Marie Sklodowska Curie,

Research and Innovation Staff

Exchange (RISE)

Ref. Ares(2018)4606221 - 07/09/2018

European Commission

Horizon 2020
European Union funding
for Research & Innovation

ENhancing seCurity and privAcy in the Social wEb: a user-centered approach for the protection of minors

ENCASE
ENhancing seCurity and privAcy in the Social wEb

WP6 - Sensitive content detection and protection

Deliverable D6.2: "Development of content protection techniques that use steganography, encryption and watermarking"

Editor(s):	Zenonas Theodosiou (SGX)
Author(s):	Zenonas Theodosiou (SGX), Christos Mourouzis (CyRIC), Gabriel Emile Hine (ROMA3), Andreas Pafitis (SGX)
Dissemination Level:	Public
Nature:	Demonstrator
Version:	0.2

PROPRIETARY RIGHTS STATEMENT
This document contains information, which is proprietary to the ENCASE Consortium. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to any third party, in whole or in parts, except with prior written consent of the ENCASE consortium.

ENCASE Project Profile

Contract Number	691025
Acronym	ENCASE
Title	ENhancing seCurity and privacy in the Social wEb: a user-centered approach for the protection of minors
Start Date	Jan 1st, 2016
Duration	48 Months

Partners

Cyprus University of Technology	Cyprus University of Technology	Cyprus
UCL	University College London	United Kingdom
ARISTOTLE UNIVERSITY OF THESSALONIKI	Aristotle University	Greece
ROMA TRE UNIVERSITÀ DEGLI STUDI	Università degli Studi Roma Tre	Italy
Telefonica Telefónica Investigación y Desarrollo	Telefonica Investigacion Y Desarrollo SA	Spain
SignalGeneriX ADVANCED SIGNAL SOLUTIONS	SignalGenerix Ltd	Cyprus
CyRIC	Cyprus Research and Innovation Center, Ltd	Cyprus
LS TECH	L S Tech	Spain

Document History

AUTHORS

(SGX)	Zenonas Theodosiou
(CYRIC)	Christos Mourouzis
(ROMA3)	Gabriel Emile Hine
(SGX)	Andreas Pafitis

VERSIONS

Version	Date	Author	Remarks
0.1	08.08.2018	SGX	Initial partial draft
0.2	05.09.2018	SGX	Corrections and finalization of the deliverable

Executive Summary

OSNs give to users the opportunity to develop relationships among themselves and exchange information related to their professional or private life. The protection of the sensitive information, in a transparent to the user way, is of utmost importance. While non-visual content can be protected using the existing encryption methods, in the case of visual content transparent protection is not obvious. ENCASE aims to develop a web-browser add-on where only the authorized users can see specific content (visual and/or non-visual) without noticing that it is encrypted, while unauthorized users won’t be able to see it in case they legally gain access.

This deliverable, D6.2 - “Development of content protection techniques that use steganography encryption and watermarking” covers the work conducted in the framework of Task T6.2 - “Steganography and digital watermarking” and Task T6.3 - “Group encryption and attribute-based encryption”. The aforementioned two tasks are dedicated to the protection of sensitive content using steganography, watermarking and encryption techniques to sensitive content. The objectives that covered through these tasks are as follow:

a) The development of steganography-related techniques that enable users to specify groups of people that can view certain sensitive content;
b) The development of digital watermarking techniques on sensitive content so that in the event of unauthorized leakage, the culpable parties can be identified and sanctioned; and
c) The development of cryptographic techniques that allows users to specify groups of people that can view certain sensitive content using homomorphic, group and attribute-based encryption.

This document describes in detail, and demonstrates the implemented techniques for the protection of sensitive content, which allow users to specify the groups of people that can view certain sensitive content.

Table of Contents

Executive Summary 4
List of Figures 6
1. Introduction 7
2. Steganography 9
2.1. Image Steganography 10
2.2.1 Least Significant Bit (LSB) Steganography....................10
2.2. Image Steganography in Social Media....................11
2.3. Methodology 12
2.4. Use Case....................14
2.5. Examples....................14
3. Digital Watermarking....................16
3.1. Methodology 17
3.1. Use case 18
3.2. Example....................19
4. Cryptography 20
4.1. Encryption Types – Schemes 21
4.2. Identity-Based Encryption 23
4.3. Attribute-Based Encryption 25
4.3.1. Ciphertext Policy Attribute-Based Encryption (CP – ABE) 25
5. Key Policy Attribute-Based Encryption (KP – ABE) 26
5.1. Encryption and Decryption on a Social Media Platform....................28
5.2. Development of CP-ABE algorithm....................29
6. Example....................30
6.1. Demonstration of CP-ABE algorithm in OSN framework....................32
7. Conclusion....................35
8. References 36

List of Figures

Figure I: Steganography example: (a) The input image, (b) the image with a hidden message (“Hello World”). 9
Figure II: Steganography Classifications [5] 10
Figure III: Proposed steganography task architecture 13
Figure IV: Hiding an Image in a cover image 15
Figure V: Hiding text in a cover image 16
Figure VI: Visible watermarking. The image was downloaded from Pexels [19] 16
Figure VII: Proposed watermarking task architecture 18
Figure VIII: Image Watermarking example 19
Figure IX: Plaintext and Ciphertext 20
Figure X: Image Plaintext and Ciphertext 20
Figure XI: - Symmetric Encryption - Decryption [26] 22
Figure XII: Public Key Cryptography 22
Figure XIII: Asymmetric Encryption 23
Figure XIV: Setup and Key Generation 24
Figure XV: Encryption & Decryption Phase 24
Figure XVI: Access Structure T 25
Figure XVII: Setup & Key Generation – KP-ABE 27
Figure XVIII: Encryption - Decryption – KP-ABE 28
Figure XIX: Access Tree Policy Example 28
Figure XX: Set Attributes for Checking – Policy 30
Figure XXI: Random Generation of Public and Master Key for the PGP 30
Figure XXII: - Setup, Key Generation and Encryption phase 31
Figure XXIII: Failed Decryption | Successful Decryption 32
Figure XXIV: Generated files 32
Figure XXV: Demo – Login 33
Figure XXVI: Selection of users allowed to access the data 33
Figure XXVII: Choose file to access 34
Figure XXVIII: Decrypted file 34

1. Introduction

Social media platforms are internet-based form of communication which allows users to have conversations, post information on their profile, upload different type of media online and share their thoughts. Different forms of social media can be found including blogs, networking websites, photo-sharing websites, video-sharing, instant messaging, etc. Their use connects people who are miles away with their friends or family or even other people of the same interest and helps them among other things to develop new interests and broaden their knowledge. However, social media platforms can have their own drawbacks, especially for the sensitive people and especially the minors.

Online social networks constitute a breeding ground for the spread of malicious behavioral patterns, such as spamming, sybil attacks (forged profile identities), phishing and the even more dangerous statutory rapes and pedophile attacks. In this direction, the social network providers, the authorities as well as the scientific community are invested in analyzing social media data and identify or even predict such the behavioral patterns.

The online protection even though is a major issue and has been studied a lot, it can't be offered on a 100% rate. An enormous number of digital data are created and shared everyday through OSNs. Due to the ease of transmission of this content through the internet, it is dangerous and possible to have an unwanted leakage of this information to any not desirable group of people or even in public.

Steganography and digital watermarking techniques can be used to protect sensitive content which is uploaded and shared between OSNs users. These techniques are falling under the Information Hiding and Security area and their goal is to hide any information from any possible threats. Steganography is the art of concealing the existence of any digital information (cannot be observed) while Digital Watermarking physically covers any content so that is not visible to any unauthorized parties or to identify a digital media for copyright reasons.

Cryptographic allows users to specify groups of people that can view certain sensitive content on line. Attribute-based encryption is a new mean for encrypted access control. The main idea is that the user's private key and the cyphertext are associated with attributes, so that only users with specific attributes can decrypt data. Attributes may include age of the user, his/her type of subscription to the service, the kind of relationship with the data owner (e.g. colleague, relative, etc.).

The present document describes the efforts have been done in the framework of ENCASE project to protect the sensitive content that the minors post online either this is text, or an image or a video that they upload on social media, including steganography, watermarking and attribute-based encryption techniques.

Section 2 is dedicated to the Steganography techniques which have been developed in Task 6.2. The section starts with a presentation of the method and focuses on image steganography. Afterwards, the steps which were followed for the development of an image steganography technique are described, and some output examples (hiding image or text in an image) are shown. The Digital Watermarking is presented in Section 3, where the developed method is analysed and an example (cover a face after face detection) is reported. Section 4 deals with the attribute-based encryption schemes that were developed in the framework of Task 6.3. The historical background of the cryptographic techniques is described and then an implementation of encryption method is presented along with a demo that shows how this method can be embedded into an On-line Social Network so that a user can control who can access his/her content among his/her friends based on the type of relationship they have with him/her. Finally, conclusions regarding the performed activities are given in Section 5.

2. Steganography

Steganography is a method of the Information Hiding and Security area and it is used to prevent the detection of messages hidden in a cover. Its name is derived from Greek, which means "covered writing" (from Greek word "stegos" (cover) and "grafia" (writing)). This technique has a very long history since it was firstly used in 440 BC, as Herodotus mentions two examples in his "Histories" [1], [2].

In its digital form, steganography is used to hide any digital data in a digital cover (multimedia files). The main goal of steganography, in contrast with other methods (cryptography, watermarking) in the area, is to conceal the existence of the embedded data instead of encoding or covering the data [3]. Namely, this technique is used for secure communication in such a way that a message is not actually visible to the observer (Figure I). The receiver of a Stego data would only be able to extract the hidden message if he/she knows that the message exists and the way (algorithm) to reveal it.

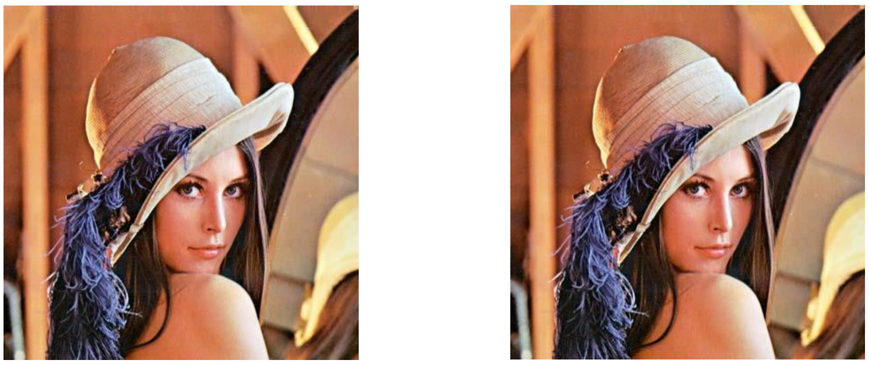


Figure I: Steganography example: (a) The input image, (b) the image with a hidden message ("Hello World").

