Mossebo Tcheunteu *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2021) 10:43
https://doi.org/10.1186/s13677-021-00258-2

Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH Open Access

Distributed data hiding in a single cloud storage environment

Stéphane Willy Mossebo Tcheunteu[1,2,3], Leonel Moyou Metcheka[1,2,3] and René Ndoundam[1,2,3*]

Abstract

Distributed steganography is an approach to concealing the secret in several files, leaving fewer traces than the classical approach. Recent work proposed by Moyou and Ndoumdan have improved this approach by preserving the integrity of these files in a multi-cloud storage environment. However, the approach requires a large size of the stego-key and the management of several cloud storage environments. Our contribution consists is to improve this approach by using a single cloud storage environment and reducing the size of the stego-key. In this work, a single cloud storage environment is used to solve the problems of managing several credentials, monetary costs and data controls associated with multi-cloud storage environments. The comparisons showed interesting results with simpler operations to be performed by the participants during the process.

Keywords: Distributed steganography, Stego-key, Data integrity, Management of cloud storage environments

Introduction

The protection of information has know a growing expansion over time, with the multiplicity of intrusions and thefts of information by the entities and institutions [1, 2]. Thus, many security techniques have been set up to counter these attacks[3]. Two main techniques are associated: Cryptography and steganography. Cryptography aims to make an information unintelligible by transformation [4], while steganography seeks to conceal an information in an inconspicuous way[5]. Steganography is mainly used in secret communications through an unsecured communication channel and a cover media. The most commonly used cover media are texts, images, sounds and videos[6–9].

When the scheme uses several cover media for the embedding of the secret, the approach is called distributed steganography[10]. In this approach, the secret is split into several shares that are then embedded into multiple carrier files. The main interest being to make the detection of the entire secret message extremely difficult. The embedding strategies used are based on the modification of each carrier file. However, these modifications can reveal the presence of a secret, through methods of steganalysis[11]. The carrier files of the secret are generally stored in a cloud storage environment for integrity and confidentiality requirements after embedding of the secret[12].

Moyou and Ndoundam[13] have proposed a new paradigm of steganography transparent to any attacker and resistant to the detection and extraction of the secret. The secret was distributed in a multi-cloud storage environment through several file extensions, and the use of a multi-cloud storage environment allowed to mask the presence of a communication channel between the communicating parties. The different files used were considered as a pointer to the secret data and constituted elements of the stego-key. Thus the proposed approach considered: the management of several cloud storage environments, a large size of the stego-key due to different lists of files and different credentials in the cloud accounts.

In this paper, our contribution consists is to propose a new distributed data hiding scheme in a single cloud storage environment, that improves the work done by Moyou

*Correspondence: ndoundam@yahoo.com
[1]Team GRIMCAPE
[2]University, IRD, UMMISCO, F-93143 Bondy, France
[3]Department of computer Science, University of Yaounde I, 812 Yaounde, Cameroon

and Ndoundam[13]. The goal is to avoid the management problems of multi-cloud storage environments and to reduce the large size of the stego-key. The technique uses a single cloud storage environment that provides a service of authenticity and confidentiality of files present in the cloud and masks the presence of a communication channel. The files do not undergo any modification during the steganographic process. The contribution of this work is focused on the following points:

- Problem management of multi-cloud storage environments using a single cloud storage environment.
- Reducing of large size of the stego-key.
- Simple operations to be performed by the participants during the embedding and extraction of the secret.

The rest of this paper is organized as follows: "Presentation of distributed models in steganography" section presents related work on distributed steganography models and their limits. "Our contribution" section is devoted to our contribution on a new distributed data hiding scheme in a single cloud storage environment. The experimental results are done in "Experimentation" section and finally "Conclusion" section is devoted to the conclusion.

Presentation of distributed models in steganography

Distributed steganography

Distributed steganography refers to the distribution of the secret into several parts which are then embedded into several cover media. In this approach, the secret is shared between several independent senders and a single receiver which receives the union of secret inputs in the communication[14]. The most commonly used cover media are images for their large data redundancy. The process requires meticulous modifications of these images, in order to go unnoticed to an unauthorized user in the communication[15]. The success of this approach lies in a good visual imperceptibility and a sufficient amount of payload[16].

Visual imperceptibility lies in the undetectability of a communication, while the payload guarantees a great capacity of secret that can be concealed[17]. Several approaches transit with visual imperceptibility as an indicator of images distortion to avoid detection of a secret message concealed[18, 19]. Others approaches use the distribution of the payload in the images[20, 21]. While hybrid approaches merge the features of several images[22] or combine the texture and payload associated with several images [23]. The interest of these approaches lies in a better resistance of the blind universal pooled steganalysis compared to other existing approaches.

The intervention of several human resources in a distributed steganography process is also applied to secret sharing[24]. In this secret sharing scheme, the secret key is divided into several parts, among a set of participants such that only a subset of these participants can reconstruct the secret key[25]. Thus, the system must define an efficient key sharing strategy between the participants in order to recover the target key. Counting-based secret sharing is presented as this promising approach to secret sharing that generates its sharing using simple specific bit replacement operations. This has several application domains in securing bank sensitive accounts and error tracking, voting systems trust, medical agreements, wills and inheritance authentication management[26]. Several security improvements of the secret key have been proposed, modifying the sharing generation process[27–29]. The generated share thus obtained, improves the security of access to the system[30, 31]. The generated secret keys are generally concealed in texts[32–34] and images[35, 36] using steganographic schemes.

