Sumanth C *et al*, International Journal of Computer Science and Mobile Applications,

Vol.3 Issue. 10, October- 2015, pg. 13-19 ISSN: 2321-8363

A Stochastic Digest Based 'Falsification Encryption' and 'Re-establishing Decryption' of Multimedia Data in Android device

[1]Sumanth C

[1]M.Tech Student,
Department of Computer Science and Engineering,
Sir Visveshwaraiah Institute of Science &
Technology,
Madanapalli- 517325, Andhra Pradesh, India,
sumanthc1989@gmail.com

[2]K Dinesh Kumar
[2]Assistant Professor,
Department of Computer Science and Engineering,
Sir Visveshwaraiah Institute of Science &
Technology,
Madanapalli- 517325, Andhra Pradesh, India,
dineshkumar0904@gmailcom

ABSTRACT

When it comes to era of modern age of computing it is evident that human beings have made their migration from static life style to dynamic lifestyle, i.e., all users activities will be carried out through usage of mobile. Mobility is totally been involved in all people lifestyles, it is very hard to find the person without a mobile phone, especially in well-developed or developing countries. Consider any mobile devices; they are mainly concerned with efficient power (Battery usage), processor time and memory utilization but when comes to security we are lagging behind in providing security to the data present in the mobile [1]. We came across various cryptographic algorithms but all the cryptographic algorithms which we are using their working is mainly concentrated to work better with desktop applications. There are very least known or no algorithms on cryptography for mobile applications. The available cryptographic algorithms like DES, AES, Triple DES, elliptical curve etc. algorithms demand much computational power than mobile devices can provide. In this paper we will be proposing the efficient and 2 layer secured 'distortion encryption' and 'restoring decryption' methods for multimedia data for portable mobile devices especially for images (Multimedia) [2][3].

Keywords - *Message digest, encryption [4], positional digest image, distortion encryption, restoring decryption*

1. INTRODUCTION

Mobile phones are undetectable part of human beings in recent days, and they have many flaws because of their size, limited computation power, and limited battery life. So far as we know, at recent days android mobile phones are available in the market with some MHz (probably 500- 100 MHZ) of processing power and RAM of (100 - 400) MB with 32 GB expandable external memory slot. When these portable devices are compared with the processing power of the desktop computer (are in terms of GHz), comparison vary in big factor. So there needs to be one efficient and secure multimedia encryption and decryption algorithm for mobile devices which consumes less processing power and battery life (for ex- some of military applications require the real-time encryption of images captured from soldier's mobile phone before sending the captured image to the military repository, in order to prevent the data leakage to enemy groups) [5].

The idea involved in proposing the distortion encryption and restoring decryption is message digest algorithm, the message digest algorithm which we used in this paper is MD5[6] (the algorithm which generates Message digest is secure one-way hash function that take arbitrary-sized data and output a fixed-length hash value). The MD5 hash also known as checksum for a file is a 128-bit value, something like a fingerprint of the file. There is a very small possibility of getting two identical hashes of two different files.

The following figures will give the brief working procedures of the algorithms to be discussed in the later part. Following Figure 1 shows input and outputs of distortion encryption algorithm, which shows input to the algorithm is original unencrypted image and a encryption key. Output of algorithm is encrypted image.

Figure 2 shows input and outputs of restoring decryption algorithm. Input to the algorithm is encrypted image and a key. Output of algorithm is original image.