Steganography has many applications depending on the digital cover object [3][4]:

- Image Steganography: The digital message (text or file) is hidden in an image. The most common method to hide the information is to digitally manipulate the pixels of the cover image.
- Network Steganography: The cover media is a network protocol such as TCP, UDP, ICMP, IP etc. The message is hidden in unused bits used by a protocol.
- Video Steganography: Video is a combination of images (frames) that change in a specific frequency in time and give the perception of a moving image. Steganography can be applied by hiding the information in each of the images in video.
- Audio Steganography: The cover media of the hidden message is an audio file. Audio steganography uses digital formats such as WAVE, MIDI, AVI, MPEG, etc.

- Text Steganography: This technique uses number of tabs, white spaces, capital letters, etc., to hide the information in a text format.

2.1.Image Steganography

The main goal of Image Steganography is to hide any sensitive content (text or image) in an image so that the receiver of this image is not able to reveal this content if he/she is not authorized to do so.

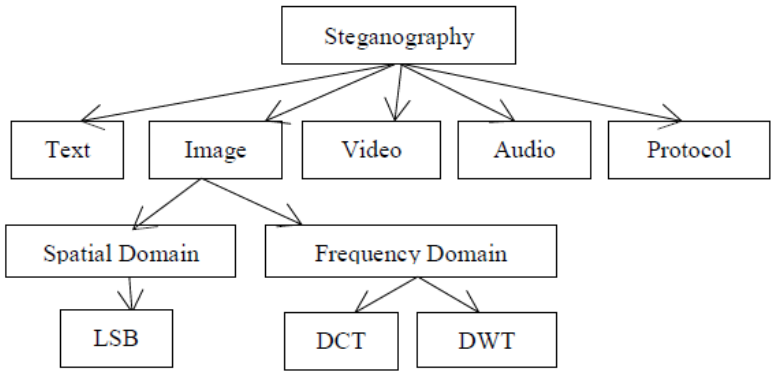


Figure II: Steganography Classifications [5]

Image Steganography can be performed in two domains; spatial and frequency domain (Figure II). In the spatial domain, the message (text, image, binary file) is hidden directly in the pixel values, while in the frequency domain the message is embedded in the frequency components after the image is transformed from the spatial domain to the frequency domain [6][7]. The most common methods are the Least Significant Bit (LSB) in the spatial domain and Discrete Cosine Transform (DCT)[8], or Discrete Wavelet Transform (DWT)[9] in the frequency domain [10]. In this task it is chosen to use LSB Steganography due to its simplicity, ability of storing a large amount of information and having less chance for degradation of the original image [4].

Image steganography can also be combined with cryptographic methods to add more security by ciphering the message with a private key [11]. This is preferable, in the occasion that even if someone manages to extract the hidden message either by using steganalysis or other methods then she/he will not be able to read it without providing the correct key.

2.2.1 Least Significant Bit (LSB) Steganography

One simple way to use steganography in the spatial domain is to encode the data of the cover image at the level of the Least Significant Bits (LSB) of the pixels. This technique is implemented by directly changing the LSB of a pixel with a bit of the hidden message. Recent researchers like Habes et al. [12] proposed new methods for hiding secret images in a cover image by using more bits (4 LSBs) to increase possible capacity. Some other methods use randomness to select which bit of a pixel will be changed to hold a bit of the secret message [13].

To understand how this technique works an example is described below:

In a 24-bit (RGB) image you can store up to 3 bits in a pixel. If the size of the image is 1024 x 768 pixels then there is a possibility of hiding a total of 2,359,296 bits (294,912 bytes) of information. Also, if the message is compressed before steganography is applied, then a large amount of information can be embedded into the carrier. It is also clear that the capacity of a cover image is increased according to its dimensions. To the human eye, the result (stego-image) will look identical to the original image as the change of the least significant bit its insignificant small to be observed.

E.g. the letter **E** can be hidden in three pixels (assuming no compression). The original values for 3 pixels (9 bytes) may be

(10110111 10001001 00001000) (00110010 10000000 10001011) (10101010 01010101 00100010)

The binary value for **B** is **01100101**. Inserting the binary value for E in the three pixels would result in

(1011011**0** 1000100**1** 0000100**1**) (0011001**0** 1000000**0** 1000101**1**) (1010101**0** 0101010**1** 00100010)

The bits in bold are the ones that were processed but the underlined bits are only the two that actually changed in the 8 bytes used. On average, LSB requires that only half the bits in an image be changed. You can hide data in the least and second least significant bits and still the human eye would not be able to detect it [13].

2.2. Image Steganography in Social Media

Since ENCASE deals with interaction of minors with other users in social media it is interesting to examine whether Image Steganography can be applied in images uploaded in social networks without losing their hidden content. Social networks are third-party content hosts and each one of them has its own policy on what can be uploaded on their servers. The most well-known social media are Facebook, Twitter and Instagram [14].

Each social media user is able to publish images to everyone or authorize a group of people (e.g. friends only) to see the published content. Most of the social media manipulates these published images by resizing, updating metadata, compressing, embedding or watermarking and for these reasons making it difficult to use well-known steganographic techniques on them. However, researchers have started proposing solutions on embedding hidden content in published images and "surviving" any image processing of a social network [5], [14], [15], [16], [17].

What would be interesting though is to check whether a stegoimage can survive in a private chat of a social network. A minor should be protected from any users that are not authorized through ENCASE platform to view any sensitive content, even in private. The proposed methodology in this document is tested mostly on Facebook Messenger with results showing that Image Steganography is possible in chats since a social network won't process any private data of a user and will just upload it on its server.

2.3.Methodology

The code that is developed for this task is implemented as an API call. This gives the flexibility to call the API with the appropriate inputs whenever is needed.

The service was written in Python 3.6 programming language using the following libraries and frameworks:

- *Stegano*: https://stegano.readthedocs.io/en/latest/, library for LSB steganography. It gives also the opportunity to use more advanced LSB techniques for extra security (set LSB according to generator algorithms such as Fibonacci, Slieve of Eratosthenes, etc.).
- *FLASK:* http://flask.pocoo.org/. Easy to use library for creating APIs.

Two API calls are implemented for the Steganography task. The service is implement on the localhost (http://localhost:5000/) and it was tested using POSTMAN [18] application to call the APIs.

- **SteganoEncryptAPI:** This API is responsible for hiding sensitive content in a static, unsuspected image (lock) using LSB Image steganography. It receives as inputs, the content to be hidden and outputs the steganoimage or an error. In the case of hiding an image, firstly the algorithm resizes the bigger dimension to the half of the cover image and also keeping the aspect ratio, and then converts (compress) it to JPEG format in order to ensure that it can fit in the cover image.
    - *URL:* http://localhost:5000/hideData

- *Method:* POST
- *Header*
    - Content-Type: multipart/form-data
- *Body (Inputs):*
    - Key: "image", Value: Image (JPEG, PNG, BMP etc) to be hidden
      or
    - Key: "text", Value: String to be hidden
- *Outputs*:
    - SteganoImage (PNG)
    - Error (JSON)

- **SteganoDecryptAPI:** This API is responsible for revealing the hidden content in an image.
    - *URL:* http://localhost:5000/unhideData
    - *Method:* POST
    - *Header*
        - Content-Type: multipart/form-data
    - *Inputs:*
        - Key: "image", Value: SteganoImage (PNG)
    - *Outputs*:
        - Revealed Image (JPEG)
        - Revealed Text (JSON)
        - Error (JSON)

The proposed Architecture is shown in Figure III and is described in the next section.

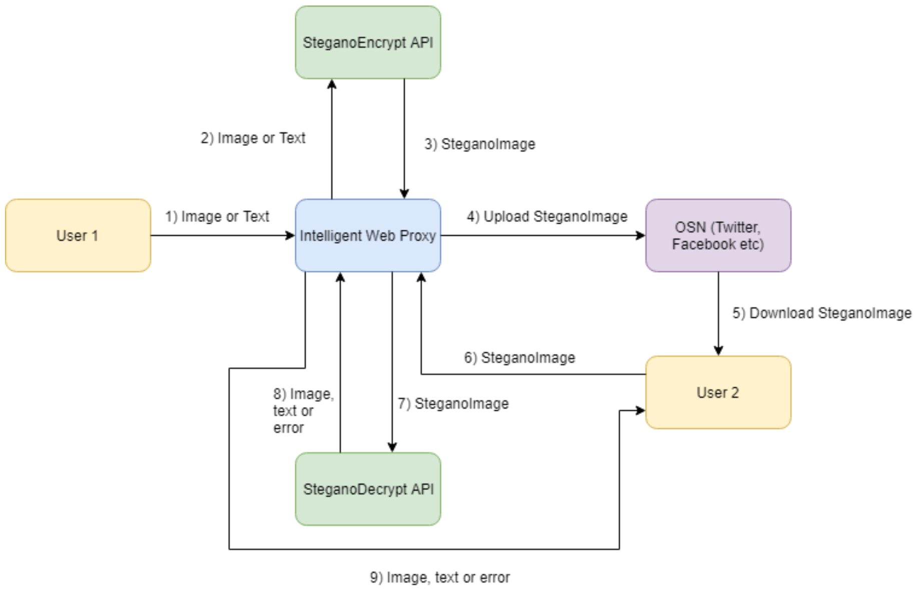


Figure III: Proposed steganography task architecture

2.4.Use Case

1) **User 1** selects to send a sensitive *image or text* to **User 2** using ENCASE. The *image or text* is sent to the **Intelligent Web Proxy (IWP)**.
2) **IWP** makes a post request to the **SteganoEncrypt API**. *Input* of the API is the *sensitive data (image or text)*.
3) **SteganoEncrypt API** hides the sensitive data in a standard static image (e.g. a lock, or an ENCASE logo). The *output* of the API (*SteganoImage*) is sent to **IWP.**
4) **IWP** posts the *SteganoImage* to the Online Social Network (**OSN**).
5) *SteganoImage* is downloaded to **User 2**.
6) If **User 2** uses ENCASE, then he/her right clicks *SteganoImage* and clicks “Show Original”. *SteganoImage* is sent to **IWP**. Note: if **User 2** does not use ENCASE then he is not able to decrypt *SteganoImage*.
7) **IWP** makes a post request to the **SteganoDecrypt API**. *Inputs* of the API is the *SteganoImage*.
8) **SteganoDecrypt API** extracts the hidden content. If the process is successful, then the *output* of the API is the *revealed sensitive data*. Otherwise the output is an *error*.
9) **User 2** receives the output of the API.

