VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CYBER SECURITY (CSE4003)

Slot: C1

J Component

Title: Steganography using RSA Algorithm

REVIEW 3

BY: Rahul Sanjeev- 19BCE0952
K Gokul Raj-19BCE0522
Aradhya Bagrodia- 19BCE0982

INDEX

ACKNOWLEDGEMENT

First of all, we thank the almighty who showered his immense blessings on us, that has helped us to complete this project with success.
Our heartful thanks to Prof. Joshva Devadas T for giving us the necessary environment to acquire knowledge and skills. Our sincere and warmest thanks for his valuable and inspiring guidance and encouragement given throughout the period of this project. We feel indebted to him for guiding our project with reviews, evaluations, guidance and suggestions offered throughout this project. We also express my whole hearted thanks to our parents for their encouragement to bring this project to a successful completion.

ABSTRACT

In this project we have tried to create an online system where users can send a text which is encrypted in an image. A sender must send it to the user of his choice by adding the Mobile number of the receiver. He can then choose the image that will be used as the medium of steganography and input his message that should be encrypted. The receiver must login as a receiver and then generate the private key required to decrypt the message. The receiver can then input the key and see the secret text that has been encrypted. The user can check a table of past messages and see the messages that have been sent through past conversations. We have used HTML for the frontend part of website. The values are taken in HTML form and then passed on to the Mysqli.connector that helps connect us to our backend servers. All the values that are passed to the HTML form is later collected in the database. We have implemented Steganography using RSA Algorithm with Python Apache.

REFERENCE PAPERS USED

[1] Burnett, S., & Paine, S. (2001). The RSA security's official guide to cryptography. McGrawHill, Inc..

[2] Gao, T., & Chen, Z. (2008). A new image encryption algorithm based on hyper-chaos. Physics Letters A, 372(4), 394-400.

[3] Puech, W., & Rodrigues, J. M. (2004, September). A new cryptowatermarking method for medical images safe transfer. In Signal Processing Conference, 2004 12th European (pp. 1481-1484). IEEE.

[4] Chen, G., Mao, Y., & Chui, C. K. (2004). A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons & Fractals, 21(3), 749-761

[5] Saranya et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5708-5709 A Study on RSA Algorithm for Cryptography

TECHNOLOGY USED

1) **Front-End:**
    - HTML
    - Bootstrap

2) **Back-End:**
    - CGI script (calling Python modules by directing form action to respective modules)
    - Mysqli.connector used in python to connect with Database(PHP Myadmin – Mysql server)
    - PHP – For checking session variables and href values to pages, and retrieving values in html from database

METHODOLOGY & RUN THROUGH

Modules used:

1) Back end (Connecting python using CGI script):

Python Modules:

- **RsaPublicKeyDist.py –**

  - Imported classes – Math , random, cgi , mysql.connector
  - Math function used for using mathematical functions
    Cgi – for taking input values from html forms
    Random – for generating random Public-Key, Modulus, Private-Key
    mysql.connector – for inserting, selecting and manipulating data in database though sql queries in mysql database
  - We generate random Public-Key, Modulus, Private-Key and store it in the **KeyLog** Table, creating a new record for all recievers
  - If the user clicks generate key, it creates a new record or updates the existing values of the public and modulus.

- **RsaSender.py –**

  - Imported classes – PIL, cgi , mysql.connector,os
  - PIL for manuplating the images
  - When the sender sends a message. Its Encryted and the characters are Converted to its respective ASCII binary values. These Values are then used to change the values of Pixels of the selected image for steganography
  - The Image is taken from image folder and Encrypted image is stored in the enc_image folder. The path of the enc_image is stored in database with sender mobile and the receiver mobile as the unique identifier of a record in the **MessageLog** Table

- **RsaReceiver.py –**

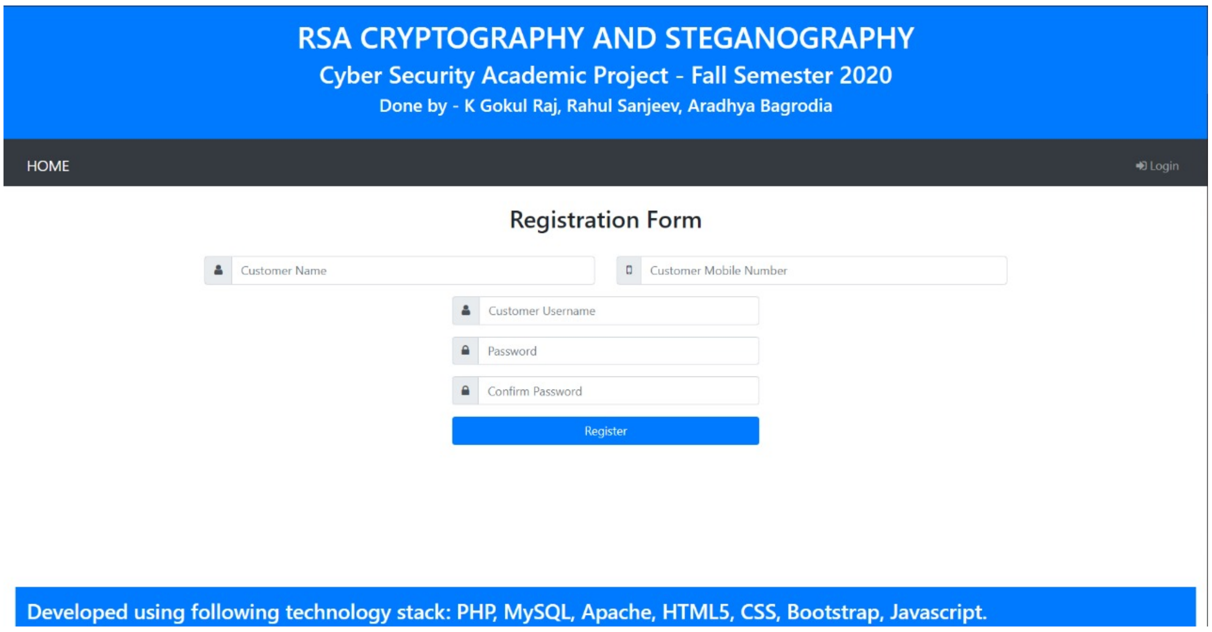
  - Imported classes – PIL, cgi , mysql.connector,os

