The place value 8 of the Dynamic Key Character Set is the character 'f'.
KEY_CHAR_ARR[8] is 'f'
5. Now when outputting the salt text 't' will be changed to cipher text 'f'.
6. These processes from 1-5 will be repeated for all the characters in the Salted Text Array.

Input:	a b c
CHAR_ARR character:	t a b j c
KEY_CHAR_SET place value:	2 7 1 4 5
STKEY_ARR integer:	8 9 2 1 7
Output:	f z t b a

Layers

There are a total of three main layers in the SAS-STE algorithm. From which one layer (layer2) is divided in to three parts. Below are the details of each one of these layers.

Layer0: Layer0 is the layer responsible for all the salting processes. The main salting process carried out in this layer is adding two extra random characters into a random place within every three characters of the input plain text based on a function integer in the Dynamic Key. In other words, every three characters are changed to five. In addition to this main function, the input plain text (specifically half-salted text) will also go through a few more salting functions.

As current SAS-STE releases allows to automatically re-encrypt the input data up to 10 times, keep in mind that every time the data is encrypted, its size is going to increase at the rate of two more characters for each three. For instance, when encrypting twice, three characters will be increased to about six to eight characters depending on function integers.

Layer1: Layer1 will take the output of Layer0 as an input and process the whole input through the algorithm explained before using the Static Key assigned to Layer1 out of the four Static Keys. The output of this layer will be then again taken as an input and re encrypted random number of times depending on a randomly generated function integer in the Dynamic Key.

Layer2: Layer2 is divided into three different parts, each will process its piece of input like Layer1 using the Static Key assigned to each part. This layer will take the output of Layer1 as input and split the input into three pieces and randomly give each piece as an input to a single part of Layer2 depending on a randomly generated function integer in the Dynamic Key. Like Layer1, this layer will also take its output as an input and re encrypt it multiple times.

Once the inputted plain text has passed through all layers, the final output (output of Layer2) will be taken as an input (input of Layer0) and re encrypted through all layers while being re encrypted at specific layers such as Layer1 and Layer2. This full re encryption is also done multiple times based on the 'Encrypt Amount' number the user has manually selected while generating the Dynamic Key.

Supported Characters

As the SAS-STE algorithm processes on high level characters, there are limitations to the number of characters the algorithm can process. Currently by default, SAS-STE supports 95 (including SPACE) ASCII characters. Any unsupported character will only pass through Layer0 and will not go through Layer1 or Layer2. Though the algorithm can handle unsupported characters while processing plain text, it is not recommended to encrypt plain text with a large number of unsupported characters. However, as SAS-STE is an opensource algorithm, one can always customize it and create private variants which can support other characters such as characters of other languages. Below is a list of all the currently supported characters by SAS-STE:

0.	13. -	26. :	39. G	52. T	65. a	78. n	91. {
1. !	14. .	27. ;	40. H	53. U	66. b	79. o	92.
2. "	15. /	28. <	41. I	54. V	67. c	80. p	93. }
2. #	16. 0	39. =	42. J	55. W	68. d	81. q	94. ~
4. \$	17. 1	30. >	43. K	56. X	69. e	82. r	
5. %	18. 2	31. ?	44. L	57. Y	70. f	83. s	
6. &	19. 3	32. @	45. M	58. Z	71. g	84. t	
7. '	20. 4	33. A	46. N	59. [72. h	85. u	
8. (21. 5	34. B	47. O	60. \	73. i	86. v	
9.)	22. 6	35. C	48. P	61.]	74. j	87. w	
10. *	23. 7	36. D	49. Q	62. ^	75. k	88. x	
11. +	24. 8	37. E	50. R	63. _	76. l	89. y	
12. ,	25. 9	38. F	51. S	64. `	77. m	90. z	

Security

SAS-STE is a very secure encryption algorithm due to many unique reasons. First of all, one of the more straightforward reasons is due the large variation of key combinations (Static Keys and Dynamic Key combinations). A different output will be given for each of the possible Dynamic Keys even with the same Static Keys. Likewise, a different output will also be given for all the possible Static Keys using the same Dynamic Key. Therefore, the total number of possible unique key combinations can be given as the product of the total number of Static Keys and the total number of Dynamic Keys. Do keep in mind that there are four Static Keys which are also unique. Moreover, whenever a function integer changes in a Dynamic Key, all the possible character set combinations also become unique.