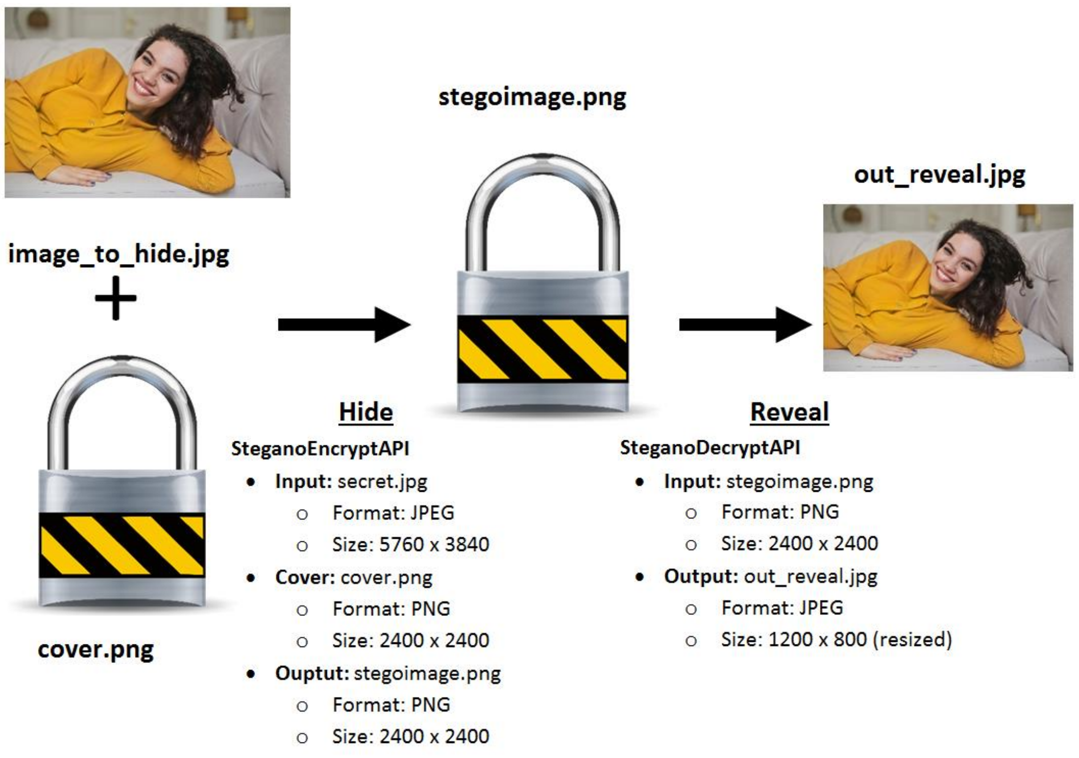
2.5.Examples

The code was tested by hiding different images or text in a static cover image (lock). This cover image is only used for demo purposes. It could be changed to any other cover image according to the requirements of ENCASE.

The outputs (stegoimages) of SteganoEncryptAPI were also uploaded on Facebook messenger. After downloading the images from the chat, the images were successfully decrypted (their content was revealed) by calling the SteganoDecryptAPI.

Hiding an Image

The image of the following example (Figure IV) was downloaded from Pexels [19] which is a website with Royalty-free licence policy. That means that images hosted on Pexels can be used for free for commercial and non-commercial use (https://www.pexels.com/photo-license/).



SteganoEncryptAPI

- **Input:** secret.jpg
  - Format: JPEG
  - Size: 5760 x 3840
- **Cover:** cover.png
  - Format: PNG
  - Size: 2400 x 2400
- **Ouptut:** stegoimage.png
  - Format: PNG
  - Size: 2400 x 2400

SteganoDecryptAPI

- **Input:** stegoimage.png
  - Format: PNG
  - Size: 2400 x 2400
- **Output:** out_reveal.jpg
  - Format: JPEG
  - Size: 1200 x 800 (resized)

Figure IV: Hiding an Image in a cover image

Hiding Text

The image of the following example (Figure V) was downloaded from Pexels [19] which is a website with Royalty-free licence policy. That means that images hosted on Pexels can be used for free for commercial and non-commercial use (https://www.pexels.com/photo-license/).

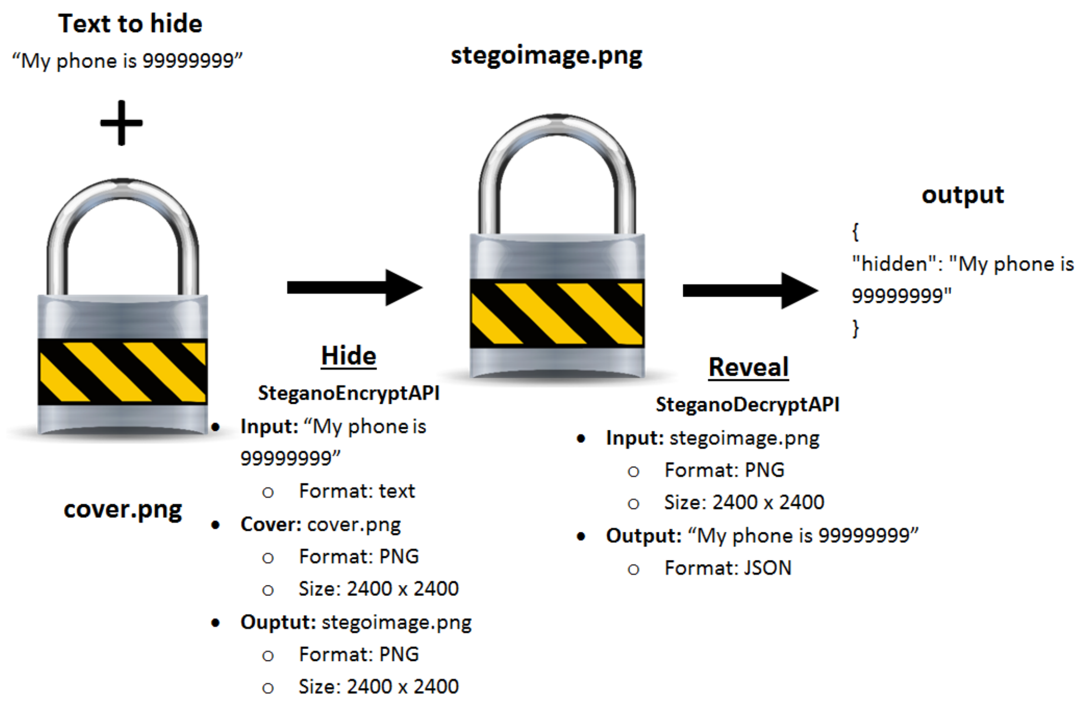


Figure V: Hiding text in a cover image

3. Digital Watermarking

Another technology which is closely related to Steganography and belongs to the Information Hiding and Security field is the Digital Watermarking. While Steganography aims to hide secretly any information in a cover media, Digital Watermarking’s main goal is to cover the object itself. It is divided in two main categories; visible and invisible. Visible watermarking (Figure VI) is the method of covering a media with some content (e.g. logo, text) which shows the owner of the media or covers sensitive information [20]. Invisible watermarking is digitally changing the digital form of a media in a unique way to digitally fingerprint (sign) it [21]. Watermarking is mostly used for copyright reasons, content authentication and protection [22].

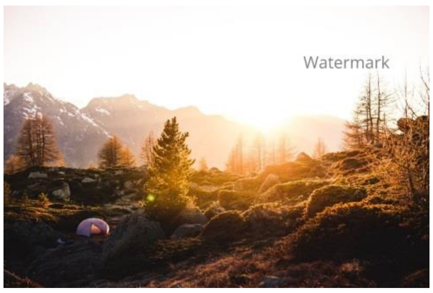


Figure VI: Visible watermarking. The image was downloaded from Pexels [19]

The use of watermarking in this task aims to cover any sensitive data (face, skin, nudity, etc.) of a user image so that this information cannot be visible to another user. Briefly, when ENCASE detects any inappropriate information on image this image should be processed and watermarked to cover and protect the minor for showing this sensitive content. Thus, visible watermarking will be used for this task.

3.1.Methodology

The code that developed for this task is implemented as an API call. This gives the flexibility to call the API with the appropriate inputs whenever is needed.

The service is written in Python 3.6 programming language using the following recipes and frameworks:

- Watermark with PIL: http://code.activestate.com/recipes/362879/, recipe to apply a watermark to an image using Python Image Library (https://pillow.readthedocs.io/en/5.2.x/).
- FLASK: http://flask.pocoo.org/. Easy to use library for creating APIs.

One API call is implemented for the digital watermarking. The service is implemented on the localhost (http://localhost:5000/) and it was tested using POSTMAN [18] application to call the API.

- **WatermarkAPI:** This API is responsible for covering sensitive content in an image, according to the coordinates (x1, x2, y1, y2) of sensitive content included in that image. The input is an image and a JSON string containing the coordinates (as a JSON array) that indicate the position-s of the content that is needed to be covered. The output is the watermarked image.
    - *URL:* http://localhost:5000/watermark
    - *Method:* POST
    - *Header*
        - Content-Type: multipart/form-data
    - *Body (Inputs):*
        - Key: “image”, Value: Image (JPEG, PNG, BMP etc)
          or
        - Key: “coordinates”, Value: JSON string:
          {"coordinates":[[x1,x2,y1,y2],[x1,x2,y1,y2], …]}
    - *Outputs*:
        - Watermarked Image (PNG)
        - Error (JSON)