- PIL for manuplating the images
- When the receiver sees his message log he can see the message sent. He can click the message he wishes to see. He has to enter the Private-Key which was generated and assigned to him.
- When he enter the private key, He can see the message sent to him by the sender

Front-End:

1) Signing in as a new User:

New Users must sign in before they log in to send and receive messages and they can do this easily by the Sign Up as new user option


RSA CRYPTOGRAPHY AND STEGANOGRAPHY

Cyber Security Academic Project - Fall Semester 2020

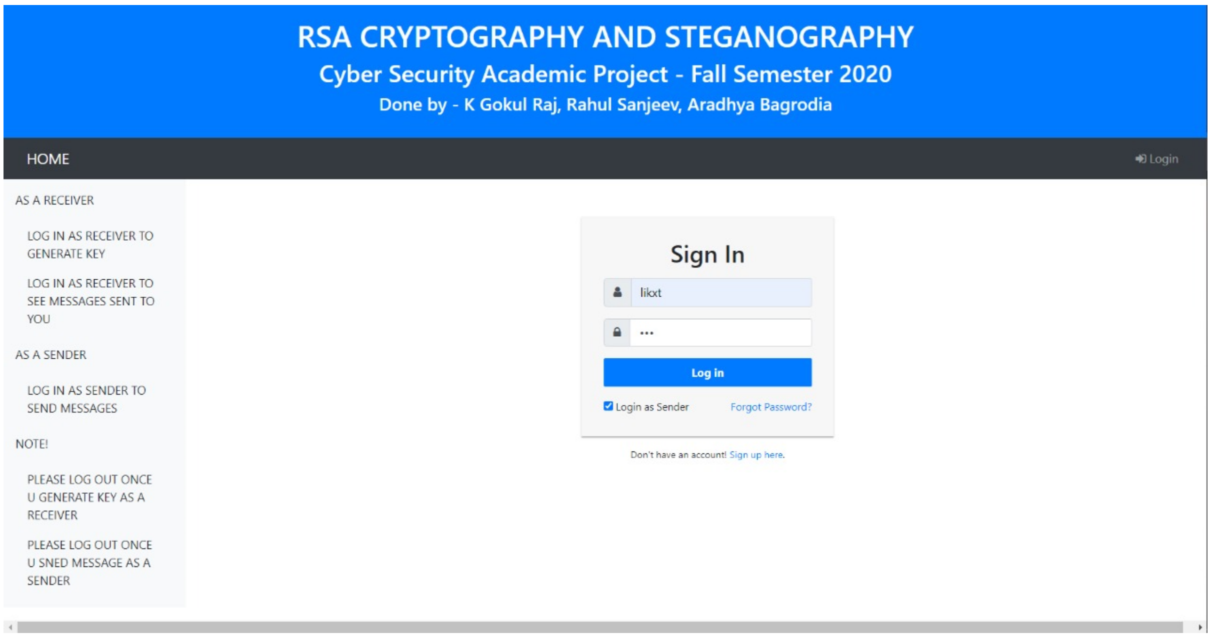
Done by - K Gokul Raj, Rahul Sanjeev, Aradhya Bagrodia

Registration Form

Developed using following technology stack: PHP, MySQL, Apache, HTML5, CSS, Bootstrap, Javascript.

2) Logging in as Sender:

We can put in our details and log in using your username and password to send messages



