Thumbnail Preserving Encryption for JPEG

Gagan Solur Venkatesh
Graduate Student
University at Albany-SUNY, NY
gsolurvenkatesh@albany.edu

Vishwanath Murundi Dhananjaya Murthy
Graduate Student
University at Albany-SUNY, NY
vmurundidhananjayamurthy@albany.edu

Abstract

Now a day, security is very important aspect to one's document. Day to day the data is increasing and people are using cloud memory to store their documents. And the cloud security to our data is a major concern. This project proposes an encryption technique which helps in securing the image in cloud storage. There are few encryption techniques which are currently in use will encrypt files in such a way that the file contents can be read by the uploaded user. In our technique we allow only the reconstruction of small thumbnails, but not the entire content of the image from the cipher image, and hence when attacker get the files he will not get complete data.

Introduction

Cloud service is widely used, it has become very popular tool to for storing the day and sharing it with the users. This widely use cloud service is popular because, cloud services provide storage space in low cost, available in any part of the world, reliable, provides data even though the network failure and there is no limit for storage.

Same time the cloud service providers are desired for the users and, they need the user to have faith in storage provides for security of their data from outside attackers, which happened once "Apple iCloud accounts [1]", in which intimate photos were stolen from hundreds of victims and posted to the Internet. And, the users should have faith in cloud service so that no the outsiders, perhaps the inside employees also not misusing the data, "which happened in Facebook [2]".

The best way that the user need to encrypt his images before uploading it over any available cloud service. He can use a key which is only know to him for encryption so that it makes attackers difficult to get the original images from cipher image. Now the user need to think how to encrypt the data before storing it. First, traditional file encryption tools like PGP or openssl which gives a

blur or opaque output, which is not accepted by the cloud services, which they only accept the valid format. Format compliant encryption schemes provides encryption on the image data with valid. “Recent work on P3 [3] Cryptogram [4]” has proposed using similar schemes to protect images posted to social networking sites.

Figure 1: Different Approaches to Cloud Storage of Images. (a) Images stored in plaintext are available to both user and hacker. (b) Conventionally encrypted. Less usable for the client but secure from attackers. (c) Thumbnail-preserving. Attacker has access to the obscured version only. Users can preview thumbnails, then download at full resolution and decrypt locally.

Background

The best way to protect the confidentiality of sensitive data from third party user is to encrypt the image using a secret key before uploading on any cloud service. For desirable encryption, there are number of ways, among them the most important is, it should computationally secure, that any attacker tries to decrypt the image he should not achieve it without a secret key that is called “one-wayness [5]”. The goal of this project is one way, but not semantically secure. Intention is to prevent the attacker by not knowing our pixel values, but still we reveal some basics but not the main content.

Advantages

1) Our Images are in the format such that it should be able to support the cloud services like Facebook, Apple iCloud, Google Drive.
2) Main advantage is, we are hiding the main content of the image such that the cloud service or an attacker cannot get sufficient data of the image.
3) The user can reverse the encryption technique to get the original image from cipher image.

4) The crude features of the images can be effectively managed by the user from low resolution images from cloud. But only the user who knows the original images can recognize it.
5) User can degrade the image, so that when cloud services compresses the image. That's the reason the social media sites apply aggressive compression to all photos that are uploaded.

Limitations

1) We worked on .bmp files instead of .jpg/.jpeg files types and hence it supports .bmp files.
2) Supports only for image mime type but not for other mime types.

Methodology

Implementation Stated in Paper

Take M*N pixels of an image, private key and Block Size B, the images is equally divided into the blocks of neighboring pixels, B*B squares. The main goal of the author is to encrypt each block so that the cipher image expose the average pixel's value but nothing more than that. It will allow the cloud service to reconstruct the cipher image without private key. They can achieve only a low-resolution thumbnail where the details smaller than the block size, but not the original high definition image.

The user who is having the private can decrypt the cipher image to get the original high definition image. The project is designed in such a way that even the cipher image is loosely compressed, the decrypted image still has the high definition original image, the image quality reduces as the JPEG quantization increases.

If the bock size is B, then divide the original image into B*B pixels and encrypt the pixels by its neighboring pixels within the block. As a result, the original location of the pixel is not shown in cipher image, it makes the attackers difficult to reconstruct the original image. We shuffle the locations of the pixels within the block, using our cryptographically secure "PRNG to drive a random shuffle algorithm from Fisher and Yates [6]".