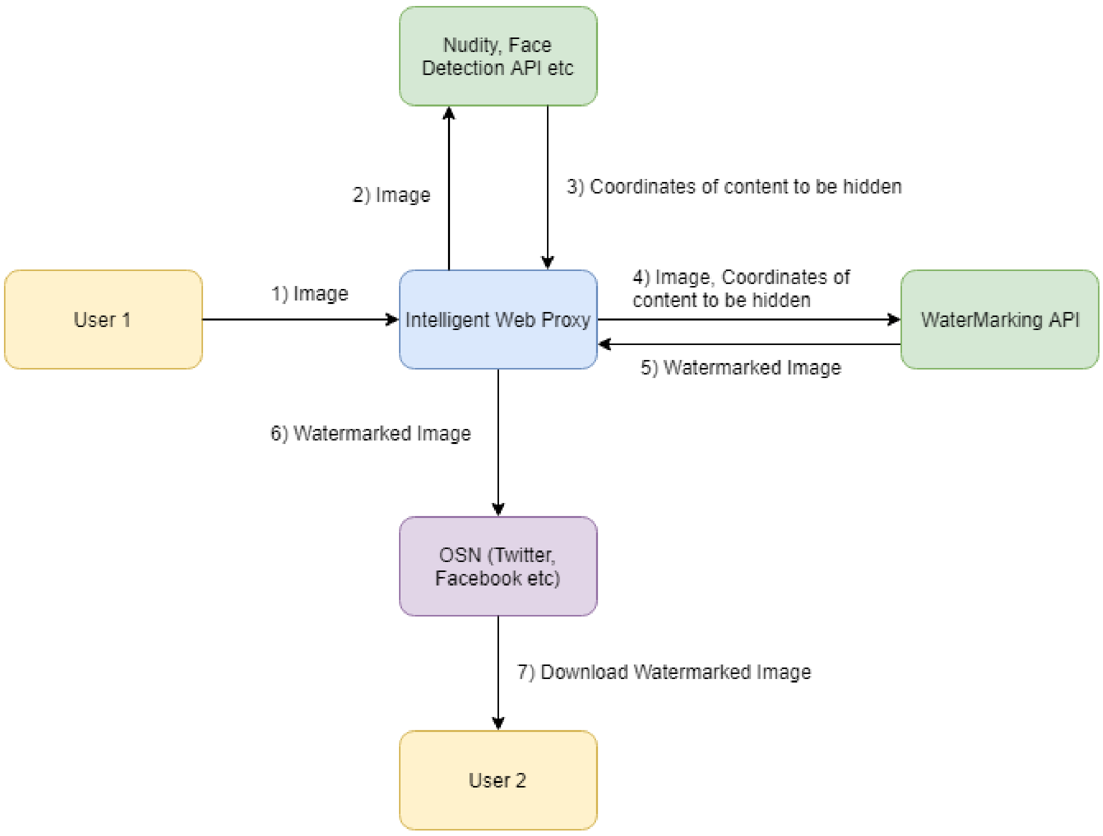

Nudity, Face Detection API etc
2) Image
3) Coordinates of content to be hidden
User 1
1) Image
Intelligent Web Proxy
4) Image, Coordinates of content to be hidden
WaterMarking API
5) Watermarked Image
6) Watermarked Image
OSN (Twitter, Facebook etc)
7) Download Watermarked Image
User 2

Figure VII: Proposed watermarking task architecture

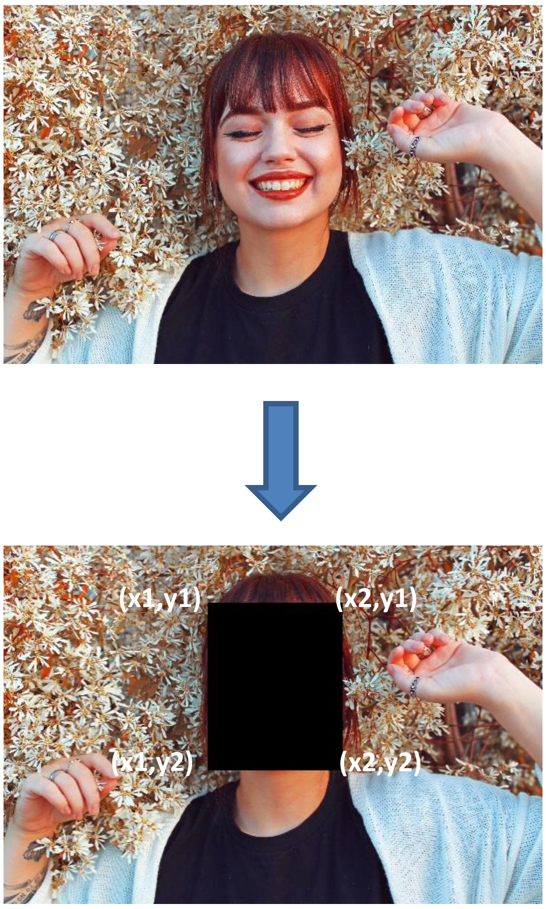
3.1.Use case

The proposed watermarking task architecture is shown in Figure VII and is described in the following steps:

1) **User 1** selects to send a sensitive *image* to **User 2** using ENCASE. The *image* is sent to **Intelligent Web Proxy (IWP)**.
2) **IWP** calls a sensitive content detection API.
3) Sensitive content is detected and **IWP** is informed about the level of confidence (e.g. probability of nudity in image), position (coordinates) of sensitive content in the image.
4) **IWP** calls the **WaterMarkingAPI**. *Inputs are the image and the position of the content needed to be covered.*
5) **WaterMarkingAPI** sends back to **IWP** its output (Watermarked Image).
6) **IWP** sends the WaterMarked Image to an OSN.
7) **User 2** receives the WaterMarked Image.

3.2.Example

The code was tested using SightEngine [23] to detect faces on images. This tool allows detection of sensitive content (nudity, minors, weapons, alcohol, drugs, face, etc.) in images or videos by calling simple APIs. For testing purposes, the face detection API of SightEngine was used. The image of the following example (Figure VIII) was downloaded from Pexels [19] which is a website with royalty-free licence policy. That means that images hosted on Pexels can be used for free for commercial and non-commercial use (https://www.pexels.com/photo-license/).



Input Image
Format: JPEG
Size: 5184 x 3456

SightEngine
Face detected between pixels
x1=1961
x2=3265
y1=552
y2=2168

WatermarkingAPI (Key: Value)

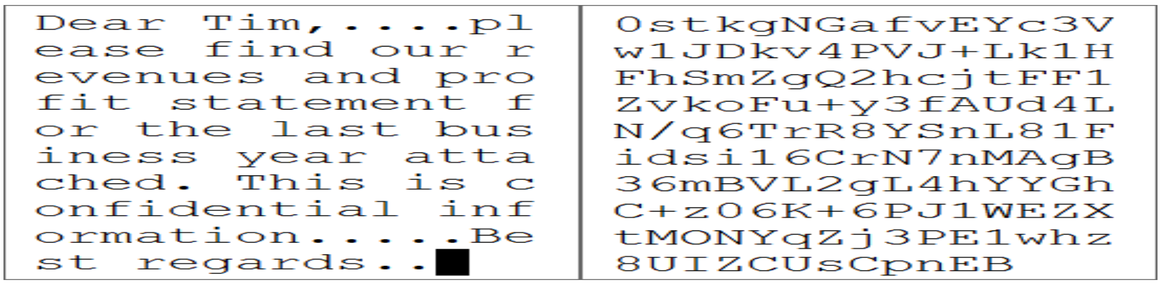
- **image: woman.jpg**
- **coordinates: {**"coordinates":[[1961,3265,552,2168]]}

Output Image
Format: JPEG
Size: 5184 x 3456

Figure VIII: Image Watermarking example

4. Cryptography

Cryptography is the method of secret writing. It is a major part of the Information security area and it is used to prevent unauthorized people to view a message or information that are not allowed to be seen. The origin of the root words is derived from two Greek words, “crypto” and “graphy”. The first root word stands for “hidden” or “secret” and the second one denotes the process or any form of writing, recording, describing. Cryptography is using mathematical principles to store and transmit any type of file in a particular form so only the intended people can read and process it. The first documented use of cryptography in writing dates back to the Egyptians around 1900 B.C. [24].



```
Dear Tim,....pl
ease find our r
evenues and pro
fit statement f
or the last bus
iness year atta
ched. This is c
onfidential inf
ormation.....Be
st regards..
```

```
OstkgNGafvEYc3V
w1JDkv4PVJ+Lk1H
FhSmZgQ2hcjtFF1
ZvkoFu+y3fAUd4L
N/q6TrR8YSnL81F
idsi16CrN7nMAgB
36mBVL2gL4hYYGh
C+z06K+6PJ1WEZX
tMONYqZj3PE1whz
8UIZCUsCpnEB
```

Figure IX: Plaintext and Ciphertext

In the days of written communication, many different types of cryptography were developed that involved some form of substitution or transposition of the alphabetical letters. The first method was substituting each letter of the initial message with a different one and the second method was repositioning the letters in such way that was scrabbling the information. During the digital age though and the widely spreading of computer networking, communication and sharing of data over untrusted mediums, a more strongly scientific approach was applied to develop cryptographic algorithms that are difficult to be broken from an adversary.

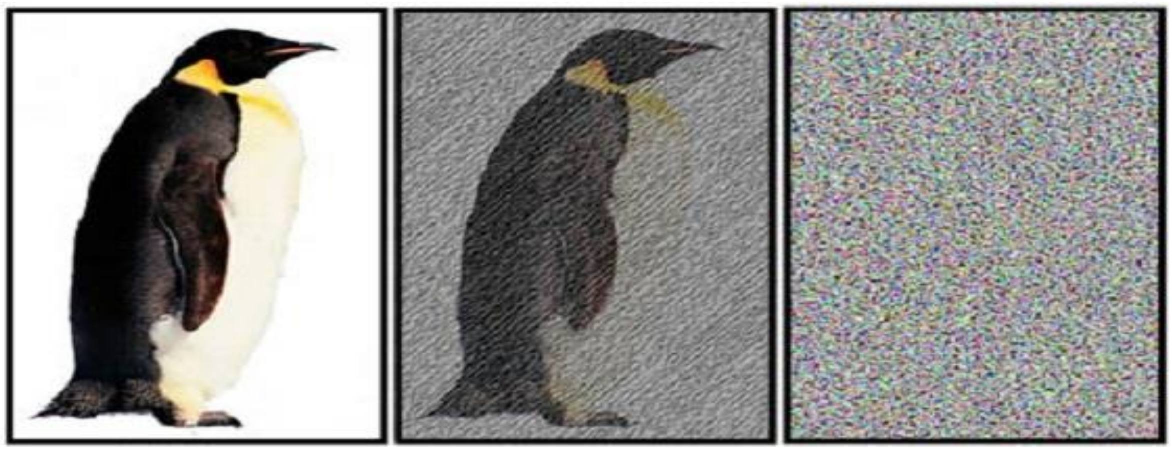


Figure X: Image Plaintext and Ciphertext

Until digital age, cryptography was used interchangeably and referred almost exclusively with the term encryption. Encryption is the process of converting any information or data into unreadable format. The original information is called plaintext and the encrypted result is called ciphertext (Figure IX, Figure X). Decryption is the reverse process of converting the ciphertext to a plaintext in order to be possible for the authorized person to read it. After the digital age advancement, cryptography is widely used as term for the science behind the process, not only from the implementation perspective, but for the mathematical theory and computer science perspective. The formal definition of a cryptographic process, called cryptosystem and it is the group of algorithms that will implement a security function. This cryptosystem is typically consisted from three algorithms: one for encryption, one for decryption and one for the generation of a pseudo-random key which will be used to scrabble the original message to ciphertext and vice versa. A cryptosystem is mathematically defined as (P, C, K, E, D) a finite list of the following properties:

- P is the set that its elements consist of plaintexts.
- C is a set with its elements consisting of ciphertexts.
- K is a set space that its elements consist of the keys.
- $E = \{E_k: k \in K\}$ is a set of functions that E_k: P -> C that its elements consist of encryption functions.
- $D = \{D_k: k \in K\}$ is a set of functions that D_k: C -> P that its elements consist of decryption functions.