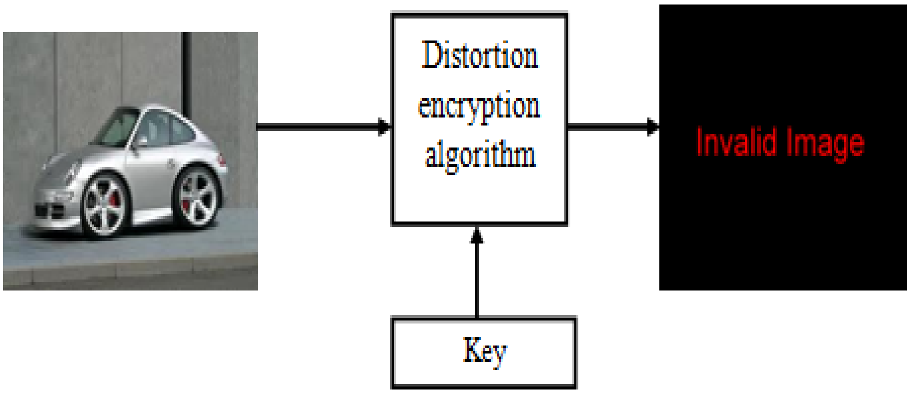


Figure 1: Inputs and output of distortion encryption algorithm

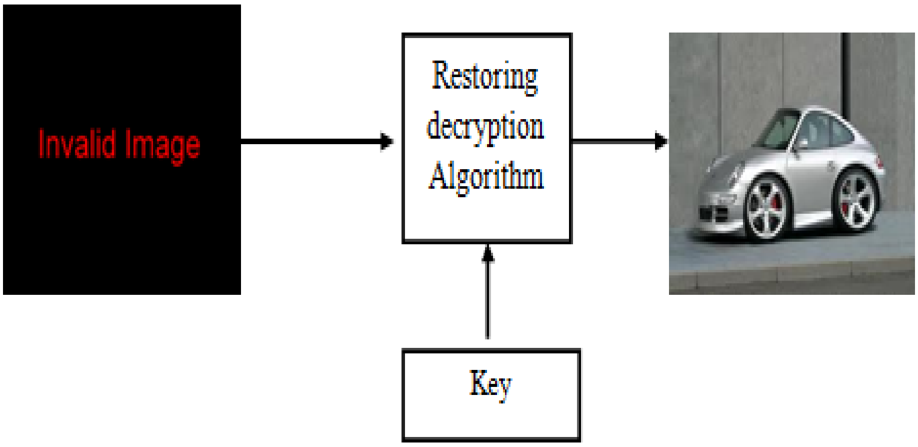


Figure 2: Inputs and output of restoration decryption algorithm

2. DISTORTION ENCRYPTION ALGORITHM

The steps are as follows.
Step1: Obtain the key from the user.

Step2: Obtain the byte stream of the image.

Step3: Obtain the message digest of the key using message digest algorithm.

Step4: Obtain the byte stream of message digest.

Step5: Calculate the number of bytes in key (position for starting distortion) also called as P_{desort}.

Step6: Insert the calculated message digest bytes from the position calculated (P_{desort}) in byte stream of images.

3. RESTORATION DECRYPTION ALGORITHM

Step 1: Obtain the key from the user.

Step 2: Obtain the byte stream of the image.

Step 3: Obtain the message digest of the key using MD5 algorithm.

Step 4: Obtain the byte stream of message digest.

Step 5: Calculate the number of bytes in key (position for starting distortion) also called as P_{desort}.

Step 6: Delete the calculated message digest bytes from the position calculated (from Pdesort to P_{desort} + Message digest bytes) from byte stream of images.

4. WORKING PRINCIPLE

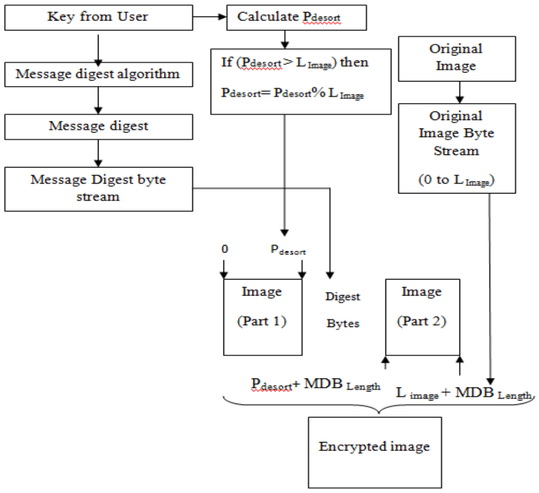


Figure 3: Distortion encryption

The working of above mentioned distortion encryption algorithm is depicted thoroughly in Figure 3. The inputs for distortion encryption algorithm are key from the user and original image.