SAS-STE also does not allow a way to detect wrong Dynamic Keys. So, if a wrong Dynamic Key is inputted to decrypt an encrypted data, the data will go through decryption process without showing any errors. However, in the end, the final output will not be the actual plain text that was encrypted (Due to this reason, SAS-STE has multiple output modes for different types of inputs, where some would prevent the override of the actual input in case the encrypted data becomes scrambled up to an unrecoverable form when accidentally using the wrong keys). This feature is beneficial in terms of security as it would make it exceptionally difficult to brute force an SAS-STE encrypted data.

User side security improvements

While SAS-STE itself is secure, it is up to you and your creativity to make your data even more secure while using SAS-STE. In other words, think of SAS-STE releases as a simple tool. Your ability to utilize this tool is what will provide exceptionally powerful security. Below are some of the actions you can take to make the best use of SAS-STE releases.

Changing Static Keys: This is rather important as changing Static Keys is fundamental for the maximum security through out of the box functions of SAS-STE.

Multiple key encryption: While SAS-STE has the inbuilt function to encrypt the data over and over multiple times via the same set of keys, why not manually re encrypt your encrypted data with another key or set of keys. That way the chances of somehow cracking your encrypted data without the actually keys are very nearly impossible.

Changing the footer and header of Dynamic Keys: Though this will not impact the processing of data, this can prove to be useful when storing or transferring keys. You can change the header and footer into some random characters making it look like a piece of encrypted cipher text. This then you can even embed within some actual encrypted cipher text making it harder to notice (When doing this make sure you embed the key in a new line within the encrypted cipher text to avoid data being scrambled up on decrypt).

Creating SAS-STE variants: As SAS-STE releases are opensource, you can customize the algorithm, keys, character sets, etc. to create your own private variant of SAS-STE if you have the skills required. This would ensure better security to your encrypted data as you will be the only one with the new variant SAS-STE algorithm (unless you redistribute it as a separate release or by embedding it within another software. In any case remember to kindly give credit to the original developer, Saaq Abdulla Saeed for the base algorithm of SAS-STE).

Above are some of the actions you can take to ensure maximum security to your data. The rest is up to your creativity and skill.

SAS-STE Releases

SAS-STE releases are encryption software which makes use of the SAS-STE algorithm to encrypt data. Currently there are two main official releases and an online tool (Not considered as an official release). The two official releases are SAS-STE-Wizard, which is a portable step-by-step command line wizard available for Linux and Windows and, SAS-STE-Mobile, a mobile encryption application available for Android.

SAS-STE-Wizard

SAS-STE-Wizard is a portable step-by-step command line wizard released for Linux and Windows. This wizard does not require any installations, however is dependent on the Java Runtime Environment (Java JRE). Therefore, in order to use SAS-STE-Wizard, Java JRE should be present in a system (Installation of Java JRE will be explained below, under the 'Setting Up SAS-STE-Wizard' sub sections). The minimum Java JRE version required is Java JRE 17.

Setting Up SAS-STE-Wizard [Windows]

1. Go to saaiqSAS Official Website (saaqsas.github.io), scroll down and under 'Projects', select SAS-STE. Then in the SAS-STE page, scroll down and download SAS-STE-Wizard for Windows under 'Releases'.
2. Then extract the 'SAS-STE-Wizard-Windows_vX.X.X.zip' file downloaded, and go into the extracted folder.

3. Now run 'the sas-ste.bat' file to start SAS-STE-Wizard. If you are prompt with a 'Windows protected your PC' box, click 'More info' and then click 'Run anyway' (Don't worry, no malicious processes will be present in SAS-STE-Wizard).

Setting Up SAS-STE-Wizard [Linux]

1. Go to saaqsas.github.io, scroll down and under 'Projects', select SAS-STE. Then in the SAS-STE page, scroll down and download SAS-STE-Wizard for Linux under 'Releases'.
2. Now extract the downloaded 'SAS-STE-Wizard-Linux_vX.X.X.zip' file. This can be done using a graphical interface or by using the 'unzip' command. For the 'unzip' command, follow the steps below.