The above definition is modified to adjust on the different types of encryption schemes. The mostly used and common cryptosystems nowadays are the symmetric-key and public-key cryptosystems. There are other ways to classify the cryptographic algorithms, but the main way is based on the number of cryptographic keys that are used to employ the encryption and decryption algorithms. The literature of cryptography often uses fictional characters for convenience and to aid comprehension, so this document will use them in order to help the better understanding. Usually, the name Alice “A” is the sender and the name Bob “B” is the receiver.

4.1.Encryption Types – Schemes

The first type of cryptographic algorithms called symmetric key cryptosystems [25], because of the employment of one single pseudo-random key in order for the encryption and decryption to happen. As Figure XI shows, the sender of the plaintext uses a secret shared key to encrypt it and the receiver applies the same key to decrypt the ciphertext and have access to the plaintext. This form of cryptography, it is one of the simplest types due this shared secret key. The secret shared key can be anything from a number, a word or a string of random letters and special characters. The main drawback of this type is if the shared key compromised or one of the people involved is untrusted it would easily lead to compromise of the encrypted information. Symmetric key algorithms are employed either as a block cipher or stream cipher.

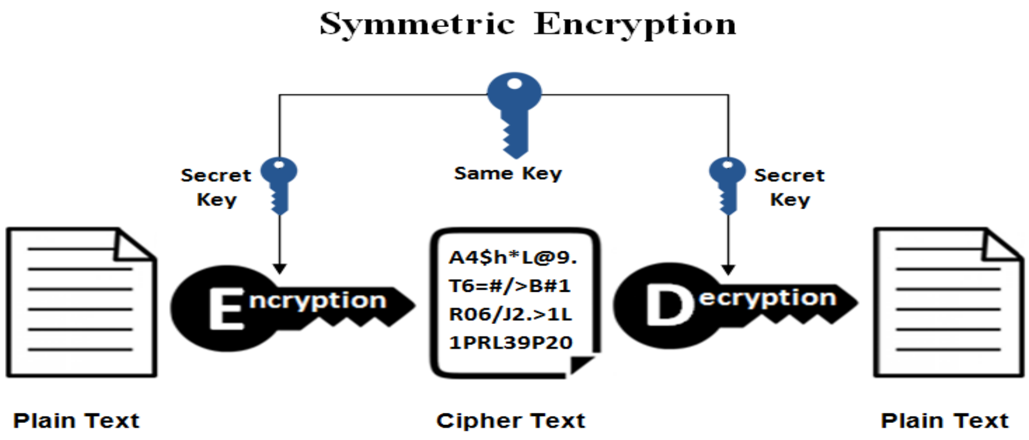


Figure XI: - Symmetric Encryption - Decryption [26]

The first one receives the input as a block of the plaintext and applies the encryption cipher as opposed to the stream cipher that receives as input each character individually. The mostly known block cipher algorithms are Advanced Encryption Standard (AES) and Data Encryption Standard (DES), even though the latter was withdrawn after the adoption of the first one. These two designs are chosen by the US government as a cryptography standard.

The second type of cryptographic algorithms called public key cryptography or asymmetric key cryptosystem. This cryptosystem is called asymmetric key in contrast of the symmetric key in order to set the distinguish between them. This system was a breakthrough for cryptography back in 1976 when Diffie-Hellman [26] proposed this type. The system uses two different keys, which are mathematically related and called – a public and a private key. The calculation of the private key from the public key is computationally infeasible that’s why it is considered one of the most secure systems. The so-called public key is freely distributed while the private one remains secret. The Figure XII shows how a public key cryptosystem works. The public key is used to encrypt the plaintext while it’s paired private or secret key is used for decryption.

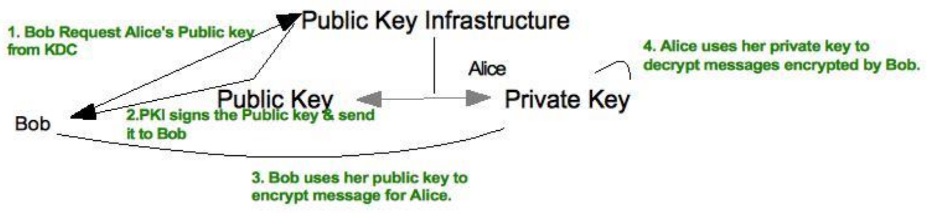

1. Bob Request Alice's Public key from KDC
Public Key Infrastructure
4. Alice uses her private key to decrypt messages encrypted by Bob.
Alice
Bob
Public Key
2.PKI signs the Public key & send it to Bob
Private Key
3. Bob uses her public key to encrypt message for Alice.

Figure XII: Public Key Cryptography

The keys are issued from a trusted system, called Key Distribution Center (KDC) that creates, stores and distributes a certificate which verifies that a public key belongs to a certain person. For example, Alice wants to send a message to Bob. She uses a large pseudo-random number which will compute and derive a pair of keys, a public and a private one. The same applies for Bob, who as well, receives a pair of keys. Alice asks from KDC for the Bob's public key and the center will sign the public key and sends it to Alice. Then uses this public key to encrypt the message and sends it to Bob. In order for Bob, to read the message, will use his paired private key and decrypt the plaintext. This process provides authentication and non-repudiation to the two parties. Bob knows that Alice sent the message and Alice cannot deny having sent the message. Some of the most used public key algorithms are RSA and Diffie-Hellman key exchange scheme.

4.2.Identity-Based Encryption

In 1984, Shamir [27] presented his idea for a possible creation of a different public-key cryptosystem which will be based on unique information about the identity of an individual or an organization. Unique information would be the e-mail address thus will be possible to be used as a public key. Shamir was trying to achieve a public key infrastructure without the need of the users to request the public key before they can encrypt the plaintext. The trusted third party, key distribution center, it will be in place but with the need to verify and generate a private key for the receiver according the public information that the sender used.

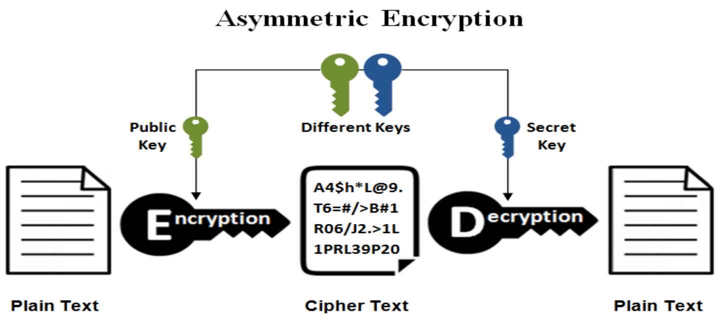


Figure XIII: Asymmetric Encryption

A concrete implementation for the identity-based cryptosystem appeared many years after Shamir's proposal. Sakai et al. [28] and Boneh & Franklin [29] proposed the first practical solution of this basing their solution on bilinear pairing. This scheme allows anyone to generate a public key by just applying a known identity value (e.g. e-mail address) of an individual or organization (Figure XIV). The corresponding private keys are generated from a trusted third-party authority – equal to key distribution center in public key cryptosystem, called Private key Generator (PKG). This happens on the setup step phase of the ID-based encryption.

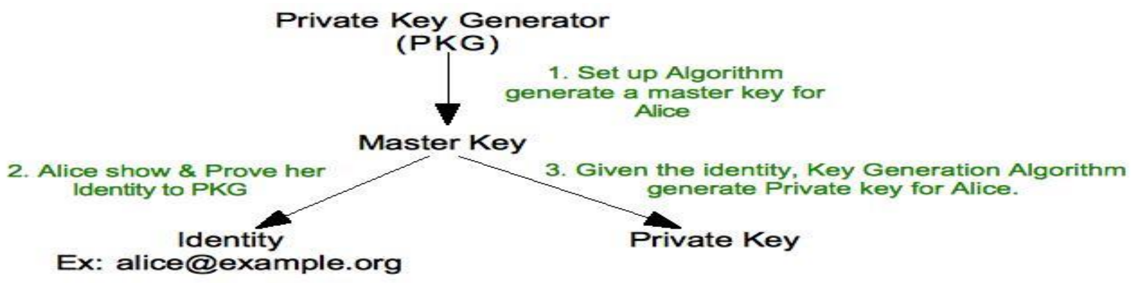

Private Key Generator (PKG)
1. Set up Algorithm generate a master key for Alice
Master Key
2. Alice show & Prove her Identity to PKG
3. Given the identity, Key Generation Algorithm generate Private key for Alice.
Identity Ex: alice@example.org
Private Key

Figure XIV: Setup and Key Generation

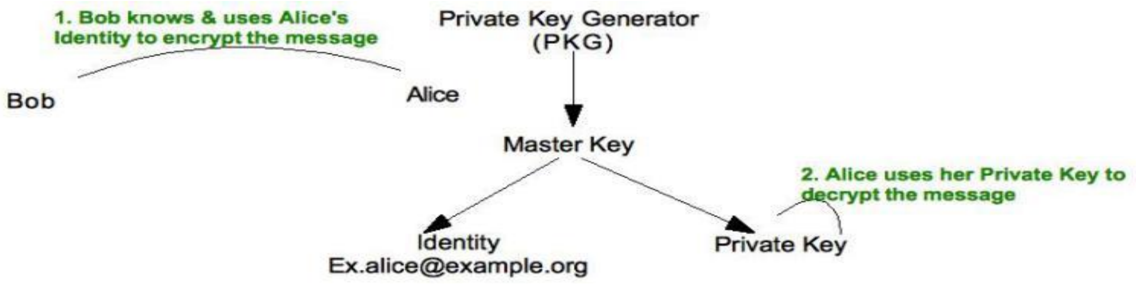

1. Bob knows & uses Alice's Identity to encrypt the message
Private Key Generator (PKG)
Bob
Alice
Master Key
2. Alice uses her Private Key to decrypt the message
Identity Ex.alice@example.org
Private Key

Figure XV: Encryption & Decryption Phase

The PKG publishes a master public key and retains a master private key. Alice uses the master public key and the identity value in order to compute a public key that will be used on the encryption phase. Before Alice receives her private key, she needs to authorize the PKG to use her identity value, since it is required for PKG to use in order to generate a private key for her using its own master private key. This whole process can be done off-line and it is one of the advantages of the identity-based cryptosystems. Similarly, Bob followed the same process and he, as well, has a private key issued from the PKG.