The following images shows the pixels shuffled in 32x32 blocks[Figure 2] and 16*16 sub blocks within the 32*32 blocks[Figure 3].

Figure 2: (left) Original and (right) 32x32 block-shuffled

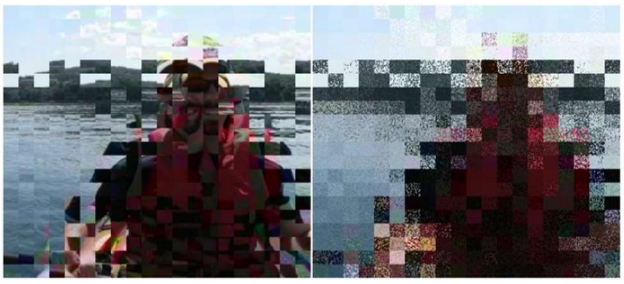

Figure 3: (left) After shuffling 16x16 sub-blocks within 32x32 blocks and (right) after shuffling pixels in sub-blocks

Implementation achieved in our approach

In our implementation, we ask the user to browse the image file [.bmp file] and once the user press encrypt button, we check the mime type of file and then we check whether the directory to save the encrypted file is available or not and if it is available the file will be stored in the specified directory. We are encrypting the image file by swapping the bytes of images and this swapping of bytes is done until all bytes are swapped for 145 times so that the image file contents are not clearly visible the attacker when the file is uploaded on drive.

Once the file is encrypted, the user will be redirected to upload page where the user must authenticate himself by providing his drive login credentials and once the user is identified the save button will be provided dynamically and by clicking on that button it pop-ups the drive window where you can select the folder where you want the save the encrypted file and once it is done, the file will be uploaded to drive successfully and the user can view the uploaded files by clicking the 'View-Drive' link available on our application. On click of that link the files uploaded on drive will be shown with their file ids on our application.

To download the file the user can click 'Download-Drive' link available and enter the file id available in view drive link and user can download file from drive by authenticating himself and the image file downloaded will show in the local system. Now, user can browse the downloaded file from local system on to our application and click the decrypt button and image will be decrypted by using the same logic that was used for encryption and save into directory and image path will be provided to user to view the file at that directory.

The following images shows the encryption of original image and the decryption of cipher image

Figure 4 – (left)Original Image (right)Encrypted Image

Efforts Shared

Gagan Solur Venkatesh

- Backend for Encryption and Decryption.
- Authenticating user before file uploading.
- Uploading cipher image to cloud.

Vishwanath Murundi Dhananjaya Murthy

- Frontend for Encryption and Decryption.
- Showing all image files uploaded to cloud with file ids by authenticating user.
- Downloading cipher image from cloud.

Conclusions and Future Work

We have implemented the new image encryption algorithm which had stated in "Thumbnail Preserving Encryption for JPEG" paper for protecting the images stored in the cloud, but still change in implementation, we tried same algorithm on BMP images. Which provides better privacy for the images and allows the cloud service which are not trustable to reconstruct a thumbnail of an encrypted photo, which is not a good resolution, but won't allow to reconstruct the high definition of the original photo.

"The original paper stated the that they make the attackers much more difficult, by applying additional transformations to modify the pixel values after sharing. The key property for these

transformations is that they must (approximately) preserve the average value of the pixels in the block otherwise the encryption will no longer be thumbnail-preserving. To achieve reasonable compression and high quality on natural images, we also approximately preserve the standard deviation of the pixels in each block" [5]. We will try to implement our project based on above statements.

References

[1] Andy Greenberg. The police tool that pervs use to steal nude pics from Apple's iCloud. Wired, September 2014.

[2] Somini Sengupta and Kevin J. O'Brien. Facebook can ID faces, but using them grows tricky. The New York Times, September 2012.

[3] Matt Tierney, Ian Spiro, Christoph Bregler, and Lakshminarayanan Subramanian. Cryptagram: Photo privacy for online social media. In Proceedings of the First ACM Conference on Online Social Networks, COSN '13, pages 75-88. ACM, 2013.

[4] Moo-Ryong Ra, Ramesh Govindan, and Antonio Ortega. P3: Toward privacy-preserving photo sharing. In NSDI, pages 515-528, 2013.

[5] Thumbnail Preserving Encryption for JPEG http://dl.acm.org/citation.cfm?id=2756618

[6] Ronald Aylmer Fisher and Frank Yates. Statistical tables for biological, agricultural and medical research. 3rd edition, 1949.