Distributed steganography presents an improvement on classical steganography by concealing the secret in multiple cover media, making detection of the secret extremely difficult. However, modifications made to the cover media present limitations when setting up a process of steganalysis of these media. Indeed, several works in steganalysis on images are able to detect the presence of a secret and extract it. In general, the process is categorized into two types. One is targeted while the other is blind. Targeted steganalysis refers to an attack on a specific secret embedding algorithm[37, 38]. Blind steganalysis refers to an attack on several types of secret embedding algorithms, in which the goal is to classify the original files and stego files[39, 40].

Distributed model of Moyou and Ndoundam

The distributed data hiding model in a multi-cloud storage environment proposed by Moyou and Ndoundam [13], presents a new paradigm of distributed steganography that preserves the integrity of the files carrying the secret. In this model, the secret is distributed in different multimedia files that carry information of the secret message without being modified. The different multimedia files are stored in different cloud environments that mask the presence of a communication channel. The sender conceals the secret in different cloud storage environments, while the receiver retrieves the secret based on the stego-key elements. The integrity of the files being preserved, the model is more robust against steganalysis processes.

Concretely, the secret message is encoded in a specific base. To each value of the encoded secret message is associated a file in a list contained in the stego-key, the associated files carry the information of the value of the encoded secret message without being modified. Then the files are deposited in different cloud storage environments by the

sender. Each file thus deposited constitutes a pointer to the encoded secret message. Finally, the two communicating parties having the same lists of files contained in the stego-key. It will be enough for the receiver to retrieve the positions of the different files in the different cloud storage environments based on these lists. Thus, the receiver reconstitutes the encoded secret message and the initial secret message.

In this model, the elements contained in the stego-key are:

- Cloud environments $c_0, c_1, \ldots, c_{n-1}$.
- Authentication for access to each cloud account W_i(user name and password), such that $0 \leq i \leq n-1$.
- An ordered set of disjoint lists , $L^{(0)}, L^{(1)}, \ldots, L^{(k-1)}$ where each list i contains B files: $L_0^{(i)}, L_1^{(i)}, \ldots, L_{B-1}^{(i)}$, such that $i = 0, 1, \ldots, k-1$, each file can take any type of format.
- The base B, such that $|L^{(0)}| = |L^{(1)}| = \ldots = |L^{(k-1)}| = B$ and $B \geq 2$.

In view of the elements contained in the stego-key, the limits listed in this scheme are based on the large size of the stego-key related to the management of several cloud account credentials and several disjointed lists, the problems of managing several cloud storage environments such that: management of several cloud account credentials, monetary costs associated with cloud storage environments, data controls in different multi-cloud environments[41–43]. The problems listed can result from: a difficulty of data controls by the participants due to the distribution of multimedia files in several cloud environments, high cost for the acquisition of different cloud accounts. So we are motivated to design a steganographic scheme, that reduces the large size of the stego-key and uses a single cloud storage environment. In our scheme, the real need to use this proposition lies in management of the problems of multi-cloud storage environments and in the size reduced of the stego-key. Indeed, in our proposed scheme, a single credential is used instead of several credentials, the cost associated with the cloud environments is reduced to one, the control of data in the cloud storage environments is reduced to a single cloud.

Our contribution

The purpose of the proposed scheme is to improve the work done by Moyou and Ndoundam[13], while preserving the integrity aspect of multimedia files and the masking of a communication channel. This objective is achieved through the use of a single cloud storage environment and a reduced number of elements contained in the stego-key. The proposed scheme is declined into 3 approaches with modification of receiver information, in order to make difficult the detection of the secret in the cloud, by an attacker possessing the stego-key. The cloud allows to conceal a secret message and to mask the presence of a communication channel, while the reduced number of elements in the stego-key allows to reduce the operations performed by the participants.

The stego-key consists of the credential of a single cloud storage environment and the base used. This is exchanged before the start of the process during an encrypted communication or a physical meeting. An example of real use of the proposed scheme is in a process of secret communication between two entities or institutions. Concretely, if we consider two entities A and B. Entity A uses the stego-key consisted of the credential of a cloud storage environment and a base, to conceal a secret in several multimedia files contained in the cloud. The multimedia files preserving their integrity. Entity B logs to the cloud storage environment and extracts the secret using the stego key.