Both users have a private key that will be used to decrypt any plaintext that will receive, and it is encrypted with their identity value. For instance, Bob wants to send an encrypted message to Alice. As the Figure XV shows, during the third and fourth step, encryption and decryption phase, he uses Alice’s identity value (e.g. e-mail address) to encrypt the message. He sends it to Alice and she uses her identity value to authenticate herself and the private key which will decrypt the message.

4.3.Attribute-Based Encryption

Attribute-based encryption is a recent approach of Identity-based encryption. It takes the ID encryption a further step by defining the identity not just a unique information that an individual or organization possess but a set of attributes i.e. role within an organization, management level, department, etc. The decryption in a system like this is possible only if the receiver of the message matches the attributes of the ciphertext. The concept of attribute- based encryption (ABE) developed this idea further and presented two variations of ABE named ciphertext-policy attribute-based encryption (CP-ABE) [30] and key-policy attribute-based encryption (KP-ABE) [31].

4.3.1. Ciphertext Policy Attribute-Based Encryption (CP – ABE)

In Ciphertext policy attribute-based encryption [30], the user’s private key is linked with a set of attributes and in a ciphertext is specified an access policy over a universal group of attributes within a system. Any user, that wants to decrypt a ciphertext, it will be possible to do it if and only if his set of attributes satisfies that access policy. The policies are defined using mathematical compound statements using conjunctions, disjunctions and (k out of n) attributes have to be present. When the user encrypts a message, it needs to specify the threshold of the access structure of the attributes that wants the decrypting user to satisfy (Figure XVI). With ciphertext policy attribute-based encryption technique, encrypted data can be kept confidential and secure against collusion attacks.

Let us assume a universe of attributes defined as {A, B, C, D} and two users receive a private key to attributes {A, B} and {D} respectively. If a ciphertext, encrypted under the policy (A ∧ C) ∨ D, then the user with the key to attribute {D} will be able to decrypt it because it falls under the disjunction symbol (OR) but the user with the key to attributes {A, B} cannot decrypt it because the policy doesn’t cover its attributes.

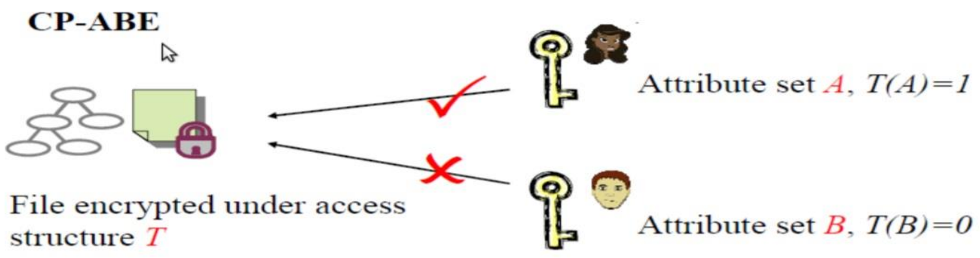

CP-ABE
File encrypted under access structure T
Attribute set A, $T(A)=1$
Attribute set B, $T(B)=0$

Figure XVI: Access Structure T

The ciphertext-policy ABE contains four steps of algorithms, which are the same as the identity-based encryption scheme but only with a slightly altered implementation. The four steps are the following:

- **Setup:** The input of this algorithm is a parameter K and returns a pair of public key PK and master secret key MK. In order for the sender to encrypt the message, he uses the PK. The MK is used by the trusted authority to generate private keys for each user.

- **Encryption:** In order for a message to be encrypted, the input is the public key PK, the message (information) M and access structure such as the figure below. The ciphertext CT is generated.

- **Key Generation:** The key is generated by receiving as input the attributes that are linked to the user along with the master secret key MK that was generated on the setup phase. The output private secret key SK is the one that enables the user to decrypt any message under the universe access tree structure T if and only if the matches this structure T.

- **Decryption:** The input is the CT and the user’s secret key. The message is showed if and only if the access structure T links with the ciphertext CT.

Advantages & Limitations

As every cryptographic algorithm ciphertext policy attribute-based has some advantages and some limitations that we need to take in account. One of the key elements is that it allows authorization instantly because the encrypted data contains authorization in the associated policy and only the people that are authorized with these attributes can access it. Furthermore, creation of a new user is possible due to the creation of the private keys after the access structure/policy took place, so there is not a need to specify a number of users prior.

On the other hand, it may be not possible for this scheme to satisfy enterprise requirements since is not flexible and efficient on specifying policies and user management attributes. Additionally, decryption keys only support user attributes to be logically organized as a single set and can only use all possible combinations in their private keys to fulfill access policies.

5. Key Policy Attribute-Based Encryption (KP – ABE)

In ciphertext policy attributed based scheme the access control and policy it is more specific and tailored to the file itself wanted to enable more general access control so they proposed a different approach for an attribute-based system [31]. The system it is a modified model of the first edition of ABE. The scheme requires that the attribute policies are associated with the key and the data (information) associated with attributes. The user that encrypts the data is linked to them with a set of attributes by encrypting them with a public key. The users that are allowed to have access to these data are assigned themselves with an access tree that covers the data attributes. The nodes of their access tree are the threshold gates.

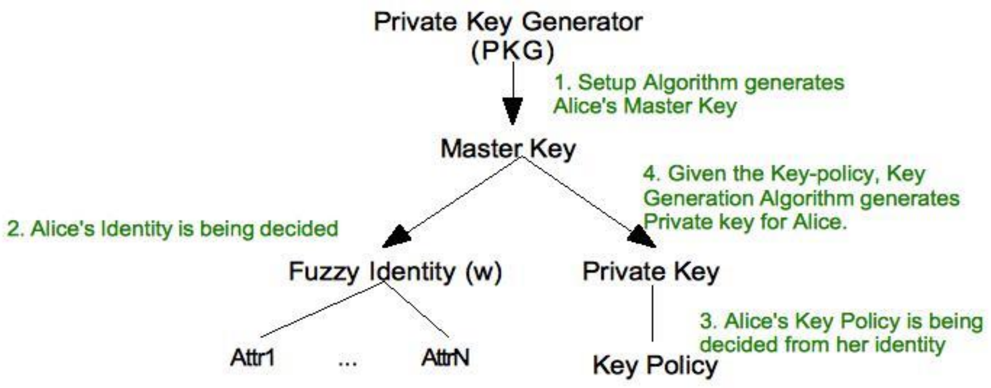

Private Key Generator (PKG)
1. Setup Algorithm generates Alice's Master Key
Master Key
4. Given the Key-policy, Key Generation Algorithm generates Private key for Alice.
2. Alice's Identity is being decided
Fuzzy Identity (w)
Private Key
3. Alice's Key Policy is being decided from her identity
Attr1
...
AttrN
Key Policy

Figure XVII: Setup & Key Generation – KP-ABE

The key-policy ABE contains four steps of algorithms, which are the same as the identity-based encryption scheme but only with a slightly altered implementation (Figure XVII, Figure XVIII) . The four steps are the following:

- **Setup:** The private key generator (PKG) generates a paired public key PK and a master secret key. The PK will be used from the sender to encrypt the message. MK will generate users’ private keys which are only known to the PKG.

- **Encryption:** The encryption takes as input the message (information) M, public key PK and a set of attributes. This will derive the ciphertext E.

- **Key Generation:** The set of attributes of the user are decided in order to be used to generate a key policy for him. This key policy will be used along with the MK to generate the private key using the key generation algorithm. The private key permits the user to decrypt the message if the attributes match.

- **Decryption:** In order for the user to decrypt the message, the input will be the user’s private key for the access structure of the user and the ciphertext E, which was encrypted under a set of attributes. The message M will be revealed if and only the attribute set satisfies the user’s access structure T.

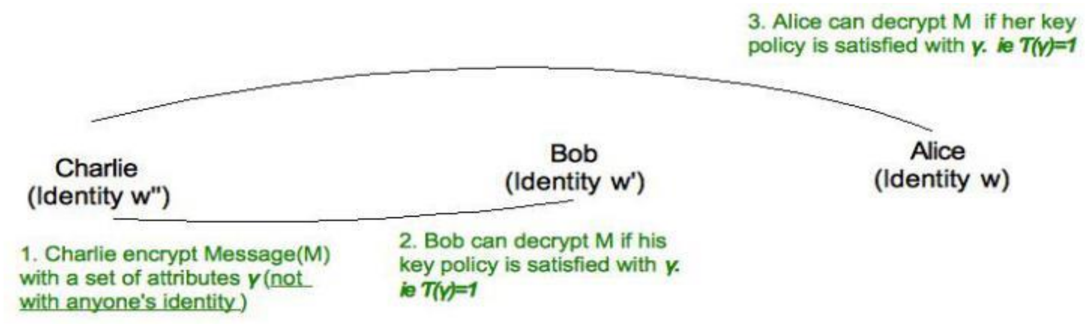

3. Alice can decrypt M if her key policy is satisfied with γ. ie T(γ)=1
Charlie (Identity w'')
Bob (Identity w')
Alice (Identity w)
1. Charlie encrypt Message(M) with a set of attributes γ (not with anyone's identity)
2. Bob can decrypt M if his key policy is satisfied with γ. ie T(γ)=1

Figure XVIII: Encryption - Decryption – KP-ABE

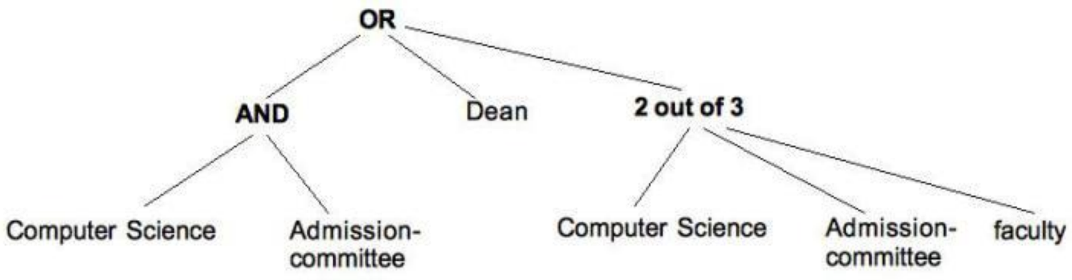


Figure XIX: Access Tree Policy Example

Let’s assume that Alice has the key policy that is shown in Figure XIX. A message M is encrypted by Charlie using an attribute set {“Computer Science”, “Admission committee”} and another one M1 with attribute set {“Computer Science”, “Program-committee”}. Alice can decrypt the M because her access tree and private key allows that, but the second message is not possible to be decrypted from her.

5.1.Encryption and Decryption on a Social Media Platform

ENCASE aims to be applied on a Social Media Platform and aspires to protect minors on them. Minors interact daily with many different people and depending on the platform that is being used, not only the known ones exist out there, it can be people they already know or people that hide their identity behind the umbrella of the World Wide Web. It will be interesting for the attributed-based encryption schemes, ciphertext-policy and key-policy, to be applied on a web browser giving the option to the owner of an information to encrypt it and allow access only to people that are specified by him. An interesting fact will be to achieve the encryption and decryption in private messages of Facebook and Twitter social platforms being the mostly used ones.