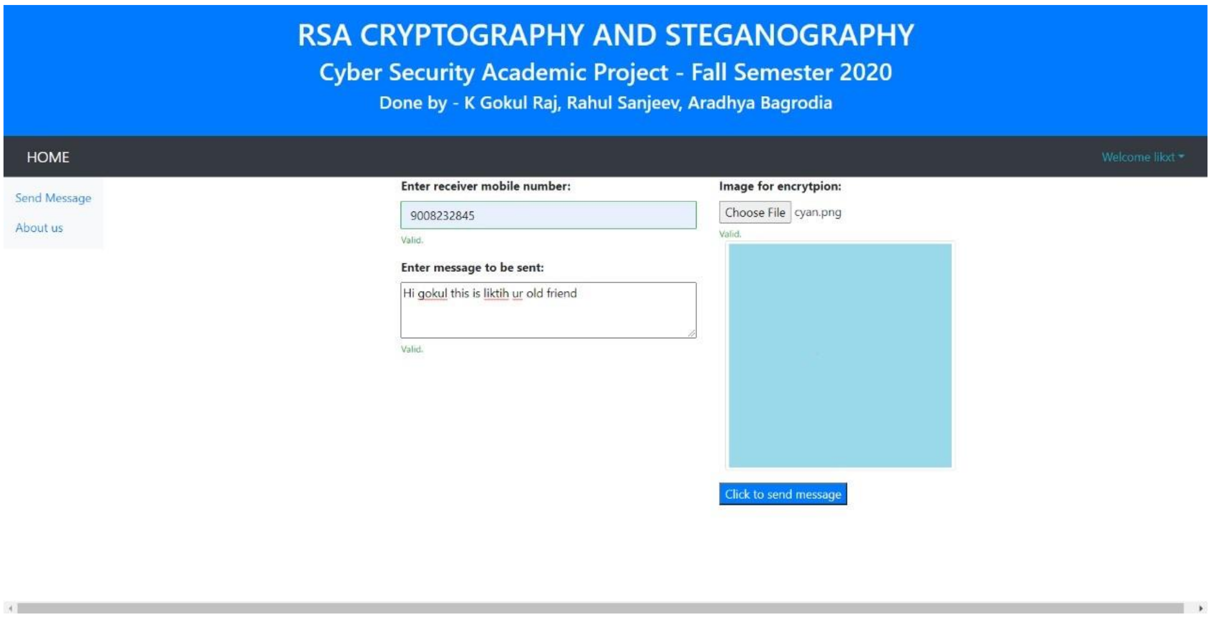
RSA CRYPTOGRAPHY AND STEGANOGRAPHY

Cyber Security Academic Project - Fall Semester 2020

Done by - K Gokul Raj, Rahul Sanjeev, Aradhya Bagrodia

HOME
Login
AS A RECEIVER
LOG IN AS RECEIVER TO GENERATE KEY
LOG IN AS RECEIVER TO SEE MESSAGES SENT TO YOU
AS A SENDER
LOG IN AS SENDER TO SEND MESSAGES
NOTE!
PLEASE LOG OUT ONCE U GENERATE KEY AS A RECEIVER
PLEASE LOG OUT ONCE U SNED MESSAGE AS A SENDER
Sign In
likxt
•••
Log in
Login as Sender
Forgot Password?
Don't have an account! Sign up here.

We can click on the send messages tab after logging in, enter the receiver's mobile number. We have to upload the image and enter the text that we want to encrypt



RSA CRYPTOGRAPHY AND STEGANOGRAPHY

Cyber Security Academic Project - Fall Semester 2020

Done by - K Gokul Raj, Rahul Sanjeev, Aradhya Bagrodia

Enter receiver mobile number: 9008232845

Valid.

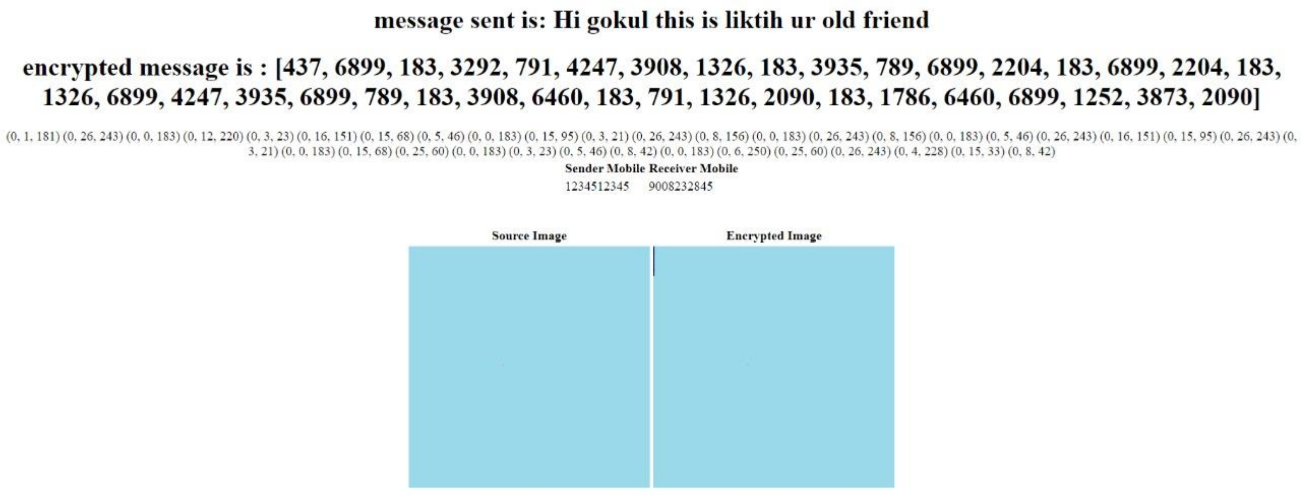
Enter message to be sent: Hi gokul this is liktih ur old friend

Valid.