Notations and hypothesis

The Table 1 gives the different symbols and their representations in the 3 proposed approaches

Hypothesis

The cloud storage environment consists of a set of cover folders containing several multimedia files. The number of cover folders is at least equal to the size of the secret. The number of files in each folder is at least equal to the value of the base. The following relationships are checked in the cloud storage environment:

$$\begin{cases} \forall\, i_1, i_2, j_1, j_2, 0 \leq i_1, i_2 < n, 0 \leq j_1, j_2 < B \ \text{ if } (i_1 = i_2 \text{ and } j_1 \neq j_2) \text{ then } F_{j1}^{(i1)} \neq F_{j2}^{(i2)} \\ \forall i_1, i_2, j_1, j_2, 0 \leq i_1, i_2 < n, 0 \leq j_1, j_2 < B \quad \text{if } (i_1 \neq i_2) \text{ then } F_{j1}^{(i1)} \neq F_{j2}^{(i2)} \end{cases}$$

Table 1 symbols and their representations in the proposed approaches

Symbol	Representation
s	the secret message formatted in base 2
B	the base used to encode the secret, such that $B \geq 2$
$(s_{n-1}s_{n-2}\ldots s_0)_B$	the representation in base B of the secret
$F^{(i)}$	the i^{th} folder which contains different types of file, such that $0 \leq i < n$
$L^{(i)}$	the i^{th} list which contains different types of file, such that $0 \leq i < n$
$\tilde{F}$	the stego folder which contains the concealed secret
$\tilde{F}_i$	the file i in the stego folder
$F_j^{(i)}$	the j^{th} file in folder number i, such that $0 \leq i < n$ and $0 \leq j < B$
$L_j^{(i)}$	the j^{th} file in list number i, such that $0 \leq i < n$ and $0 \leq j < B$,

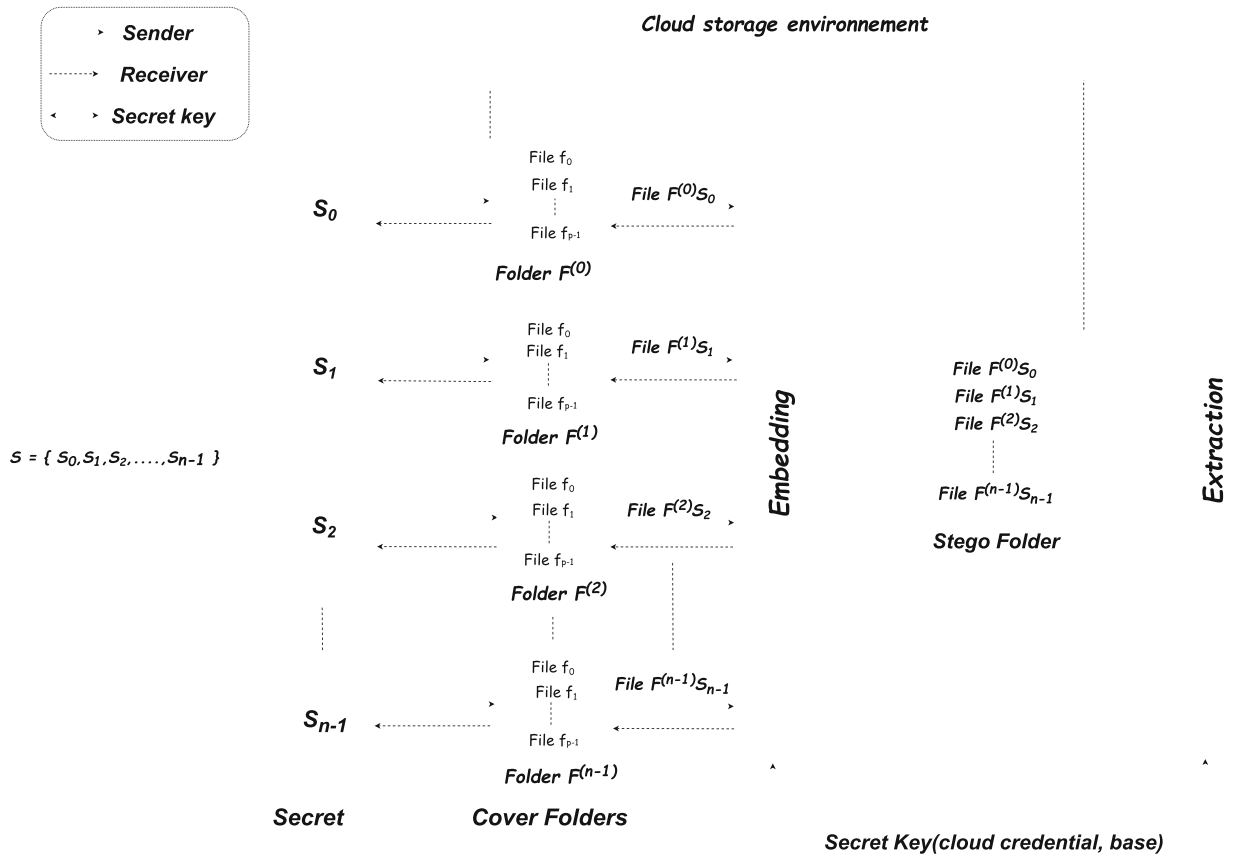

Fig. 1 Overview of a first approach

First approach

Overview

In this scheme proposed in Fig. 1, the secret communication process presented is described according to the following steps:

- The sender and receiver share the credential of a cloud account and the base used, before the communication.
- The sender encodes the secret in the base described in the secret key.
- For each value of the encoded secret, the file corresponding to this value in the position folder of this index is copied in a folder representing the stego-folder.
- Since the receiver shares the same secret key with the sender, it logs to the cloud storage environment and searches the correspondence between the files in the stego-folder and the files in the different cover folders. The receiver reconstructs thus the secret message based on this match.