5.2. Development of CP-ABE algorithm

The source code implemented on NetBeans Integrated Development Environment (IDE). The source code is executed locally on the author’s laptop. The inputs are given locally, and the outputs are generated on the project’s folder. For the purposes of task 6.4, the source code will be transferred on the web browser by implementing API Calls which will notify the minor/user while being on social media platform to perform encryption when posting, uploading content online. This task is not responsible for the detection of sensitive content and will work along with the corresponding algorithms.

The source code was written with the Java programming language (JDK 8) and using the following libraries and packages:

Java Standard Library with sub-packages:

- Javax.crypto: https://docs.oracle.com/javase/7/docs/api/javax/crypto/package summary.html
- Java.security https://docs.oracle.com/javase/7/docs/api/java/security/package-summary.html
- Java.io: https://docs.oracle.com/javase/7/docs/api/
- Java.util: https://docs.oracle.com/javase/7/docs/api/

Open-source Java Pairing-based cryptography library (JPBC):http://gas.dia.unisa.it/projects/jpbc/

Four main methods are defined in order to cover the four stages of the attribute-based encryption schemes as followed:

Setup: This method is responsible for the setup phase of the scheme. It takes as inputs the public file and the master key file (path of the file) which was generated before using a security parameter λ.

- `public void setup (String publicfile, String masterKeyfile)`

KeyGeneration: This method is responsible for the key generation phase of the scheme. It takes as inputs the public file, the path where the master key file is the private file which defines the path of the private key file and the attributesString which defines the private attributes. The attributes can have the following presentation:

Panagiotis_private_string = “Encase school_name age_group etc..”

Antonis_private_attribute_string = “Encase family_relationship age_group etc..”

The order doesn’t make any difference since the source code parses the keywords.

- `public void keyGeneration (String publicfile, String privatefile, String masterkeyfile, String attributesString)`

Encryption: This method is responsible for the encryption phase of the scheme. It takes as inputs the public key file, the policy for the access structure tree, the input file for the file to be encrypted and then encrypted file is the result. The policy has the following form:

“school_name age_group 2of2 first_name last_name 2of3 or family_relationship 1of2” (example policy which will have more specific attributes). The above means that someone to decrypt the file needs to have at least 2 attributes from ((school_name age_group), first_name, last_name)) or (family_relationship)).

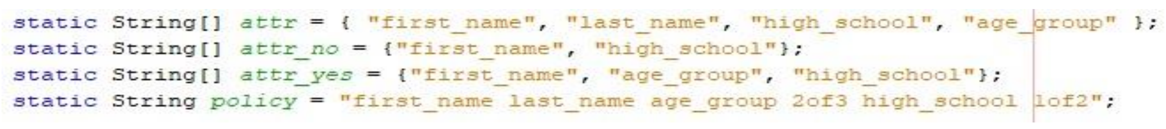
- `public void encryption (String publicfile, String policy, String fileForEncryption, String encryptionfile)`

Decryption: This method is responsible for the decryption phase of the scheme. It takes as inputs the public key file, the private key file, the encrypted file that was created before and generates a decrypted file same as the original.

- `public void decryption (String publicfile, String privatefile, String encryptionfile, String decryptionfile)`

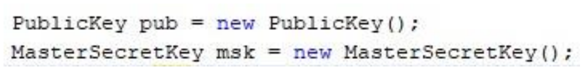
6. Example

The attribute-based ciphertext policy scheme was implemented using the java programming language. Figure XX shows the attributes stated for testing. The *attr* states 4 different elements which are the generic elements that will be used for the encryption and need to be a part of the key. The second (*attr_no*) and the third (*attr_yes*) arrays present two different people with different attributes. The last array (*policy*) states the policy that a person-user needs to verify in order to access the file. Figure XXI shows the constructor that calls the classes public key and master key in order to generate a new pair of two keys that will be used for the encryption.



```
static String[] attr = { "first_name", "last_name", "high_school", "age_group" };
static String[] attr_no = {"first_name", "high_school"};
static String[] attr_yes = {"first_name", "age_group", "high_school"};
static String policy = "first_name last_name age_group 2of3 high_school 1of2";
```

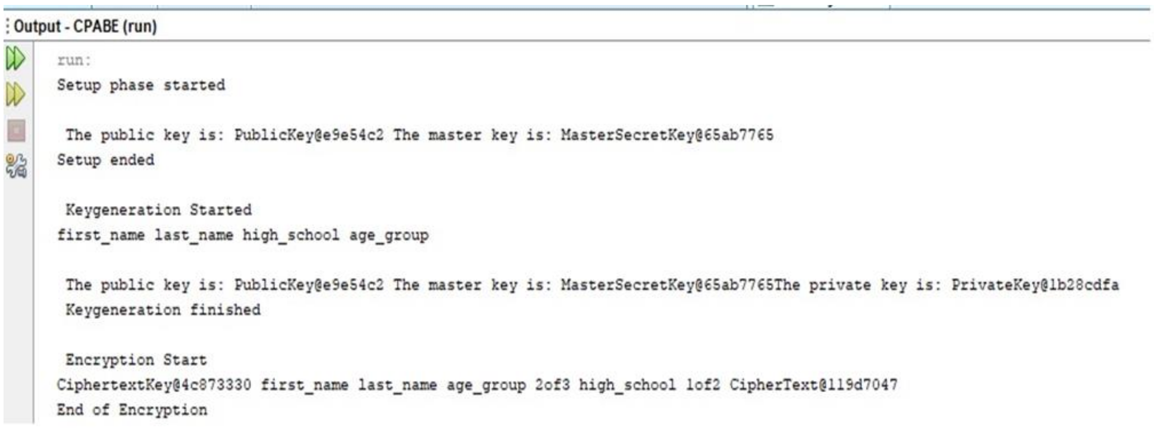
Figure XX: Set Attributes for Checking – Policy



```
PublicKey pub = new PublicKey();
MasterSecretKey msk = new MasterSecretKey();
```

Figure XXI: Random Generation of Public and Master Key for the PGP

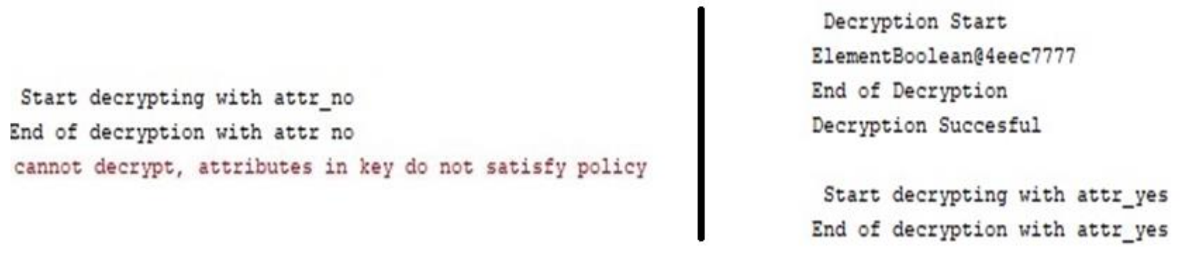
Figure XXII presents the relevant messages of execution: setup, key generation and encryption of the file. It also shows the public and master key that were generated and confirmed that are used for the generation of the private key on the key generation phase. The encryption phase shows the ciphertext’s id and the policy that is used to encrypt and produce the private key, as it is already described before.


Output - CPABE (run)

```
run:
Setup phase started

 The public key is: PublicKey@e9e54c2 The master key is: MasterSecretKey@65ab7765
Setup ended

 Keygeneration Started
first_name last_name high_school age_group

 The public key is: PublicKey@e9e54c2 The master key is: MasterSecretKey@65ab7765The private key is: PrivateKey@1b28cdfa
 Keygeneration finished

 Encryption Start
CiphertextKey@4c873330 first_name last_name age_group 2of3 high_school 1of2 CipherText@119d7047
End of Encryption
```

Figure XXII: - Setup, Key Generation and Encryption phase

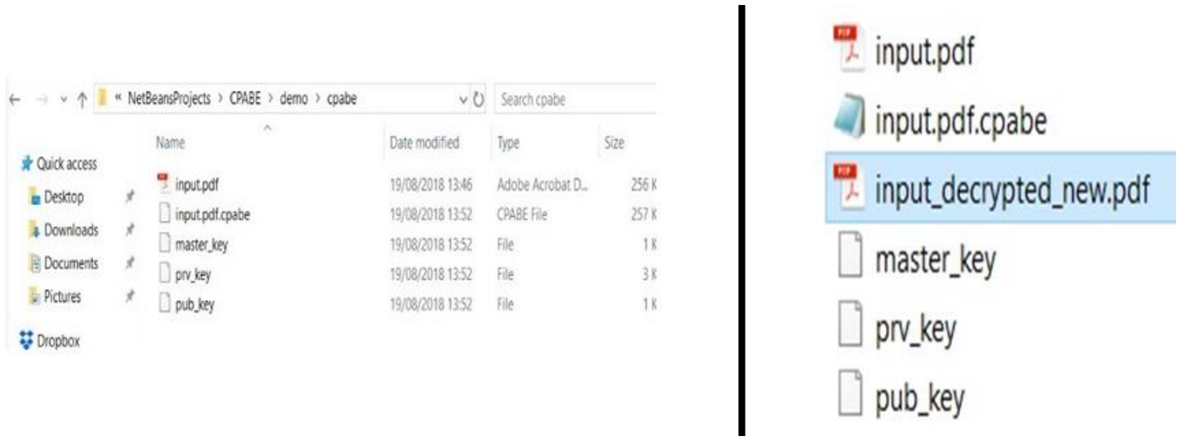
Figure XXIII shows side by side a failed and a successful decryption. The left part of the Figure shows that the user with attributes (*attr_no*) cannot decrypt the file and the attributes in key do not satisfy the policy. On the other hand, the decryption of the file with the user’s private key that has (*attr_yes*) has been successful and the file is accessible. Figure XXIV shows output folder on the left being the file still encrypted in an unknown format (.cpabe) and the generated master, private and public key files and on the right being the successful decryption with the file generated as an input_decrypted_new.pdf accessible to the user.



```
 Start decrypting with attr_no
End of decryption with attr no
cannot decrypt, attributes in key do not satisfy policy
```

```
 Decryption Start
ElementBoolean@4eec7777
End of Decryption
Decryption Succesful

 Start decrypting with attr_yes
End of decryption with attr_yes
```

Figure XXIII: Failed Decryption | Successful Decryption



Name	Date modified	Type	Size
input.pdf	19/08/2018 13:46	Adobe Acrobat D...	256 K
input.pdf.cpabe	19/08/2018 13:52	CPABE File	257 K
master_key	19/08/2018 13:52	File	1 K
prv_key	19/08/2018 13:52	File	3 K
pub_key	19/08/2018 13:52	File	1 K

input.pdf
input.pdf.cpabe
input_decrypted_new.pdf
master_key
prv_key
pub_key

Figure XXIV: Generated files

6.1. Demonstration of CP-ABE algorithm in OSN framework

A demo that demonstrates the benefits of Attribute Based Encryption techniques in the context of Online Social Networks was implemented. The demo was based on Cipher-policy ABE algorithms, exploiting both their effectiveness and simplicity for access control applications. For usability reasons, only access control policies were implemented based on OR operators, since security without usability leads to failure. If a user is asked to type complex logical constructs, such as [(friends AND colleagues) OR relatives], she/he would probably do not use such a service.

