Design and Implementation of a Lock-Key Image Encryption and Decryption, based on a User Provided Password

[1]Harinandan Tunga
Computer Science & Engineering Department
RCC Institute of Information Technology
Kolkata, India

[2]Akash Ghosh
Computer Science & Engineering Department
RCC Institute of Information Technology
Kolkata, India

[3]Arnab Saha
Computer Science & Engineering Department
RCC Institute of Information Technology
Kolkata, India

[4]Swashata Ghosh
Computer Science & Engineering Department
RCC Institute of Information Technology
Kolkata, India

ABSTRACT
This paper is about encryption and decryption of images using a secret password provided by the user. The encryption machine takes the password and the source image as input and generates a key pattern image by using Secure Hash Algorithm (SHA) and a lock image by using Mcrypt algorithm. It provides additional security using Image scrambling. The decryption machine takes the lock image, key image and the password as input to generate the original image. It also checks if the input is valid and blocks the user when an invalid input is provided. In this paper the results are obtained by simulating the entire algorithm in PHP. You can check out the project by visiting www.imglockkey.com.

General Terms
Image Encryption and Decryption algorithm, Cryptography, Lock-Key Algorithm, Image Scrambling, Secure Hash Algorithm (SHA), PHP, PHPGD

Keywords
Mcript, SHA, MCRYPT_RIJNDAEL_256, SHA- 256, Lock-Key Algorithm, Image Scrambling

1. INTRODUCTION
What is data? Data can be anything starting from numbers, texts to images. Data in simpler term can be defined as information. So these data are highly valuable and certainly no one wants these data to fall in wrong hands. For example, a bank account passwords, credit information need to be kept safely and undisclosed from adversaries (third party). However a data during transmission from a client to a server or vice-versa is where it is most vulnerable. This particular point is where most of the data falls to the wrong hands and which might lead to havoc.

And that is how this particular problem leads to the concept of cryptography. Cryptography is an idea where the original data is coded into a stream of alpha-numeric array, transmitted and then decoded back to its original form by only those who are intended for can read and process. This idea leads to two advantages , firstly it definitely avoids the original data to fall in the hands of an adversary and secondly even if the alpha numeric data finds its place in wrong hands, it becomes not much of a use since the coded stream standalone means nothing but rubbish. After the transmission this stream of alpha numeric data should then be decoded back in order to retrieve to its original form. So the function that is used to code and decode the actual data at the two end points remains completely unknown to the third party and therefore an upper hand is obtained in the process.

2. REVIEW OF LITERATURE
As security and integrity of data has become the main concern in past few years due to exponential rise of threats from third party [3]. And in the present scenario almost all the data is transferred via network pathways and so are vulnerable to various threats. [1] [2] Has given a brilliant approach to image encryption based on blowfish algorithm. This approach can be used on both color and black & white images (.TIF images only). [5] Has given a performance analysis on [1] and was concluded that the modified blowfish algorithm is as strong as the original blowfish algorithm. [6] A partial encryption and decryption technique is used for transmitting data: a real-time approach for image and video communication and processing. In this method only a partial encryption is involved which is around 13-27% of the whole image is encrypted, reduces the computation time marginally.
[7] Dedicated work on AES (Advance Encryption Standard) which is a replacement for DES (Data Encryption Standard), shows implementation of AES in Very High Speed Integrated Circuit Hardware Description Language (VHDL). The AES: Rijndael is used in this algorithm (Lock-Key). The threats regarding the data security during a transfer and communication process has made researchers to work and research on the field of cryptography and thus led to invention of new algorithms which day by day becomes more secure and reliable. This paper is based on image encryption and decryption based on user provided secret key. The Lock-Key algorithm uses previously made encryption algorithms like SHA-1, SHA-256 and Rijndael-256.

3. PROPOSED METHOD
The proposed algorithm depends on two previously created algorithms:

3.1 SHA (Secured Hash Algorithm) Version 1
This is used intermediately to create the key image.

3.2 Mcrypt
This is used to intermediately create the lock image. More specifically following encryption algorithms are used:

- MCRYPT_RIJNDAEL_256: A specific AES (Advanced Encryption Standard) implementation also known as Rijndael. The main reason of using rijndael is, it is free for any public or private use, commercial or non-commercial.
- SHA- 256: Another version of SHA used to generate the key of mcrypt implementation. The reason to use SHA256 is it generates different output than SHA1 which is already used to make the lock image.