Image for encrytpion: cyan.png

Valid.

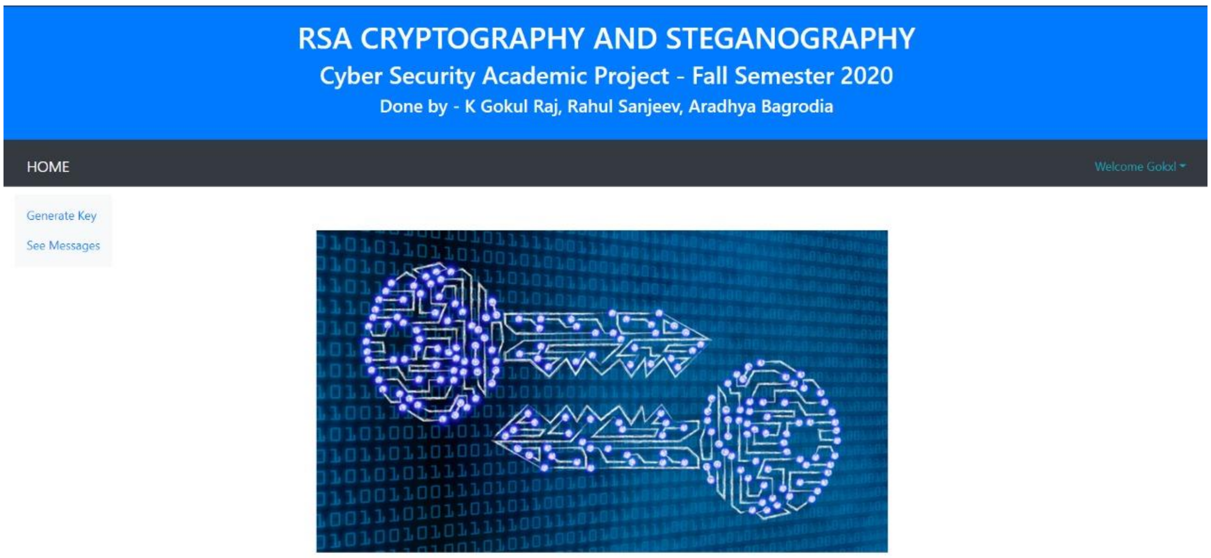
We can click to send the message and we can see the confirmation of the message sent. The encrypted image has a little line on the top left corner

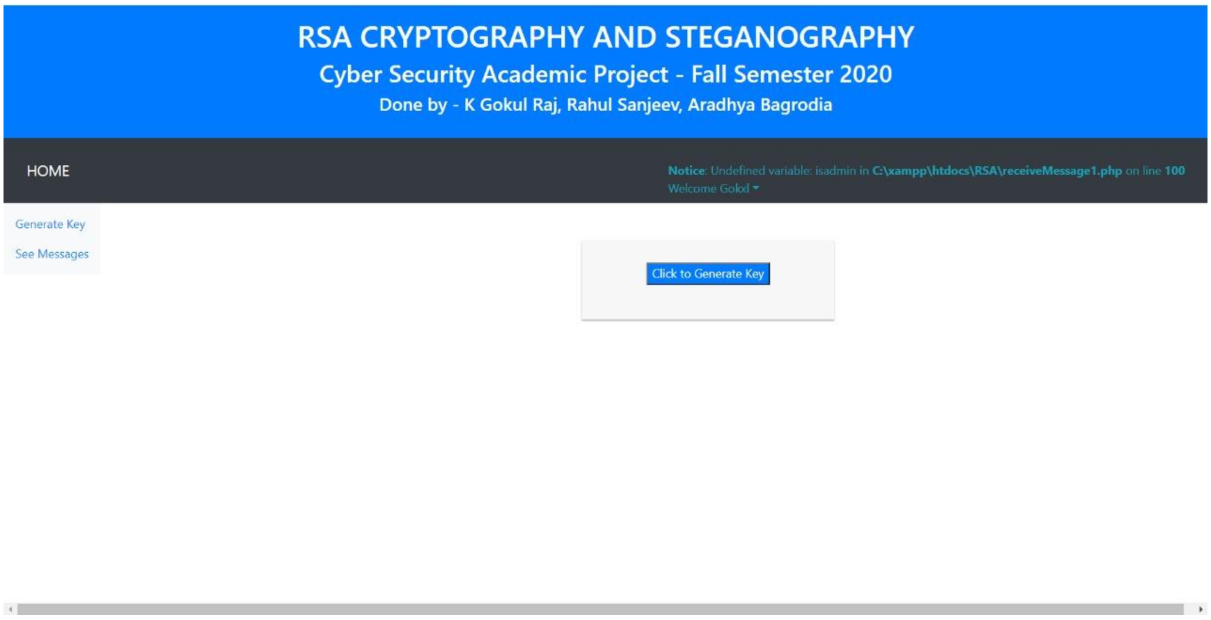


```
message sent is: Hi gokul this is liktih ur old friend

encrypted message is : [437, 6899, 183, 3292, 791, 4247, 3908, 1326, 183, 3935, 789, 6899, 2204, 183, 6899, 2204, 183,
1326, 6899, 4247, 3935, 6899, 789, 183, 3908, 6460, 183, 791, 1326, 2090, 183, 1786, 6460, 6899, 1252, 3873, 2090]

(0, 1, 181) (0, 26, 243) (0, 0, 183) (0, 12, 220) (0, 3, 23) (0, 16, 151) (0, 15, 68) (0, 5, 46) (0, 0, 183) (0, 15, 95) (0, 3, 21) (0, 26, 243) (0, 8, 156) (0, 0, 183) (0, 26, 243) (0, 8, 156) (0, 0, 183) (0, 5, 46) (0, 26, 243) (0, 16, 151) (0, 15, 95) (0, 26, 243) (0,
3, 21) (0, 0, 183) (0, 15, 68) (0, 25, 60) (0, 0, 183) (0, 3, 23) (0, 5, 46) (0, 8, 42) (0, 0, 183) (0, 6, 250) (0, 25, 60) (0, 26, 243) (0, 4, 228) (0, 15, 33) (0, 8, 42)
Sender Mobile Receiver Mobile
1234512345    9008232845
```