If you don't have 'unzip', then type the following command to install it via APT.

```
> sudo apt update
> sudo apt install unzip
```

Now use the following syntax to extract the '.zip' file.

```
> unzip [.zip file]
```

3. Once you have extracted the downloaded .zip file, go to the extracted directory.
4. Now give execute permission to the 'sas-ste' file using the following command.

```
> sudo chmod +x sas-ste
```

6. Now execute the sas-ste file using the following command to start SAS-STE-Wizard.

```
> ./sas-ste
```

Options in SAS-STE-Wizard

As SAS-STE-Wizard is a wizard, you will be continuously presented with options or choices to choose from until the final output. When you are presented with such a set of options, you need to enter the number or the character within the square brackets '[]'. Incases where you are to enter an input, you have to enter the type of input that is being asked to input (e.g., file path, text, etc.).

There are three main set of options, which is 'Function-Key option', 'Input Method option' and 'Output Method option'. In every run of SAS-STE-Wizard, you will go through all these three main

sets of options. In addition to these three, there will be additional set of options which takes further detailed choices. Below, the details of each one of the main sets of options are provided.

Function-Key option:

When you first start SAS-STE-Wizard you will be greeted with a banner followed the main root set of options which is 'Select A Function-Key' options. In this set of options, you will be selecting your function (Encrypt or Decrypt) combined with your key related option.

```
Select A Function-Key (Select Between [1]-[5]):
```

```
[1] Encrypt-No Key
```

```
[2] Encrypt-New Key
```

```
[3] Encrypt-Existing Key
```

```
[4] Decrypt-No Key
```

```
[5] Decrypt-Keyed
```

```
[?] Help
```

```
[x] Exit
```

```
>
```

1 - Encrypt data without a Dynamic key (uses a hardcoded fixed basic Dynamic Key without the user having to manually deal with it) using the protection from the Static Keys. On selecting this option, you will directly go to the 'Input Method option'.

2 - Encrypt data after generating a new Dynamic Key. On selecting this, a new message will be printed asking you to input the number of times you want to encrypt data (Encrypt Amount). Here you can input any number between one to ten (Higher the number, the more the size of input data would increase as mentioned in the 'Security' section). After entering a number, a Dynamic Key will be generated and you will be asked whether you accept the key or not. If no, the Dynamic Key will be re-generated ask the same message will be showed again. If yes, then you will go to the 'Input Method option'.

3 - Encrypt data using an already existing (previously generated) Dynamic Key. On selecting this option, you will be asked to enter a Dynamic Key. Once you have entered it then you will go to the 'Input Method option'.

4 - Decrypt encrypted data that was encrypted using the 'Encrypt-No key' option. On selecting this option, you will directly go to the 'Input Method option'.

5 - Decrypt encrypted data using the Dynamic key that was used to encrypt it. On selecting this option, you will be asked to input a Dynamic Key. This key will be used to decrypt the input data. After entering a valid Dynamic Key, you will go to the 'Input Method option'.

? - Prints a brief help menu.

x/X - Exits SAS-STE-Wizard.

99 - Goes to the Static Keys generator to change Static Keys. On selecting this additional option, you will be asked to enter an 'Identifier Name' which will be used to identify Static Keys files. Once you have given an 'Identifier Name', a full set of Static Keys will be printed and you will be asked whether you accept them or not. If you choose no, then the keys would be re-generated and the same message will be printed. If you choose yes, then the generated Static Keys will be applied and the SAS-STE-Wizard will be exited. Now if you re-open SAS-STE-Wizard you can see the 'Identifier Name' you entered under the banner after '[?] Static Keys:'. This means that the new Static Keys are being used.

98 - Restores Static Keys to default.

Input Method option:

This is where you will be choosing the type of input you want to give, such as text and text-based files.

Select An Input Method (Select Between [1]-[3]):

[1] Single Line Text

[2] Multiple Text

[3] Text Based File(s)

[?] Help

[<] Back

[x] Exit

>

1 - Select this option to input a single line of text one single time. On selecting this option, you will have to input your single line text and then you will go to the 'Output Method option'.

2 - Select this option to input a single line of text multiple times. On selecting this option, you will directly go to 'Output Method option'. Once you have finished giving all the output related inputs, a command menu will be printed. Below it you will be asked to input your single line text and then the result will be outputted based on your chosen option in the 'Output Method option'. After that you will once again be asked to enter a single line text to process. This loop will continue until you give the command for ending as an input (Commands with its assigned functions will be displayed in the command menu).

3 - Select this option to process the text within a text-bases file or files. On selecting this option, you will be asked to input the full path (Starting from the root of the filesystem) of the file you want to process. Here you can enter multiple full file paths separated by a ';' (Semi colon – should be in between two paths). Moreover, the full path of a directory can even be inputted here. If this is

done then a directory crawler will go through all the files and directories within that directory path and select all the text-based files within them.