3.3 Algorithm

3.3.1 Generating Image from string:

This algorithm will be used on various situations. For the rest of the document, it will be referred as **GEN-ENC-IMG.**

1. To make this algorithm work, three parameters are passed:
    a. The encryption string (STR).
    b. The width of the image (WID)
    c. The height of the image (HEI)
2. Create an empty image (IMG) of width WID and height HEI.
3. Loop through every pixels of IMG, starting from top left, continuing across the width, thereby jumping to the next set of rows (of pixels) when it reaches the end of width and hence going all the way to bottom-right pixel.
4. The RGB (Red, Green, Blue) value of each of the pixels is calculated as follows:
    a. Check if the length of STR is less than 3. If less than three, then append a full copy of STR to the end of the hash string.
    b. Take 3 bytes from the beginning of STR and delete those 3 bytes from STR.
    c. For each of the byte's ASCII value (BASC) the corresponding color value (CV) is generated using the formula: CV = BASC % 256.
    d. The CV from first byte is assigned as the Red value, same of second and third are assigned as Green and Blue respectively.
5. Return the resulting image.

3.3.2 Encryption Engine:

The workflow of encryption engine is as follows:

3.3.2.1 Getting the Input:

1. Get the source image from the User, which is referred as **SOI**.
2. Get the password from the user, which is referred as **PASS**.

3.3.2.2 Calculate the first intermediate of the key image:

1. Calculate the SHA1 hash of PASS, which is referred as PASS-SHA.
2. Generate the intermediate key image (KEYI) by passing PASS-SHA, width of SOI, height of SOI to GEN-ENC-IMG.

3.3.2.3 Calculate the internal image for the lock image:

1. Calculate the RAW bit stream of the PASS, using SHA256 algorithm. This shall be referred as PASS-SHA2.
2. Create Mcrypt hash of PASS-SHA using Rijndael algorithm and PASS-SHA2 as the key of mcrypt. The output shall be referred as PASS-MCRP
3. Calculate the internal image (IL-IMG) by passing PASS-MCRP, width of SOI, height of SOI to GEN-ENC-IMG.

3.3.2.4 Calculate the second intermediate of the key image:

1. For every pixels of KEYI, do as follows:
    a. For every color value of RGB, do as follows:
        i. If the value of color is odd, then subtract it by 1.
2. Store the resulting image as second intermediate key image (KEYII).

3.3.2.5 Calculate the final lock image and key image:

1. Create an empty image LOCK with same height and width of SOI.
2. For every pixels of SOI:
    a. For every color values of the pixel (RGB color values):
        i. Subtract the corresponding color value of IL-IMG from SOI.
        ii. If the value is positive store to the corresponding position of LOCK.
        iii. If the value is negative, store the absolute value to the corresponding position of LOCK and increase the corresponding value of KEYII by 1.
3. Finally the modified KEYII is generated which is the Final Key Image (KEY-IMG).
4. Scramble the pixels of LOCK, if the scramble checkbox is checked.
5. The modified LOCK is generated which is the Final Lock Image (LOCK-IMG).

3.3.3 Decryption Engine:

3.3.3.1 Getting the input:

Take three inputs from the users:

1. The Key Image File (IMG-KEY).
2. The Lock Image File (IMG-LOCK).
3. The Password (PASS).

3.3.3.2 Authentication:

1. Generate authentication image (IMG-AUTH) by passing SHA1 hash of PASS, width of IMG-KEY, height of IMG-KEY to GEN-IMG-ENC.
2. For every pixel in IMG-KEY
    a. For every color in the pixel:
        i. Compare the color value with the corresponding pixel of IMG-AUTH.
        ii. If difference is 0 or 1 then accept.
        iii. If difference is greater than 1 then block
3. If block, then not authenticated.

3.3.3.3 Generating the Source Image:

1. Calculate the SHA1 hash of PASS, which is referred as PASS-SHA.
2. Calculate the RAW bit stream of the PASS, using SHA256 algorithm. It is referred to as PASS-SHA2.
3. Descramble the IMG-LOCK if the descramble checkbox is checked.
4. Create Mcrypt hash of PASS-SHA using Rijndael algorithm and PASS-SHA2 as the key of mcrypt. The output shall be referred as PASS-MCRP
5. Calculate the internal image (IL-IMG) by passing PASS-MCRP, width of SOI, height of SOI to GEN-ENC-IMG.

6. Create an empty image (IMG-EMP) of same width and height as of IMG-LOCK.
7. For every pixels of IMG-LOCK:
   a. For every color of pixel:
      i. If the color value IMG-KEY is even, then: IMG-EMP[pixel][color] = IMG-LOCK[pixel][color] + IL-IMG[pixel][color]
      ii. Else: IMG-EMP[pixel][color] = IL-IMG[pixel][color] - IMG-LOCK[pixel][color]

The modified IMG-EMP is generated which is the original image IMG-ORG.