3) **<u>Logging in as Receiver:</u>**

You can log in as the receiver and first you have to generate a key. This key is randomly generated by the program.


RSA CRYPTOGRAPHY AND STEGANOGRAPHY
Cyber Security Academic Project - Fall Semester 2020
Done by - K Gokul Raj, Rahul Sanjeev, Aradhya Bagrodia
HOME
Welcome Gokul
Generate Key
See Messages



RSA CRYPTOGRAPHY AND STEGANOGRAPHY

Cyber Security Academic Project - Fall Semester 2020

Done by - K Gokul Raj, Rahul Sanjeev, Aradhya Bagrodia

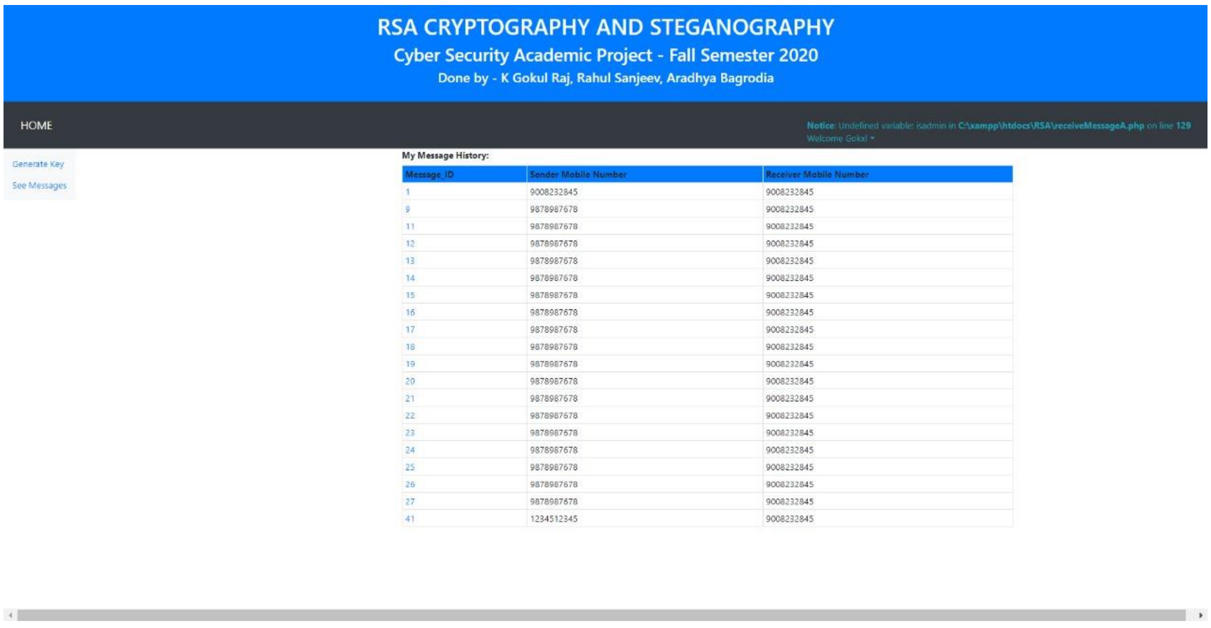
```
Notice: Undefined variable: isadmin in C:\xampp\htdocs\RSA\receiveMessage1.php on line 100
```

Please Remember Your Private Key is 1975

Public key is 7

Modulus is 7081

You can then login in again and select the See Messages tab. It opens a list of all received messages in a tabular form and then select the message you have to open