Once you have inputted one or more file or directory paths, the path of all the text-based files which exists within the paths you have provided will be displayed (Empty text-based files will be omitted). Then you will be asked whether you want to omit any of the detected files from being processed. If you want to omit, then you can enter the number assigned to the file you want to remove (E.g., if the file to omit is '[3] someDirectory/text.txt' then type '3' to remove the file). Then all the detected files will be again displayed after removing the file whose assigned number you entered.

Once you have removed all the files you want to remove from being processed or have no file to remove, then to continue on to the next step, you can input any character which is not a number assigned to a file or which is not a command.

? - Prints a brief help menu.

< - Goes back to the previous main set of options.

x/X – Exits SAS-STE-Wizard.

Output Method option:

Here you will be choosing how you want to output the processed data (E.g., print to terminal or save to file). The options available for selection in this set of options will vary depending on the option selected in the 'Input Method option'. Below, details of these options will be provided based on the type of input given for processing.

If single line text or multiple text was given as input:

Select An Output Method (Select Between [1]-[2]):

[1] Print

[2] Save To File

[?] Help

[<] Back

[x] Exit

>

1 – Prints output to the terminal.

2 – Saves output to an existing text-based file or a to a new text-based file. On selecting this option, you will be asked to input the full path of the text-based file to which you want to save the output to. Here you can also enter a full path of a non-existing text-based file which would then be created and the output saved to it. Do not that this option will not override any text-based files but simply add the output just below the last used line in the file.

? - Prints a brief help menu.

< - Goes back to the previous main set of options.

x/X – Exits SAS-STE-Wizard.

If a single text-based file was given as input:

Select An Output Method (Select Between [1]-[2]):

[1] Print

[2] Save To File

[3] Save To Input File

[?] Help

[<] Back

[x] Exit

>

1 – Prints output to the terminal.

2 - Saves output to an existing text-based file or a to a new text-based file. On selecting this option, you will be asked to input the full path of the text-based file to which you want to save the output to. Here you can also enter a full path of a non-existing text-based file which would then be created and the output saved to it. Do not that this option will not override any text-based files but simply add the output just below the last used line in the file.

3 – Overrides the input files with the output.

? - Prints a brief help menu.

< - Goes back to the previous main set of options.

x/X – Exits SAS-STE-Wizard.

If multiple text-based files were given as input:

Select An Output Method (Select Between [1]-[2]):

[1] Print

[2] Save To Input File

[?] Help

[<] Back

[x] Exit

>

- 1 – Prints output to the terminal.
- 2 – Overrides the input files with the output.
- ? - Prints a brief help menu.
- < - Goes back to the previous main set of options.
- x/X – Exits SAS-STE-Wizard.

Usage Sample

Encrypt: [One line text, new key saved to key file, output saved to file]

1. Open SAS-STE-Wizard and select the 'Encrypt-New Key' option.
2. Enter the number of times you want to encrypt the text with your key.
3. Select 'Yes, Save Key To File' to accept the key and save it to a key file.
4. Enter the full path where you want to create the new key file (Enter the path with the file name without any file extension, e.g., "D:\dirOne\dirTwo\keyFile").
5. Then select the 'Single Line Text' in the 'Input Method option', and enter the text you want to encrypt.
6. Now select 'Save To File' option in the 'Output Method option'.
7. Enter the full path where you want to create the file to which the output will be saved (Enter the path with the file name and file extension, e.g., "D:\dirOne\outputFile.txt").
8. Select 'Yes' to create the non-existing file in the path you gave and save the output to it.
9. That's all.

Decrypt [Single file, key from key file, output saved to input file]

1. Open SAS-STE-Wizard and select the 'Decrypt-Keyed' option.
2. Select 'Key File' as a key input method.
3. Enter the full path to the key file which was created before (Enter the path with the key file name and the .skey file extension, e.g., "D:\dirOne\dirTwo\keyFile.skey").
4. Then select 'Text Based File(s)' in the 'Input Method option'.
5. Enter the full path of the file to which the output was save before saved (Enter the path with the file name and file extension, e.g., "D:\dirOne\outputFile.txt").
6. Now press any key other than '1' ('1' is the number assigned to the file. When you enter '1' the file will be omitted from processing).
7. Now select 'Save To Input File' in the 'Output Method option'.