4. RESULTS AND DISCUSSION

In this paper the algorithm is simulated by making an application using PHP. The user can check out the project by visiting www.imglockkey.com. This page contains the basic project information. The user can select from the two options:

1. Encrypt Form
2. Decrypt Form

According to the option selected the user will get access to the encryption or the decryption form.

Three images are used for encryption to get the corresponding lock and key image. Finally the decryption algorithm is used to get back the source image.

4.1 Case 1

Here a color JPEG image is used.

Figure 1: Encryption Machine Input (Image 1)

The encryption form is selected and appropriate inputs for the image and password is entered. The user should unselect the Image Scrambling checkbox and submit the form.

The encryption form gives outputs with the corresponding key image and the lock image.

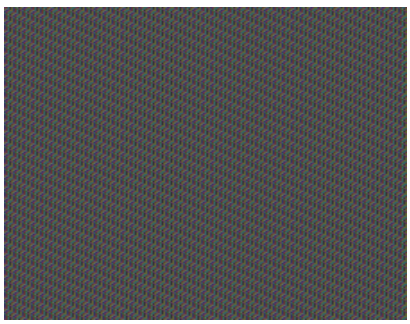


Figure 2: Key Image of Image 1

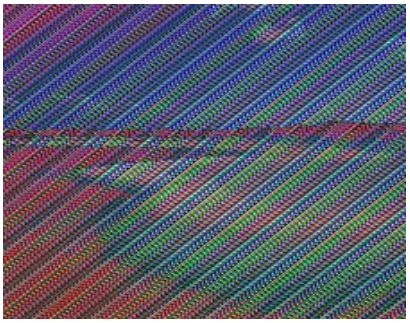


Figure 3: Lock Image of Image 1

If Image Scrambling checkbox was selected, the following lock image would have been produced. The key image is not scrambled.

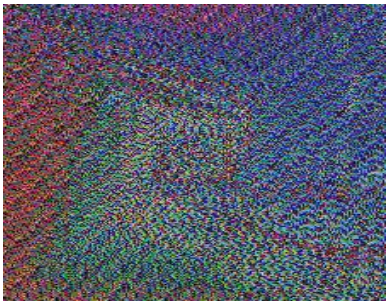


Figure 4: Lock Image (Scrambled) of Image 1

Now the user should select Decrypt and load the decrypt form and enter appropriate inputs for the key image, lock image and the password and select if scrambling was selected or not. Then submit the form. The decryption machine outputs an image which is the same as the source image.

Figure 5: Decryption Machine output of Image 1

Comparing the results of Figure 3 and Figure 4, a much greater level of visual security is provided with the image scrambling method. The original image which is partially visible in Figure 3 is not visible at all in Figure 4.

4.2 Case 2

Here a black and white image GIF image is used.

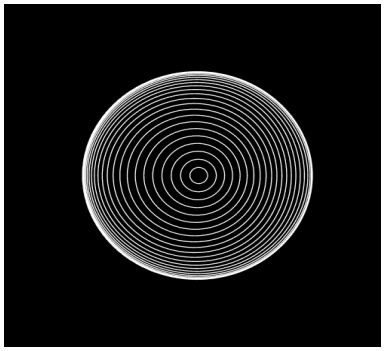


Figure 6: Encryption Machine Input (Image 2)

The encryption form gives outputs with the corresponding key and lock image without scrambling.