The embedding and extraction algorithms are performed through the following elements:

- The cover object represents any multimedia files extension located in the cover folders of cloud storage environment.
- The cover folders represents a set of multimedia files.
- The stego folder represents a set of multimedia files that conceal the secret.
- The secret message represents any message format encoded in a specific base.
- The secret key represents the elements shared between the sender and the receiver.

Embedding

The embedding algorithm of the secret message performed by the sender is defined as follows:

Input:

C: the cloud account;
s: the secret message;
B: the base used;
$F^{(0)}, F^{(1)}, \ldots, F^{(n-1)}$: the cover folders in the cloud account;
$F^{(i)}$: the folder number i in the cloud account;
$\tilde{F}$: the stego folder in the cloud account;

$F_j^{(i)} : 0 \leq i < n$ and $0 \leq j < B$, file j from folder number i in cloud account;
W: access for authentication in the cloud account;

Output:
$\tilde{C}$: the modified cloud account;

Begin

I Create an empty folder $\tilde{F}$ that represents the stego folder;
II Convert the secret message s in base B such that $s = (s_{n-1}s_{n-2}\ldots s_0)_B$, where $0 \leq s_i < B$;
III For each position i of the secret message s: $i = 0, 1, \ldots, n-1$

A) Find the file with index s_i in the folder $F^{(i)}$;
B) Select and copy the file $F_{s_i}^{(i)}$ and paste in the stego folder $\tilde{F}$;

End

Extraction

The extraction algorithm of the secret message performed by the receiver is defined as follows:

Input:
$\tilde{C}$: Cloud account modified;
W: access for authentication in the cloud account;
B: the base used;
$F^{(0)}, F^{(1)}, \ldots, F^{(n-1)}$: the cover folders in the cloud account;
$F^{(i)}$: the folder number i in the cloud account;
$\tilde{F}$: the stego folder in the cloud account;
$\tilde{F}_i$: the file i in the stego folder;
$F_j^{(i)} : 0 \leq i < n$ and $0 \leq j < B$, file j from folder number i in cloud account;
$tab[0, \ldots, n-1]$: integer array that retrieves each index s_i of the folder number i;

Output:
s: the secret message;

Begin

I $i = 0$; // first file of the stego folder $\tilde{F}$
II while($i < n$) // i browse each file in the stego folder $\tilde{F}$

A) For each folder j in the cloud account : $j = 0, 1, \ldots, n-1$

1) For each file k in the current folder $F^{(j)}$: $k = 0, 1, \ldots, B-1$

a) Compare the file $\tilde{F}_i$ in the stego folder with the file $F_k^{(j)}$;
b) if $(\tilde{F}_i = F_k^{(j)})$ then

i) $tab[j] = k$;
ii) $i = i + 1$; // next file of the stego folder $\tilde{F}$
iii) go to instruction II);

III Compute $m = \sum_{j=0}^{n-1}(tab[j] \times B^j)$;
IV Convert m to binary and get the secret message s;
V Delete the stego folder $\tilde{F}$;

End

Second approach

Overview

In this scheme proposed in Fig. 2, the receiver logs to the cloud storage environment and copies the files from the cover folders in different lists before the secret communication. These lists of files will allow to perform the correspondence with the files of the stego-folder, because the files in the cloud storage environment that conceal the secret are cut during the process. The interest is to prevent an attacker to perform any correspondence of the files of the stego-folder, in case of access of this one in the cloud storage environment. The secret communication process is described according to the following steps:

- The sender and receiver share the credential of a cloud account and the base used, before the communication.
- The sender encodes the secret in the base described in the secret key.
- For each value of the encoded secret, the file corresponding to this value in the position folder of this index is cut in a folder representing the stego-folder.
- Since the receiver shares the same secret key with the sender, it logs to the cloud storage environment and searches the correspondence between the files in the stego-folder and the files in the different lists. The receiver reconstructs thus the secret message based on this match.

The embedding and extraction algorithms are performed through the following elements:

- The cover object represents any multimedia files extension located in the cover folders of cloud storage environment.
- The cover folders represents a set of multimedia files.
- Lists of files held by the receiver.
- The stego folder represents a set of multimedia files that conceal the secret.
- The secret message represents any message format encoded in a specific base.
- The secret key represents the elements shared between the sender and the receiver.