8. That's all. The decrypted text will be in the file you inputted.

SAS-STE-Mobile

SAS-STE-Mobile is the official mobile encryption application for the SAS-STE algorithm which is currently released for only Android. This mobile application is developed in such a way that it would allow users to carry out all the major functions that can be carried out via a desktop release of SAS-STE (E.g., Encrypting text-based files and plain text, generating both Static and Dynamic Keys, saving to file) while taking up as less time as possible with a simple yet straight forward and effective user interface. However, do note that SAS-STE-Mobile is not fit out to carry out too heavy tasks as the application would most often be run on low power mobile devices unlike the desktop releases of SAS-STE.

In addition to the basic functions provided in SAS-STE releases, SAS-STE-Mobile has the feature to scan text from images and camera using an optical character recognition system provided by the Google's Machine Learning Kit (Google ML Kit). Furthermore, the mobile application also provides easy sharing of output text and keys to others via emails, SMS, social media platforms or any other applications which allows similar actions to be carried out.

Setting Up SAS-STE-Mobile [Android]

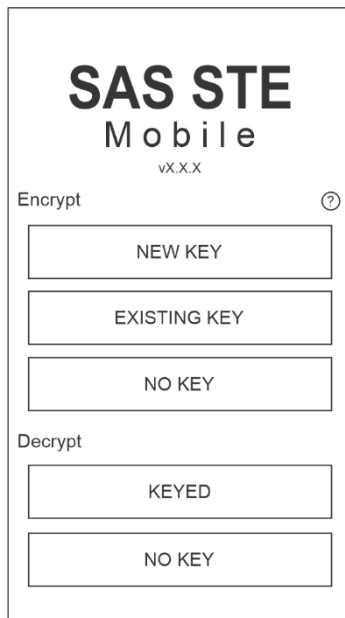
1. Go to saaqsas.github.io, scroll down and under 'Tools', select SAS-STE. Then in the SAS-STE page, scroll down and download SAS-STE-Mobile for Android under 'Latest Releases'.
2. Now using a file explorer, navigate to your device's 'Downloads' directory and select the SAS-STE-Mobile APK file.
3. If you are asked to give permission to the file explorer to install external APKs, then give the permission either via the popup or via settings (This permission is needed by applications which are trying to install other applications outside of Google Play Store).
4. Once installation is complete open up SAS-STE-Mobile.
5. Now once asked, give SAS-STE-Mobile the permission to access all files on your device (This permission is compulsory to read and write text-based files while processing).
6. Once you have given the permission, wait for about 6-10 seconds for the SAS-STE-Mobile to generate its file system in the 'Downloads' directory (This process takes place only during the first time you open the application after installing or if the filesystem has missing files).
7. Then after you see the 'StaticKeys: Default' message, you are ready to use SAS-STE-Mobile.

Pages In SAS-STE-Mobile

SAS-STE-Mobile consists of six main pages, where each page is specialized for a certain type of input or task. In addition to these main pages, additional pages are also present in the application. Below will be the details of these pages.

Main page:

This is the first page you will be greeted with on startup of SAS-STE-Mobile.



SAS STE
Mobile
vX.X.X

Encrypt ?

NEW KEY

EXISTING KEY

NO KEY

Decrypt

KEYED

NO KEY

Encrypt-NEW KEY - Encrypt data after generating a new dynamic key. On selecting this option, you will go to the 'Key Generator' page. On holding and releasing this button three times, you will go to the 'Static Key Generator' page.

Encrypt-EXISTING KEY - Encrypt data using an existing previously generated dynamic key. On selecting this option, you will go to the 'Key Input' page.

Encrypt-NO KEY - Encrypt data without a Dynamic key (uses a hardcoded fixed basic Dynamic Key without the user having to manually deal with it) using the protection from the Static Keys. On selecting this option, you will go to the 'Input' page.

Decrypt-KEYED - Decrypt data that was encrypted via a dynamic key. On selecting this option, you will go to the 'Key Input' page.

Decrypt-NO KEY - Decrypt data that was encrypted via the 'Encrypt-NO KEY' function.

Key Generator page:

This page deals with all the dynamic key generation and saving functions.

Amount of times to Encrypt - Select the number of times you want to re-encrypt the input data (Higher the number, the more the size of input data would increase as mentioned in the ‘Security’ section).

GENERATE KEY - Click this button to generate a Dynamic Key.