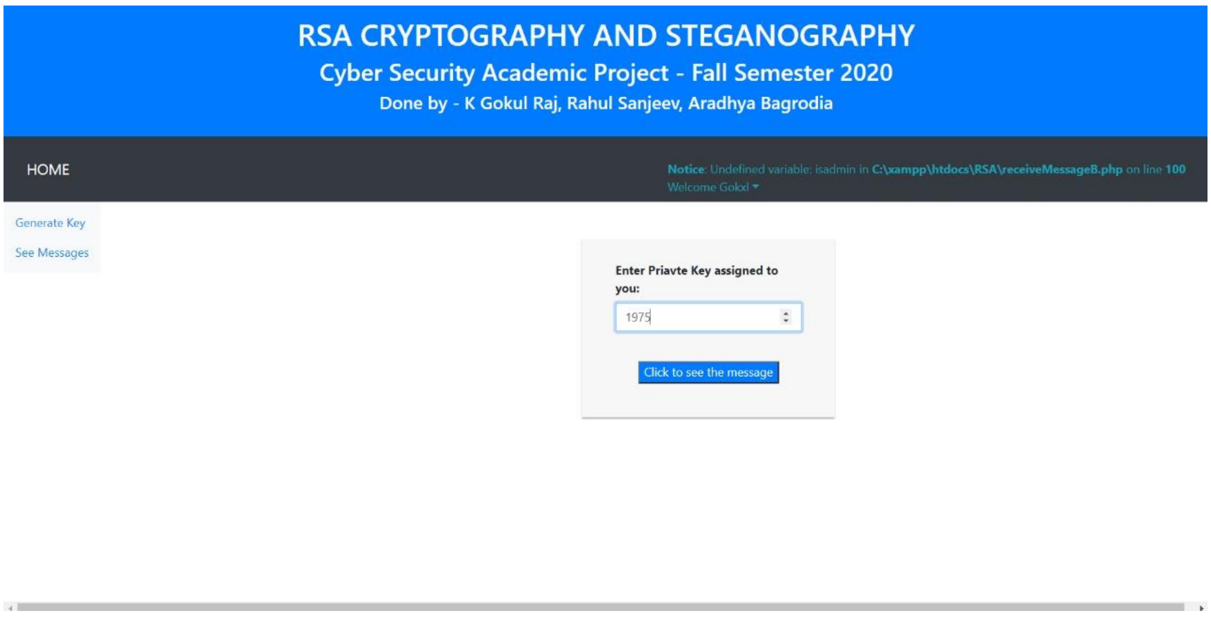
RSA CRYPTOGRAPHY AND STEGANOGRAPHY
Cyber Security Academic Project - Fall Semester 2020
Done by - K Gokul Raj, Rahul Sanjeev, Aradhya Bagrodia

```
Notice: Undefined variable: isadmin in C:\xampp\htdocs\RSA\receiveMessageA.php on line 129
```

My Message History:

Message_ID	Sender Mobile Number	Receiver Mobile Number
1	9008232845	9008232845
9	9878987678	9008232845
11	9878987678	9008232845
12	9878987678	9008232845
13	9878987678	9008232845
14	9878987678	9008232845
15	9878987678	9008232845
16	9878987678	9008232845
17	9878987678	9008232845
18	9878987678	9008232845
19	9878987678	9008232845
20	9878987678	9008232845
21	9878987678	9008232845
22	9878987678	9008232845
23	9878987678	9008232845
24	9878987678	9008232845
25	9878987678	9008232845
26	9878987678	9008232845
27	9878987678	9008232845
41	1234512345	9008232845

You can select the right phone number and then input the key. When you input the right key, the image gets decrypted and then the receiver can check the decrypted message sent.

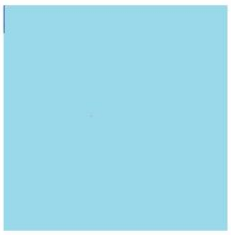


RSA CRYPTOGRAPHY AND STEGANOGRAPHY
Cyber Security Academic Project - Fall Semester 2020
Done by - K Gokul Raj, Rahul Sanjeev, Aradhya Bagrodia

```
Notice: Undefined variable: isadmin in C:\xampp\htdocs\RSA\receiveMessageB.php on line 100
```