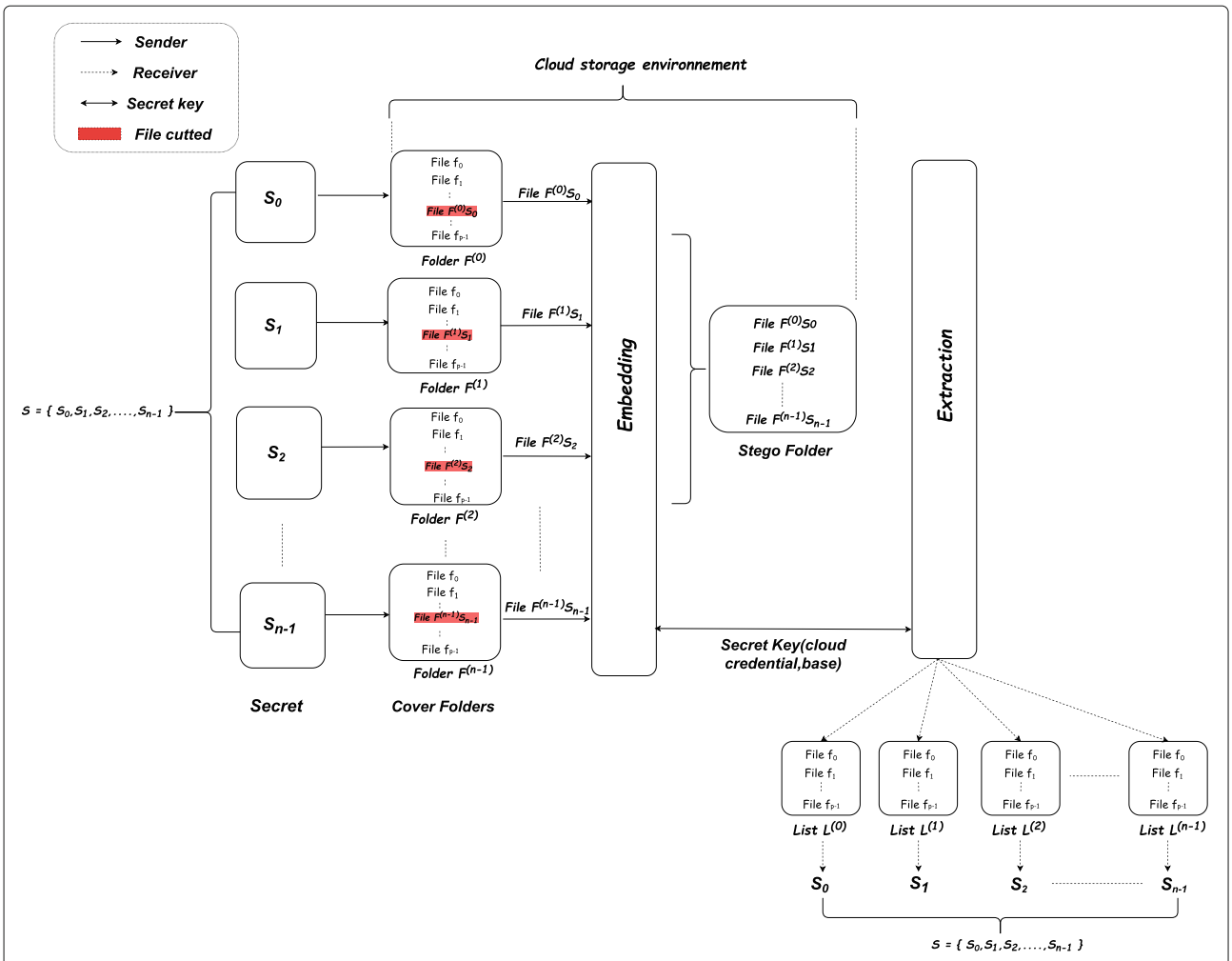

Fig. 2 Overview of a second approach

Embedding

The embedding algorithm of the secret message performed by the sender is defined as follows:

Input:

C: the cloud account;
s: the secret message;
B: the base used;
$F^{(0)}, F^{(1)}, \ldots, F^{(n-1)}$: the cover folders in the cloud account;
$F^{(i)}$: the folder number i in the cloud account;
$\tilde{F}$: the stego folder in the cloud account;
$F_j^{(i)}$: $0 \leq i < n$ and $0 \leq j < B$, file j from folder number i in cloud account;
W: access for authentication in the cloud account;

Output:

$\tilde{C}$: the modified cloud account;

Begin

I Create an empty folder $\tilde{F}$ that represents the stego folder;
II Convert the secret message s in base B such that $s = (s_{n-1}s_{n-2}\ldots s_0)_B$, where $0 \leq s_i < B$;
III For each position i of the secret message s: $i = 0, 1, \ldots, n-1$
  A) Find the file with index s_i in the folder $F^{(i)}$;
  B) Select and cut the file $F_{s_i}^{(i)}$ and paste in the stego folder $\tilde{F}$;

End

Extraction

The extraction algorithm of the secret message performed by the receiver is defined as follows:

Input:

$\tilde{C}$: Cloud account modified;
W: access for authentication in the cloud account;

B: the base used;
$L^{(0)}, L^{(1)}, \ldots, L^{(n-1)}$: the lists of files of the receiver;
$L^{(i)}$: the list number i;
$\tilde{F}$: the stego folder in the cloud account;
$\tilde{F}_i$: the file i in the stego folder;
$L_j^{(i)} : 0 \leq i < n$ and $0 \leq j < B$, file j from list number i;
$tab[0, \ldots, n-1]$: integer array that retrieves each index s_i of the list number i;

Output:
s: the secret message;

Begin

I $i = 0$; // first file of the stego folder $\tilde{F}$
II while($i < n$) // i browse each file in the stego folder $\tilde{F}$

A) For each list j: $j = 0, 1, \ldots, n-1$

1) For each file k in the list $L^{(j)}$: $k = 0, 1, \ldots, B-1$

a) Compare the file $\tilde{F}_i$ in the stego folder with the file $L_k^{(j)}$;
b) if($\tilde{F}_i = L_k^{(j)}$) then

i) $tab[j] = k$;
ii) $i = i + 1$; // next file of the stego folder $\tilde{F}$
iii) go to instruction II);