Key Display Area - The area where the generated key will be displayed.

SHARE - Click this button to share the Dynamic Key as text via any application which allows such a feature (E.g., SMS, social media applications, email, etc.).

COPY - Click to copy the Dynamic Key into clipboard.

File Name Input Field - Click to enter only a file name for the key file, not the full path (File extension should not be included in the name).

SAVE TO FILE - After giving a key file name, click to save the key to ‘Downloads/SAS-STE/Key/[keyFileName].skey’.

BACK - Click to go to the previous page.

NEXT - Click to go to the next page (Input page).

Input page:

This page deals with all the inputs needed that are to be encrypted or decrypted,

Input Text Field - Click to enter text to process.

SCAN - Click to scan text from existing images or from an image instantly taken from camera and paste the detected text on to the ‘Input Text Field’. On selecting this, you will go to the ‘Text Recognizer’ page.

PASTE - Click to instantly paste whatever is on the clipboard to the ‘Input Text Field’.

File Path Input Field - Click to manually enter the full path of the text-based file you want to process (File extension should be included).

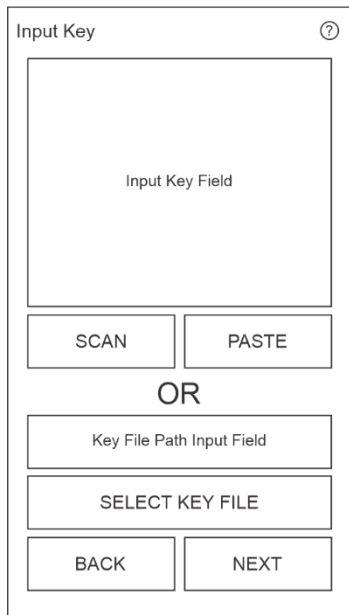
SELECT FILE - Click to select a text-based file to process via an inbuilt file explorer. On selecting this, you will go to the ‘File Selector’ page.

BACK - Click to go to the previous page.

NEXT – Click to go to the next page (Output page).

Input Key page:

In this page you are to give a Dynamic Key either as plain text or as a key file.



The 'Input Key' page features a title bar with a question mark icon. Below it is a large 'Input Key Field' box. Underneath the field are two buttons: 'SCAN' and 'PASTE'. A horizontal separator with the word 'OR' in the center follows. Below this is a 'Key File Path Input Field' box, followed by a 'SELECT KEY FILE' button. At the bottom are two buttons: 'BACK' and 'NEXT'.

Input Key Field – Click to manually enter a Dynamic Key.

SCAN – Click to scan a Dynamic Key from an existing image or after instantly capturing an image from camera, and paste the detected text on to the 'Input Key Field'. On selecting this, you will go to the 'Text Recognizer' page.

PASTE – Click to paste whatever is on the clipboard to the 'Input Key Field'

Key File Path Input Field – Click to manually enter the full path of a key file (File extension should be included).

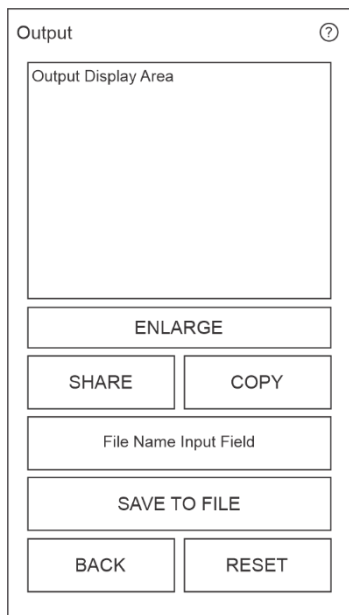
SELECT KEY FILE - Click to select a key file via the inbuilt file explorer. On selecting this, you will go to the 'File Selector' page.

BACK – Click to go to the previous page.

NEXT – Click to go to the next page (Input page).

Output page:

This page displays the processed output and provides the save output to file function.



The 'Output' page features a title bar with a question mark icon. Below it is a large 'Output Display Area' box. Underneath the area is an 'ENLARGE' button. Below that are two buttons: 'SHARE' and 'COPY'. This is followed by a 'File Name Input Field' box, then a 'SAVE TO FILE' button. At the bottom are two buttons: 'BACK' and 'RESET'.

Output Display Area - Displays the processed output as text.

ENLARGE - Enlarges the 'Output Display Area' to full screen. On selecting this, you will be taken to the 'Enlarge' page.