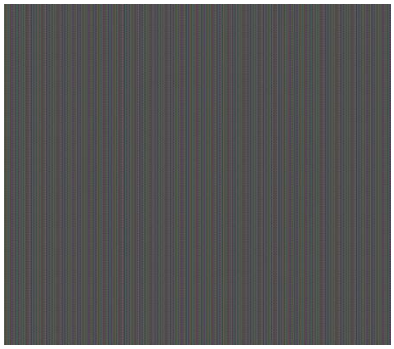


Figure 7: Key Image of Image 2

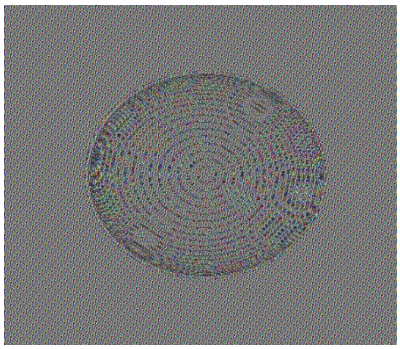


Figure 8: Lock Image of Image 2

If Image scrambling is selected, the Lock Image will be

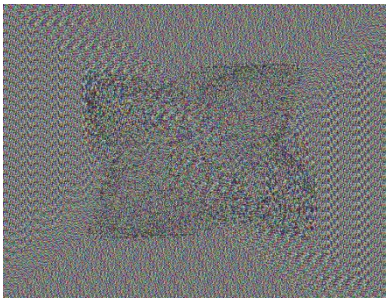


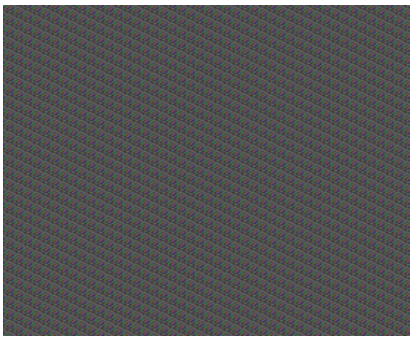
Figure 9: Lock Image (Scrambled) of Image 2

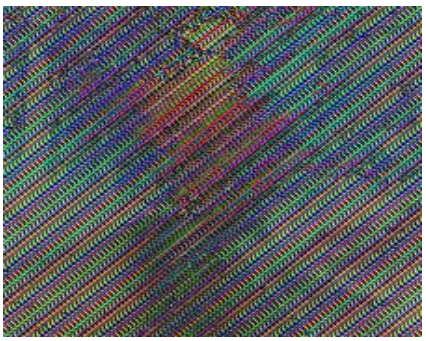
And finally the source image is generated from the key and the lock images.

4.3 Case 3

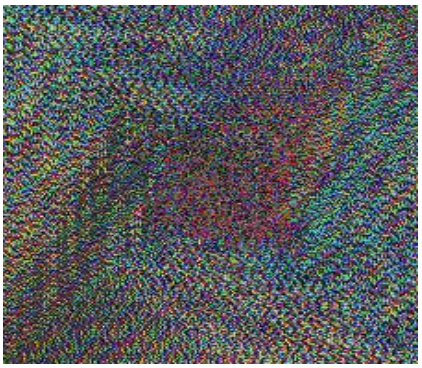
Here a color PNG image is taken.

The encryption form gives outputs with the corresponding key and lock image.





On selecting Image scrambling, the Lock Image will be



And finally the source image is generated from the key and the lock images.

5. CONCLUSION

An image of any color and of any extension (.GIF, .JPEG, .PNG) can be well encrypted and decrypted with the Lock-Key algorithm. The algorithm does not use any proprietary applications or libraries and consumes very little

resources. Since the Lock-Key based on user provided password uses strong complex algorithms like SHA-1, SHA-256, Rijndael-256 to generate the Final Lock and Key Image with help of user provided secret key which makes it exceptionally challenging to break. And also the images produced are secure and reliable in the sense that even if both the images get stolen, it would still be impossible to crack the original one without having the passkey and access to the server. The algorithm can be implemented easily on any web architecture. In cases where the client cannot be asked to install any third party applications, a server implementation of the engine will provide a cost effective way to give encryption-decryption functionalities to multiple clients. Based on the diverse examples given to test the algorithm's robustness, strength, accuracy and security it can be concluded that the Lock-Key algorithm can be of an excellent standard in the major of Image Encryption.

6. REFERENCES

[1] Image encryption and decryption using blowfish algorithm by Irfan Landge, Burhanuddin Contractor, Aamna Patel and Rozina Choudhary.

[2] A New Algorithm of Encryption and Decryption of Images Using Chaotic Mapping by Musheer Ahmad and M. Shamsher Alam.

[3] Atul, Kahate, Cryptography and Network Security, (Second Edition 2008).

[4] Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, (Second edition, 1996).B. Simpson, et al, "Title of paper goes here if known," unpublished.

[5] Dr. V Ramaswamy, Krishnamurthy G N, "Performance Analysis of Blowfish and its Modified Version using Encryption quality, Key sensitivity", International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009, pp. 1-4.

[6] H. Cheng, X.B. Li, "Partial encryption of compressed image and videos", IEEE Trans. Signal Process. 2008 48(8), pp. 2439-2451.

[7] Irfan Abdulgani Landge, Implementation of AES Encryption & Decryption using VHDL, International J. of Engg. Research & Indu.Appls. (IJERIA). ISSN 0974-1518, Vol. 4, August 2011, pp. 395-406.

[8] "Design and Implementation of a Novel Authentication Algorithm for Fool-Proof Lock-Key System Based On Visual Secret Sharing Scheme", by Harinandan Tunga, and Soumen Mukherjee, IJCSI Vol.9 Issue 3, No 1, May 2012, ISSN (Online): 1694-0814. www.ijcsi.org.