III Compute $m = \sum_{j=0}^{n-1} (tab[j] \times B^j)$;
IV Convert m to binary and get the secret message s;
V Delete the stego folder $\tilde{F}$;

End

Third approach

Overview

In this scheme proposed in Fig. 3, The lists of files held by the receiver are stored in an intermediate cloud storage environment. These lists of files will allow to perform the correspondence with the files of the stego-folder, because the files in the cloud storage environment that conceal the secret are cut during the process. The interest is to secure the files held by the receiver in the intermediate cloud account. The secret communication process is described according to the following steps:

- The sender and receiver share the credential of a cloud account and the base used, before the communication.
- The sender encodes the secret in the base described in the secret key.
- For each value of the encoded secret, the file corresponding to this value in the position folder of this index is cut in a folder representing the stego-folder.
- Since the receiver shares the same secret key with the sender and holds the credential of the intermediate cloud account, it logs to the cloud storage environment and searches the correspondence between the files in the stego-folder and the files in the lists of the intermediate cloud account. The receiver reconstructs thus the secret message based on this match.

The embedding and extraction algorithms are performed through the following elements:

- The cover object represents any multimedia files extension located in the cover folders of cloud storage environment.
- The cover folders represents a set of multimedia files.
- The credential of the intermediate cloud account held by the receiver.
- The lists of files in the intermediate cloud environment held by the receiver.
- The stego folder represents a set of multimedia files that conceal the secret.
- The secret message represents any message format encoded in a specific base.
- The secret key represents the elements shared between the sender and the receiver.

Embedding

The embedding algorithm of the secret message performed by the sender is defined as follows:

Input:
C: the cloud account;
s: the secret message;
B: the base used;
$F^{(0)}, F^{(1)}, \ldots, F^{(n-1)}$: the cover folders in the cloud account;
$F^{(i)}$: the folder number i in the cloud account;
$\tilde{F}$: the stego folder in the cloud account;
$F_j^{(i)} : 0 \leq i < n$ and $0 \leq j < B$, file j from folder number i in cloud account;
W: access for authentication in the cloud account;

Output:
$\tilde{C}$: the modified cloud account;

Begin

I Create an empty folder $\tilde{F}$ that represents the stego folder;
II Convert the secret message s in base B such that $s = (s_{n-1}s_{n-2}\ldots s_0)_B$, where $0 \leq s_i < B$;
III For each position i of the secret message s: $i = 0, 1, \ldots, n-1$

A) Find the file with index s_i in the folder $F^{(i)}$;