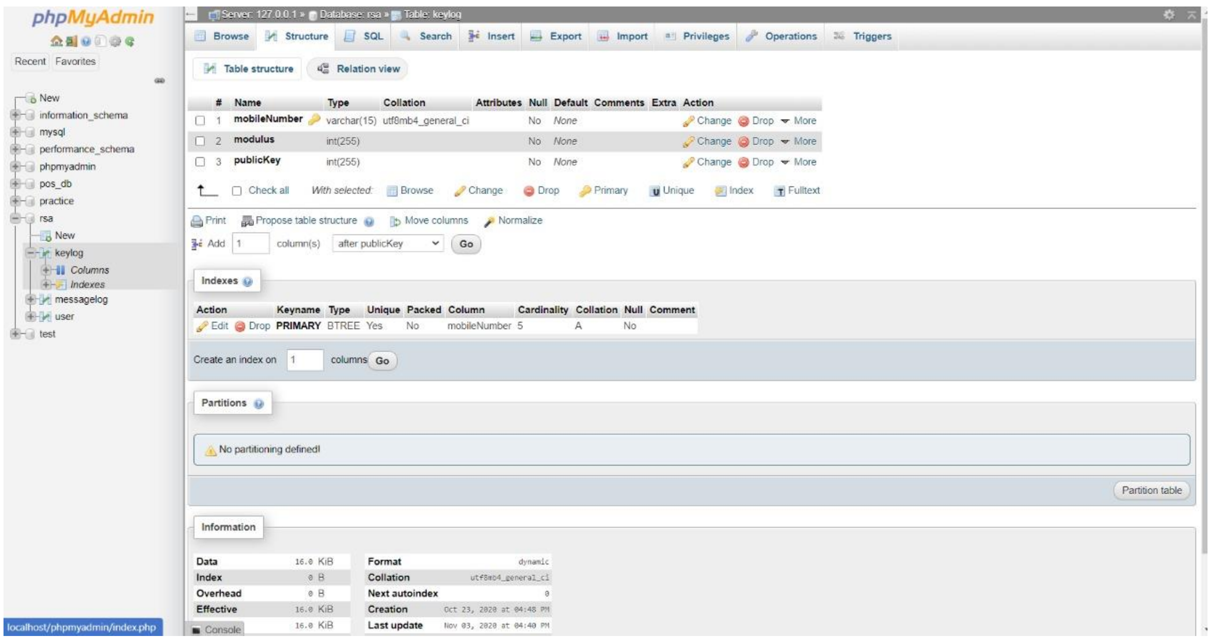
Enter Priavte Key assigned to you: 1975

```
msg length is 37

modulus values is 7081

file open successful

RGBtoI value is [437, 6899, 183, 3292, 791, 4247, 3908, 1326, 183, 3935, 789, 6899, 2204, 183, 6899, 2204, 183, 1326,
6899, 4247, 3935, 6899, 789, 183, 3908, 6460, 183, 791, 1326, 2090, 183, 1786, 6460, 6899, 1252, 3873, 2090]
```



```
Message is ['H', 'i', ' ', 'g', 'o', 'k', 'u', 'l', ' ', 't', 'h', 'i', 's', ' ', 'i', 's', ' ', 'l', 'i', 'k', 't', 'i', 'h', ' ', 'u', 'r', ' ', 'o', 'l', 'd', ' ',
'f', 'r', 'i', 'e', 'n', 'd']
```

We used the database Mysql.connector and PHP backend for the database management of all the information that is passed into the HTML form.



Server: 127.0.0.1 » Database: rsa » Table: keylog

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	mobileNumber	varchar(15)	utf8mb4_general_ci		No	*None*			Change Drop More
2	modulus	int(255)			No	*None*			Change Drop More
3	publicKey	int(255)			No	*None*			Change Drop More

Add 1 column(s) after publicKey

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	mobileNumber	5	A	No	

Create an index on 1 columns

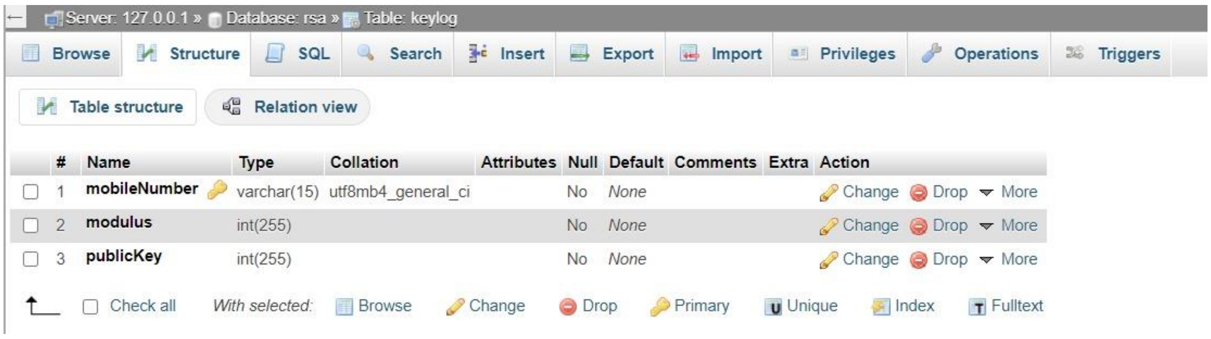
Partitions

No partitioning defined!

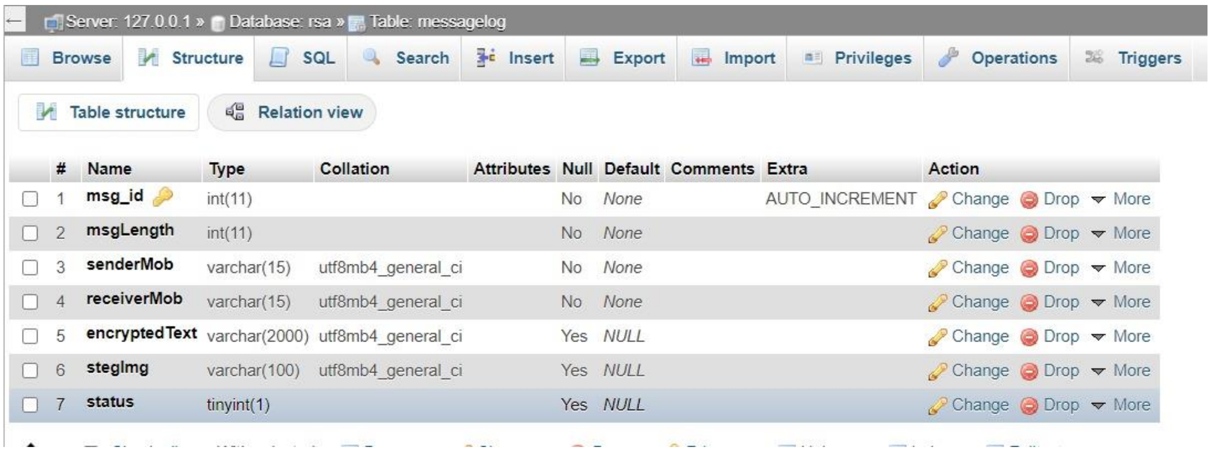
Information

Data	16.0 KiB	Format	dynamic
Index	0 B	Collation	utf8mb4_general_ci
Overhead	0 B	Next autoindex	0
Effective	16.0 KiB	Creation	Oct 23, 2020 at 04:48 PM
	16.0 KiB	Last update	Nov 03, 2020 at 04:40 PM