The following Figure 4 depicts the restoring decryption algorithm. In which original image which has been encrypted using distortion encryption will be decrypted if the message digest bytes obtained by user entered decryption key matches with the encrypted image bytes from position p_{desort} to $P_{desort} + MD_{length}$

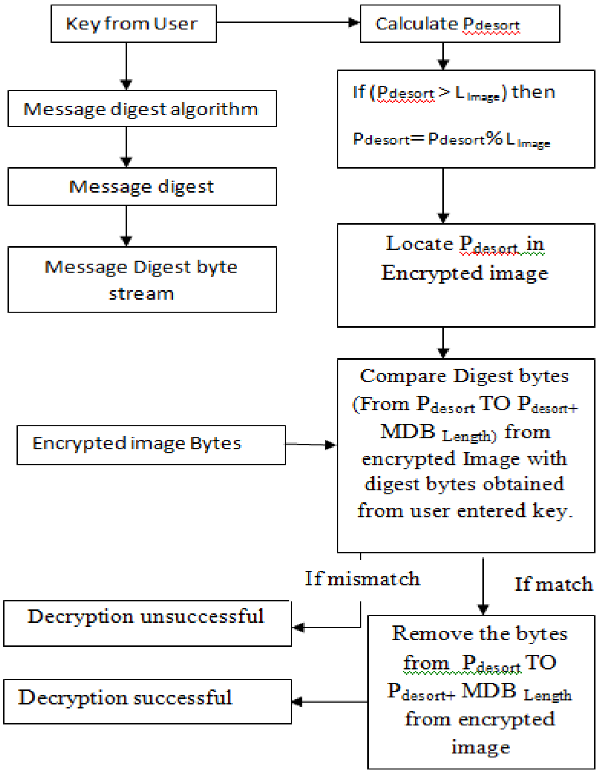

Key from User
Calculate Pdesort
Message digest algorithm
If (Pdesort > L Image) then
Pdesort= Pdesort % L Image
Message digest
Message Digest byte stream
Locate Pdesort in Encrypted image
Compare Digest bytes (From Pdesort TO Pdesort+ MDB Length) from encrypted Image with digest bytes obtained from user entered key.
Encrypted image Bytes
If mismatch
If match
Decryption unsuccessful
Remove the bytes from Pdesort TO Pdesort+ MDB Length from encrypted image
Decryption successful

Figure 4: Restoring decryption

5. IMPLEMENTATION

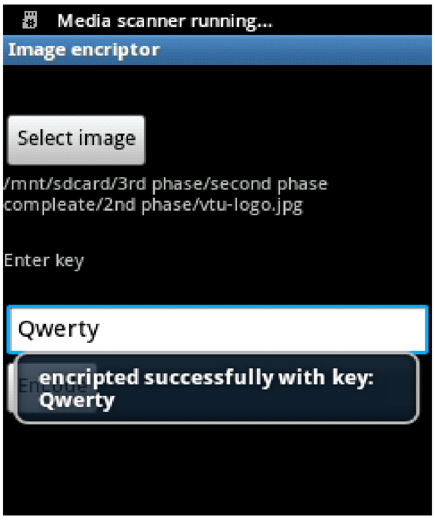
The proposed above algorithms are thoroughly and practically implemented and tested in mobile phone with the android operating system [7] (V 2.3 ginger bird and it is compatible with all the previous versions from 1.5 to 2.2). The proposed methods for encryption and decryption works very good with the portable devices like mobile phone with less power and time consumption. Following are some of the Results which will give the details of working and implementation of the above mentioned algorithms

The pseudo code of implementation of distortion encryption is given as follows. Restoring decryption is carried in reverse steps as that of distortion encryption with some computational and implementation differences.

```
FileInputStream fis=new FileInputStream(original image);
byte[] originalimage=new byte[fis.available()];
fis.read(originalimage);
distortion_encript(fis,key)
FileOutputStream fos=new FileOutputStream(f);
fos.write(encriptedimage);
```

6. RESULTS AND OUTCOMES

Following snapshot gives the encryption UI and decryption UI in mobile phone. User has to select the image from the one of the available (while encrypting) encrypted image (while decrypting), after selecting the image he has to enter the Key to encrypt or encrypt. The decryption and encryption keys must be for proper working of the algorithm. If any mismatches in the stored digest in the image and newly calculated digest the image will not get decrypted successfully. The selected image will be encrypted using the key provided using the distortion encryption algorithm.



Media scanner running...