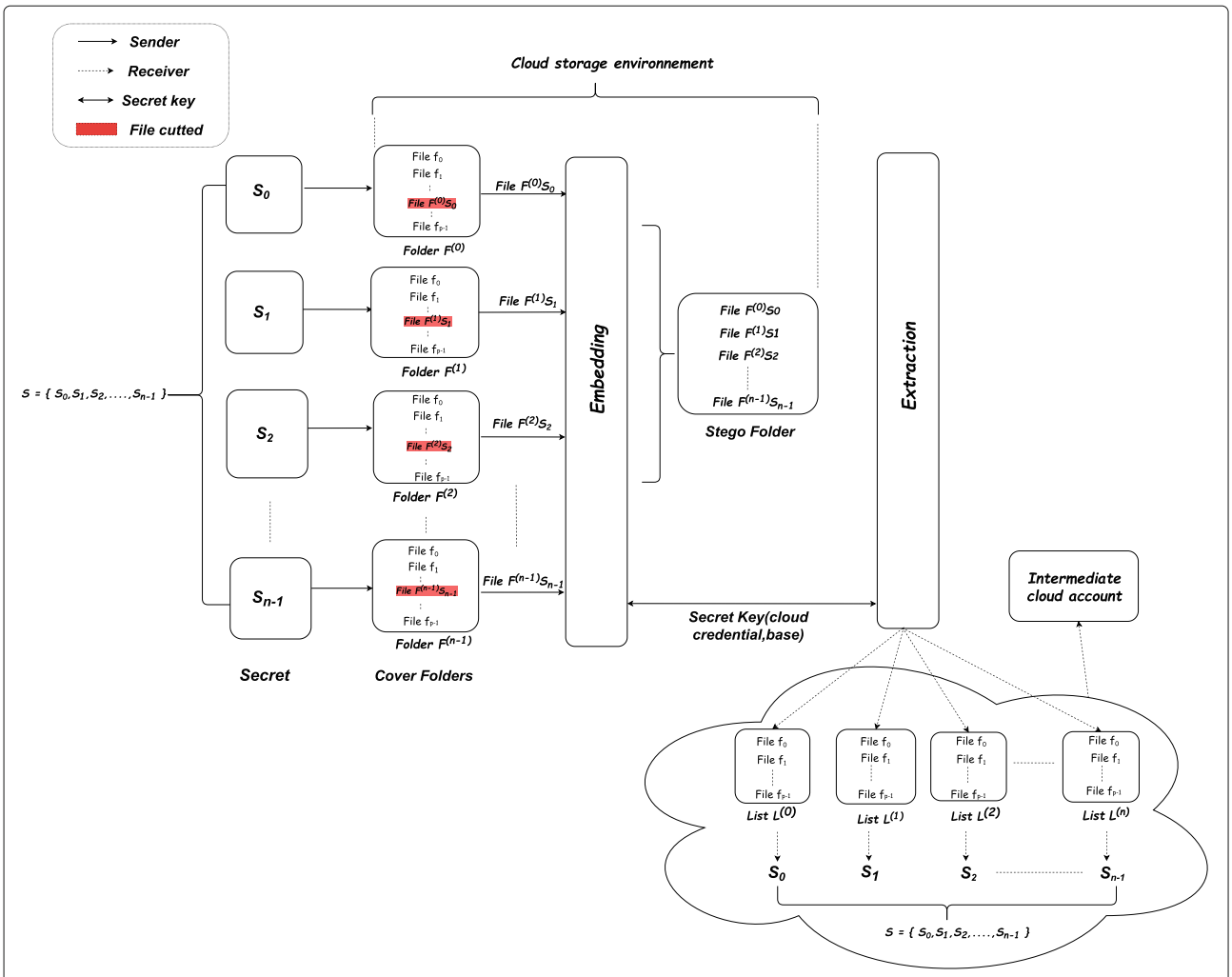

Fig. 3 Overview of a third approach

B) Select and cut the file $F_{s_i}^{(i)}$ and paste in the stego folder $\tilde{F}$;

End

Extraction

The extraction algorithm of the secret message performed by the receiver is defined as follows:

Input:

$\tilde{C}$: Cloud account modified;
W: access for authentication in the cloud account;
B: the base used;
$L^{(0)}, L^{(1)}, \ldots, L^{(n-1)}$: the lists of files of the receiver in the intermediate cloud environment;
$L^{(i)}$: the list number i in the intermediate cloud environment;
$\tilde{F}$: the stego folder in the cloud account;
$\tilde{F}_i$: the file i in the stego folder;
$L_j^{(i)}$: $0 \leq i < n$ and $0 \leq j < B$, file j from list number i in the intermediate cloud environment;
$tab[0, \ldots, n-1]$: integer array that retrieves each index s_i of the list number i;

Output:

s: the secret message;

Begin

I $i = 0$; // first file of the stego folder $\tilde{F}$
II while($i < n$) // i browse each file in the stego folder $\tilde{F}$

A) For each list j in the intermediate cloud environment: $j = 0, 1, \ldots, n-1$

1) For each file k in the list $L^{(j)}$: $k = 0, 1, \ldots, B-1$

a) Compare the file $\tilde{F}_i$ in the stego folder with the file $L_k^{(j)}$;

b) if($\tilde{F}_i = L_k^{(j)}$) then

i) $tab[j] = k$;
ii) $i = i + 1$; // next file of the stego folder $\tilde{F}$
iii) go to instruction II);

III Compute $m = \sum_{j=0}^{n-1}(tab[j] \times B^j)$;
IV Convert m to binary and get the secret message s;
V Delete the stego folder $\tilde{F}$;

End

Time complexity

In this subsection, we evaluate the time complexity of the 3 proposed approaches. We have a secret message s distributed between n cover folders, each folder contains at least B files. For the embedding of the secret message. Each index s_i of the secret message s encoded in base B, corresponds to a file in the cover folder i. The secret message s comprising n index position. Moreover, the formatted secret message s is converted into base B in $O(log_B(s))$. Therefore, the time complexity of the approaches is $O(n)$.

For the extraction of the secret message, the correspondence between the files in the stego folder and the cover folders or lists is done at $O(n^2 * B)$. The lists comprising the same files as the cover folders. Moreover, the formatted secret message is obtained in $O(log_2(s))$. Therefore, the time complexity of the approaches is $O(n^2 * B)$.

Experimentation

In this section we present an evaluation of the hidden bits capacity of our proposed schemes and the execution through 3 examples. Then we present a discussion of our work and finally a security analysis of our proposed schemes.

Evaluation of hidden bits capacity

The idea is to make an estimation of the number of bits hidden in the cloud storage environment. Each folder in the cloud has a value in base B and this value varies from 0 to $B-1$, so we have B possibilities by folder. For the set of n folders in the cloud, we have B^n possibilities, so the number of hidden bits is:

$$\boldsymbol{log_2(B^n) = n \times log_2(B)}.$$

Examples

In this subsection, we describe in detail the processes of embedding and extraction of the secret on 3 examples for each approach presented. In these examples, the formatted secret messages have the values 1010100,10101100,10011 with the respective bases $\boldsymbol{B = 8, 4, 3}$. The credentials of the cloud accounts used for each example are given in the Table 2. In each example presented, the secret key consists of the credential of a cloud account and the base used.

Example 1

In this example for the first approach, the formatted secret message is $\boldsymbol{s = 1010100}$ and the base used is $\boldsymbol{B = 8}$. The cloud storage environment presented in the Table 3 consists of 3 folders($\boldsymbol{F^{(2)}, F^{(1)}, F^{(0)}}$), with 8 mutimedia files for each folder.