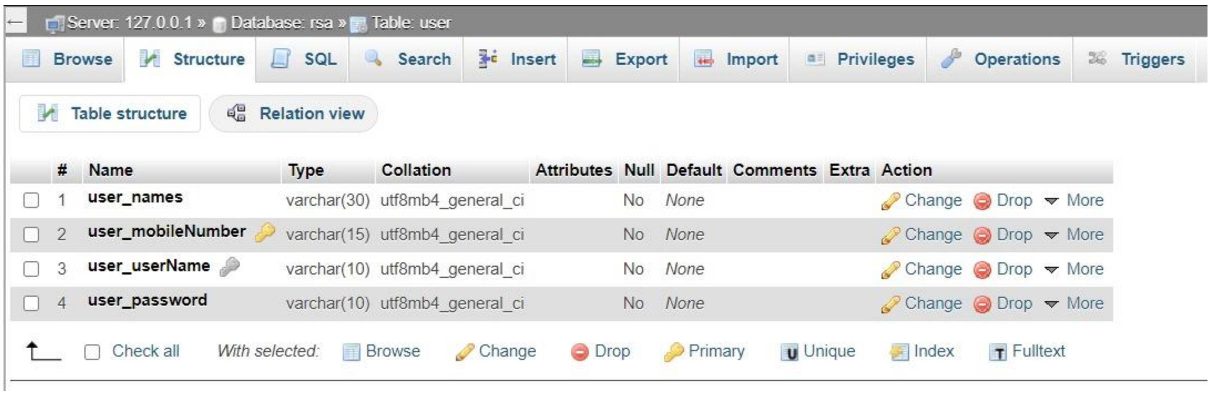
localhost/phpmyadmin/index.php



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**mobileNumber**	varchar(15)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	2	**modulus**	int(255)			No	*None*			Change Drop More
☐	3	**publicKey**	int(255)			No	*None*			Change Drop More



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**msg_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**msgLength**	int(11)			No	*None*			Change Drop More
☐	3	**senderMob**	varchar(15)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	**receiverMob**	varchar(15)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	5	**encryptedText**	varchar(2000)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
☐	6	**stegImg**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
☐	7	**status**	tinyint(1)			Yes	*NULL*			Change Drop More



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**user_names**	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	2	**user_mobileNumber**	varchar(15)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	**user_userName**	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	**user_password**	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More

CONCLUSION

We can conclude with this project that we have successfully created a system where users can safely send messages and receive messages. The messages are sent in the form of an encrypted image. By using the RSA Algorithm and connecting it with Steganographic methods by using an image as medium, we have made the sending and receiving of messages secure and safe. The security levels provided are more than RSA algorithm as steganographic methods are more secure. The users must use a randomly generated key which can only be seen by the receiver when he logs into our site. He must put the right key to decrypt the image and see the message. We have used all the materials that we were provided through our course and by the help of some Reference papers we have created a system to the best of our abilities

FUTURE WORKS

For the future, we have planned to make more improvements to our existing project. The current project works best with darker images. The encrypted text gets overlayed in the top left corner of the image as a line. Now this line is visible for light images but it is harder to notice for dark images. We want to make changes to the project in such a way that this anomaly isn't easily visible. This is the main thing that we will be striving to work for to make our project more safe and secure. We have used Python to code the implementation of RSA with Steganographic methods, hence it is easily achievable to make any changes to the existing code as Python is an easily and extensively used language

Originality report

COURSE NAME
Cyber Security - Project Report

STUDENT NAME
GOKUL RAJ K 19BCE0522

FILE NAME
GOKUL RAJ K 19BCE0522 - Cyber Plagiarism Checker Report Upload

REPORT CREATED
Nov 6, 2020

Summary

Flagged passages	2	0.3%
Cited/quoted passages	0	0%
Web matches		
stackoverflow.com	2	0.3%

1 of 2 passages
Student passage FLAGGED

…// **get loaded data and render thumbnail**.

Top web match

onload = function (e) { // **get loaded data and render thumbnail**. document.getElementById("image").src = e.target.result; }; // read the image file as ...

Show an image preview before upload - Stack Overflow https://stackoverflow.com/questions/14069421/show-an-image-preview-before-upload

2 of 2 passages
Student passage FLAGGED

…// **read the image file as a data URL**.

Top web match

target. result;); // **read the image file as a data URL**. reader.

Show an image preview before upload - Stack Overflow https://stackoverflow.com/questions/14069421/show-an-image-preview-before-upload