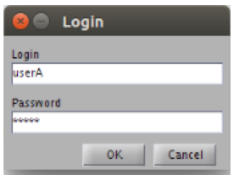
The GUI of the demo has been developed in Matlab under Ubuntu operating system. As core encryption functions, we used pyPEBEL [32], an open-source predicate-based encryption library.

Specifically, the demo simulates a very simple social network in which users share some data with other users. Each link between users is labeled with the kind of relationship they have. In the example we implemented the type of relationships we considered are: colleagues, friends, and relatives. These categories are used as attributes for an CP-ABE scheme.

Each user is described by a folder in which the following items are stored:

1. 'master-secret-key.msk' and 'public-parameters.mpk' that are create by pyCPABE-setup.py
2. a folder 'keys' containing the decrypting keys (created through pyCPABE-keygen.py) that other user may use to decrypt data:
    - colleagues.cpabe.dkey
    - friends.cpabe.dkey
    - relatives.cpabe.dkey
3. a text file 'friends' containing the list of friends of the user and the type of relationship (attribute) associated with each friend.
4. 'password.mat': a matlab file containing the password of the user.

To start the simulation, 'osn_simulation.m' must be run through Matlab. The GUI will ask to login with one of the existing accounts (Figure XXV).


Login

Login: userA

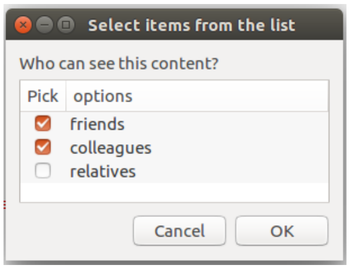
Password: *****

Figure XXV: Demo – Login

At this point the system allows to the following different actions:

1. Encrypt new data
2. View another user's content

1) If 'encrypt new data' is selected, the user is asked to select any content from his/her file system and choose which categories of users are allowed to access the data.


Select items from the list

Who can see this content?

Pick	options
✓	friends
✓	colleagues
☐	relatives

Figure XXVI: Selection of users allowed to access the data

The selected file is encrypted using the private master key and policy such like 'friends OR colleagues' shown in the paradigm depicted in Figure XXVI.

The encrypted file is stored in the user's folder with '.cbabe' extension.

2) If 'View another user's content' is selected, the user is asked to select any '.cbabe' file from the other users folder (Figure XXVII).

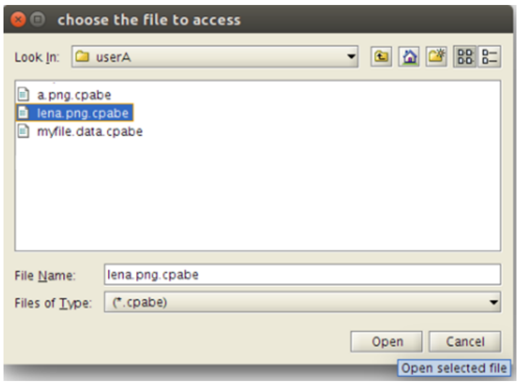


Figure XXVII: Choose file to access

Depending on the type of relationship the two users have, the corresponding decrypting key will be used to attempt to decrypt the content. If the decryption process is successful, the decrypted file is saved in the 'tmp-file' folder with the '.prime' extension (Figure XXVIII).

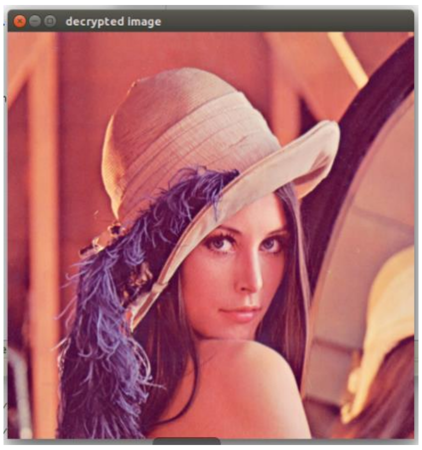


Figure XXVIII: Decrypted file

7. Conclusion

The present document provides the description of the activities carried out in the framework of Task 6.2 and Task 6.3 of the ENCASE project. The sensitive content which is uploaded and shared through OSNs should be protected from malicious activities, and users should be able to specify the groups of people that can view their sharing data.

The performed research has been focused on the development of image steganography, watermarking and cryptographic techniques as an effort to address this issue. The outcomes of the conducted research outline the importance of these techniques and their feasible integration in an OSN framework for the protection of minors.

8. References

[1] R. Das and T. Tuithung, "A review on 'A Novel Technique for Image Steganography Based on Block-DCT and Huffman Encoding,'" in *International Conference on Graphic and Image Processing (ICGIP 2012). Proceedings of the SPIE, Volume 8768, id. 87687D 5 pp. (2013).*, 2013, vol. 8768, p. 87687D.

[2] N. Rani and C. Jyoti, "Text Steganography Techniques: A Review," *Int. J. Eng. Trends Technol.*, vol. 4, 2013.

[3] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, Mar. 2010.

[4] M. Hussain, "A Survey of Image Steganography Techniques," *Int. J. Adv. Sci. Technol. (IJAST),* vol. 54, pp. 113–125, May 2013.

[5] D. Alghazzawi, *Secret Communication on Facebook Using Image Steganography: Experimental Study*, vol. 14. 2016.

[6] U. Rizwan and H. Faheem Ahmed, "A New Approach in Steganography using different Algorithms and Applying Randomization Concept," 2012.

[7] S. Goel, A. Rana, and M. Kaur, "A Review of Comparison Techniques of Image Steganography," *Type Double Blind Peer Rev. Int. Res. J. Publ. Glob. Journals Inc*, vol. 13, 2013.

[8] R. Koikara, D. J. Deka, M. Gogoi, and R. Das, "A Novel Distributed Image Steganography Method Based on Block-DCT," Springer, Cham, 2015, pp. 423–435.

[9] D. Baby, J. Thomas, G. Augustine, E. George, and N. R. Michael, "A Novel DWT Based Image Securing Method Using Steganography," *Procedia Comput. Sci.*, vol. 46, pp. 612–618, Jan. 2015.

[10] E. Walia and P. Jain, "An Analysis of LSB &amp; DCT based Steganography," 2010.

[11] U. Sheth and S. Saxena, "Image steganography using AES encryption and least significant nibble," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 0876–0879.

[12] A. Habes, "4 least Significant Bits Information Hiding Implementation and Analysis," *ICGST Int. Conf. Graph. Vis. Image Process.*, 2005.

[13] R. Das and T. Tuithung, "A novel steganography method for image based on Huffman Encoding," in *2012 3rd National Conference on Emerging Trends and Applications in Computer Science*, 2012, pp. 14–18.

[14] A. Castiglione, B. D'Alessio, and A. De Santis, "Steganography and Secure Communication

on Online Social Networks and Online Photo Sharing,” in *2011 International Conference on Broadband and Wireless Computing, Communication and Applications*, 2011, pp. 363–368.

[15] J. Hiney, T. Dakve, K. Szczypiorski, and K. Gaj, “Using Facebook for Image Steganography,” Jun. 2015.

[16] I. Nechta, “Steganography in social networks,” in *2017 Siberian Symposium on Data Science and Engineering (SSDSE)*, 2017, pp. 33–35.

[17] J. Ning, I. Singh, H. V. Madhyastha, S. V. Krishnamurthy, G. Cao, and P. Mohapatra, “Secret message sharing using online social media,” in *2014 IEEE Conference on Communications and Network Security*, 2014, pp. 319–327.

[18] “Postman | API Development Environment.” [Online]. Available: https://www.getpostman.com/. [Accessed: 30-Jul-2018].

[19] “Free stock photos · Pexels.” [Online]. Available: https://www.pexels.com/. [Accessed: 30-Jul-2018].

[20] J. K. Su, F. Hartung, and B. Girod, “Digital watermarking of text, image, and video documents,” *Comput. Graph.*, vol. 22, no. 6, pp. 687–695, Dec. 1998.

[21] F. Hartung and M. Kutter, “Multimedia watermarking techniques,” *Proc. IEEE*, vol. 87, no. 7, pp. 1079–1107, Jul. 1999.

[22] V. M. Potdar, S. Han, and E. Chang, “A survey of digital image watermarking techniques,” in *INDIN ’05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, pp. 709–716.

[23] “SightEngine.” [Online]. Available: https://sightengine.com/. [Accessed: 25-Jul-2018].

[24] R. Smith, «Understanding encryption and cryptography basics,» 2003. [Online]. Available: https://searchsecurity.techtarget.com/Understanding-encryption-and-cryptography-basics. [Accessed 1-Aug-2018].

[25] «ICO» [Online]. Available: https://ico.org.uk/for-organisations/guide-to-data-protection/encryption/types-of-encryption/. [Accessed 1-Aug-2018].

[26] W. Diffie and M. Hellman, «New directions in cryptography», IEEE Transactions on Information Theory, Vol. 22 (6), pp. 644–654, 1976.

[27] A. Shamir, «Identity-Based Cryptosystems and Signature Schemes,» *CRYPTO,* τόμ. 196, pp. 47-53, 1987.

[28] R. Sakai, K. Ohgishi και M. Kasahara, «Cryptosystems based on pairing,» *In 2000 Symposium on Cryptography and Information Security,* αρ. SCIS2000, 2000.

[29] D. Boneh και M. Franklin, «Identity-Based Encryption from the Weil Pairing,» *Lecture Notes in Computer Science,* τόμ. 2139, pp. 213-229, 2001

[30] J. Bethencourt, A. Sahai και B. Waters, «Ciphertext-Policy Attribute-Based Encryption,» 2007.

[31] V. Goyal, O. Pandey, A. Sahai και B. Water, «Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data», Alexandria, Virginia, 2006.

[32] https://github.com/jfdm/pyPEBEL.