Image encriptor

/mnt/sdcard/3rd phase/second phase compleate/2nd phase/vtu-logo.jpg

Enter key

Qwerty

encripted successfully with key: Qwerty

Figure 5: Encryption UI

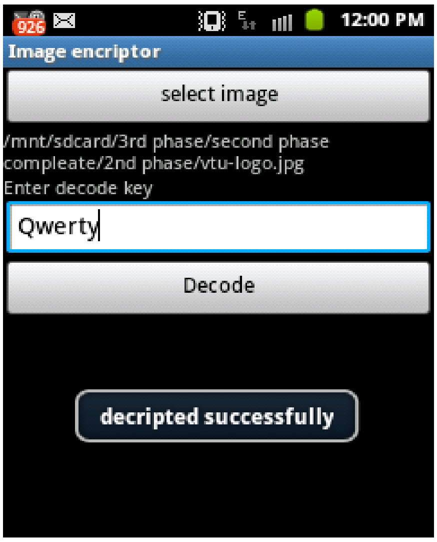


Image encriptor

/mnt/sdcard/3rd phase/second phase compleate/2nd phase/vtu-logo.jpg

Enter decode key

Qwerty

decripted successfully

Figure 6: Decryption UI

The Following outcome shows the original image and image after decryption. The Figure 7 shows the original image before encryption and the Figure 8 shows the outcome of the distortion encryption algorithm with the user entered key(for example as shown in Figure 5 : qwerty) and original image.

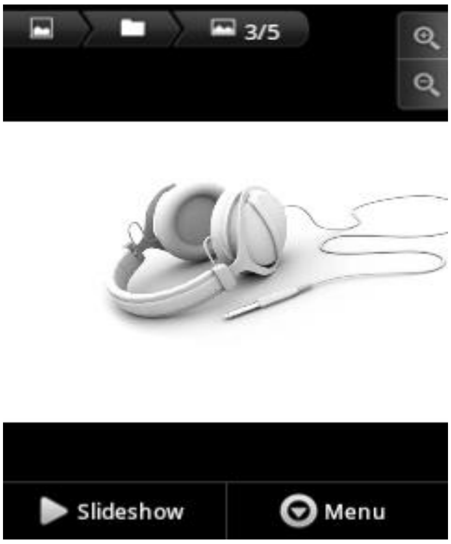


Figure 7: Before encryption

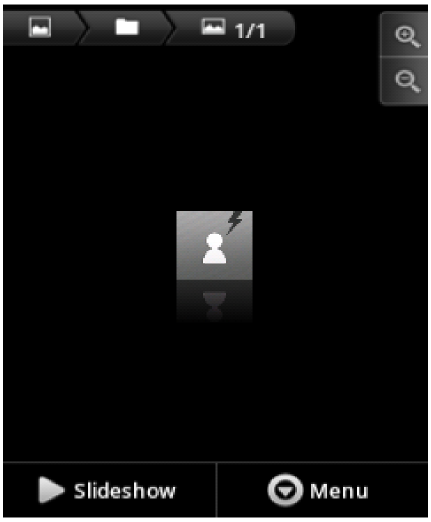


Figure 8: After Encryption

7. CONCLUSION

The whole summary of this work is to provide the efficient low power and low processor time consumption algorithm for encrypting and decrypting the multimedia data in mobile phone. As the proposed algorithm is simple and it works fine with the user key size of any length. The proposed algorithm provides two layers of security while decrypting, First by determining the P_{desort} with user provided key. Second by comparing each bytes of the digest stored in the image with the newly calculated value of digest bytes. If there is any mismatch in digest bytes, decrypting process will be suspended. This work also shows that how to apply the proposed distortion and restoration algorithms for images.

Further, this proposed work can be extended for many available multimedia types (Example- videos).

This encryption decryption algorithm can be integrated with camera of mobile phone, to encrypt the images/videos captured form camera at real time before saving to phone's memory hence providing the security for data at dynamic time.

REFERENCES

[1] Mobile Code Security in Contemporary Information System –Past, Present and Trends Denis Trcek, Faculty of Computer and Information Science, University of Ljubljana, Trzaska cesta 25, 1000 Ljubljana, Slovenia – EU.

[2] YuChen Dept. of Electr. & Comput. Eng., SUNY -Binghamton, Binghamton, NY Wei-Shinn Ku, "Self-Encryption Scheme for Data Security in Mobile Devices" Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE.

[3] Types Of Cryptographic Algorithms: http://i.thiyagaraaj.com/tutorials/introduction-of-cryptography/types-of-cryptographic-algorithms

[4] General survey on massive data encryption. Mengmeng Wang North China Univ. of Water Resources & Electr. Power, Zhengzhou, China.

[5] Shankar,T.N. CSE, GMRIT Rajam, Srikakulam, India
Sahoo, G. ; Niranjan, S. "Image Encryption for mobile devices", Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference.

[6] MD5: http://en.wikipedia.org/wiki/Md5

[7] Android Operating System: http://en.wikipedia.org/wiki/Android_(operating_system)