SHARE - Click this button to share the processed output as text via any application which allows to share text (E.g., SMS, social media applications, Emails, etc.).

COPY - Click to copy the processed output to clipboard.

File Name Input Field - Click to enter a file name with extension to which the output will be saved to. For advance users, this field can be manipulated using '../' to go to the parent directory.

SAVE TO FILE - After entering a file name, click this button to save output to 'Downloads/SAS-STE/Output/[fileName]'.

BACK - Click to go to the previous page.

RESET - Resets everything and takes you back to the 'Main' page.

Static Key Generator page:

This page deals with everything related to Static Keys. To access this page, you need to hold and release 'NEW KEY' button in the 'Main' page three times.

SAS-STE: Static Key Generator ?

Static Key Identifier Input Field

GENERATE STATICKEYS

OR

Static Keys File Path Input Field

SELECT FILE CONFIRM

OR

RESTORE DEFAULTS

VIEW CURRENT STATIC KEYS

BACK

Static Key Identifier Input Field - Click to input a name for the new Static Keys to be generated.

GENERATE STATICKEYS - After entering an identifier name, click to generate new Static Keys and view them. On selecting this, you will be taken to the 'Static Keys Display' page.

Static Keys File Path Input Field - Click to enter the full path to an already existing Static Keys file.

SELECT FILE - Click to select an existing Static Keys file via the inbuilt file explorer.

CONFIRM - Click to confirm the full path of Static Keys file in 'Static Keys File Path Input Field'.

RESTORE DEFAULTS - Hold and release this button three times to restore the default Static Keys. Do note that by doing this you will lose the current Static Keys and will not be able to recover it.

VIEW CURRENT STATIC KEYS - Click to display the current Static Keys.

BACK - Click to go to the previous page.

Static Key Display page:

Displays the content of Static Keys file that is either existing or generated.

Static Keys Display Area

BACK APPLY

Static Keys Display Area - Displays the content of Static Keys file here.

APPLY - Hold and release this button three times to confirm the Static Keys displayed in the 'Static Keys Display Area' and apply them to be used while processing data.

BACK - Click to go to the previous page.

File Selector page:

This is a simple inbuilt file explorer allowing to select files with ease.

Files Listing Area

File Path Display/Input Area

SELECT SEARCH

Files Listing Area - Lists all files in a directory including hidden ones. Here you can click directories to display the content within them and click files to select them. After selecting a file click 'SELECT' to confirm it.

File Path Display/Input Area - Displays the full path of the currently selected file or displayed directory. Here you can even manually enter a full path of a directory and click 'SEARCH' to display its content.

SELECT - Click to confirm the file path in the 'File Path Display/Input Area'.

SEARCH - Click to display the contents of a directory whose path is entered in the 'File Path Display/Input Area'.

Text Recognizer page:

This Page is responsible for extracting text from an image provided.

powered by Google's ML Kit

Image Preview Area

Recognized Text Display Area

GALLERY CAMERA

CONFIRM

Image Preview Area - Display the image selected.

Recognized Text Display Area – Displays the text recognized and extracted from the image. Here you can manually edit the recognized text.

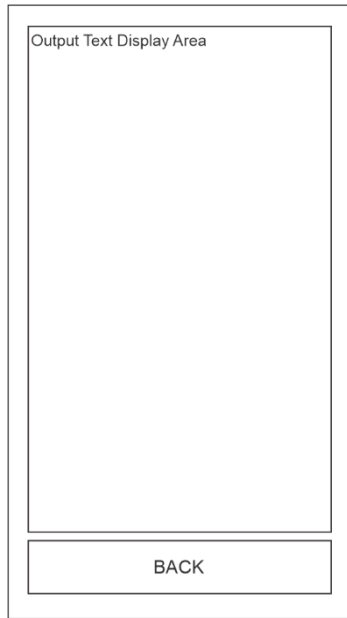
GALLERY - Click to select an existing image from the gallery or elsewhere.

CAMERA - Click to select an image after capturing it from the camera.

CONFIRM - Click to confirm the text in the 'Recognized Text Display Area'. On selecting this, you will go back to the previous page and the recognized text will be pasted in the set text input field of that page.

Enlarge page:

Allows text to be viewed in full screen.



Output Text Display Area - Displays text output from the 'Output' page in full screen.

BACK - Click to go to the previous page.

END OF DOCUMENT