The steps for the embedding of the secret message are as follows:

- Step 1: Create an empty folder $\tilde{\boldsymbol{F}}$ that represents the stego folder:

The stego folder	Folder $\tilde{\boldsymbol{F}}$
	empty

- Step 2: Convert the secret message $\boldsymbol{s = 1010100}$ in base 8: $\boldsymbol{s = (124)_8}$;
- Step 3: For each position i of the secret s: i = 0,1,2

A) Find the file with the index $\boldsymbol{s_i}$ in the folder $\boldsymbol{F^{(i)}}$:

$\boldsymbol{s_2}$	$\boldsymbol{s_1}$	$\boldsymbol{s_0}$
1	2	4
↓	↓	↓
article.txt	french.pdf	learning.docx

- Step 4: Select and copy each file found $\boldsymbol{F_{s_i}^{(i)}}$ and paste in the stego folder $\tilde{\boldsymbol{F}}$:

The stego folder	folder $\tilde{\boldsymbol{F}}$
$\boldsymbol{file_0}$	learning.docx
$\boldsymbol{file_1}$	french.pdf
$\boldsymbol{file_2}$	article.txt

The steps for extraction of the secret message are as follows:

- Step1: $\boldsymbol{i = 0}$; // first file of the stego folder $\tilde{\boldsymbol{F}}$
- Step2: while($\boldsymbol{i < n}$) // i browse each file in the stego folder $\tilde{\boldsymbol{F}}$(learning.docx, french.pdf, article.txt)

A) For each folder j in the cloud account : $\boldsymbol{j = 0, 1, 2}$

1) For each file k in the current folder $\boldsymbol{F^{(j)}}$: $\boldsymbol{k = 0, 1, 2, 3, 4, 5, 6, 7}$

a) Compare the file $\tilde{\boldsymbol{F}}_{\boldsymbol{i}}$ in the stego folder with the file $\boldsymbol{F_k^{(j)}}$;
b) if($\tilde{\boldsymbol{F}}_{\boldsymbol{i}} = \boldsymbol{F_k^{(j)}}$) then

i) $\boldsymbol{tab[j] = k}$;

Table 2 The set of 3 cloud service providers handled and their credentials

Cloud Name	Login	Password
iCloud	user1@gmail.com	password1
OneDrive	user2@gmail.com	password2
Google Drive	user3@gmail.com	password3

ii) $i = i + 1$; // next file of the stego folder $\tilde{F}$
iii) go to Step 2;

Folder $\tilde{F}$	article.txt #2	french.pdf #1	learning.docx #0
j	2	1	0
	↓	↓	↓
$tab[j]$	1	2	4

- Step 3: Compute $m = \sum_{j=0}^{2}(tab[j] \times 8^j) = tab[0] \times 8^0 + tab[1] \times 8^1 + tab[2] \times 8^2 = 4 \times 8^0 + 2 \times 8^1 + 1 \times 8^2 = 84$;
- Step 4: Convert m to binary and get the secret message s:
  $m = 84 = (1010100)_2$, the secret message $s = (1010100)_2$ is retrieved;
- Step 5: Delete the stego folder $\tilde{F}$;

Example 2

In this example for the second approach, the formatted secret message is $s = 10101100$ and the base used is $B = 4$. The cloud storage environment presented in the Table 4 consists of 4 folders($F^{(3)}, F^{(2)}, F^{(1)}, F^{(0)}$), with 4 mutimedia files for each folder.

The steps for the embedding of the secret message are as follows:

- Step 1: Create an empty folder $\tilde{F}$ that represents the stego folder:

The stego folder	Folder $\tilde{F}$
	empty

- Step 2: Convert the secret message $s = 10101100$ in base 4: $s = (2230)_4$;
- Step 3: For each position i of the secret s

A) Find the file with the index s_i in the folder $F^{(i)}$:

s_3	s_2	s_1	s_0
2	2	3	0
↓	↓	↓	↓
presentation.pptx	english.pptx	statistics.xlsx	steganalysis.pdf

- Step 4: Select and cut each file found $F_{s_i}^{(i)}$ and paste in the stego folder $\tilde{F}$:

The stego folder	Folder $\tilde{F}$
$file_0$	steganalysis.pdf
$file_1$	statistics.xlsx
$file_2$	english.pptx
$file_3$	presentation.pptx

The steps for the extraction of the secret message are as follows:

The receiver holds a copy of the files of the cover folders located in the cloud environment OneDrive. Table 5 presents the lists of files resulting of this copy. The correspondence between the files in the stego folder is made on these lists, because these files had been cut in the cover folders of the cloud environment.

- Step1: $i = 0$; // first file of the stego folder $\tilde{F}$
- Step2: while($i < n$) // i browse each file in the stego folder $\tilde{F}$(steganalysis.pdf, statistics.xlsx, english.pptx, presentation.pptx)

A) For each list j: $j = 0, 1, 2, 3$

1) For each file k in the list $L^{(j)}$: $k = 0, 1, 2, 3$

a) Compare the file $\tilde{F}_i$ in the stego folder with the file $L_k^{(j)}$;

Table 3 Set of files distributed by folder in the cloud environment iCloud

Folder	Folder $F^{(2)}$	Folder $F^{(1)}$	Folder $F^{(0)}$
$file_0$	analysis.pptx	document.pdf	conference.pptx
$file_1$	article.txt	english.pptx	cryptanalysis.pdf
$file_2$	book.pdf	french.pdf	education.pptx
$file_3$	cryptography.pdf	homework.pdf	hacking.pdf
$file_4$	exercise.docx	music.mp3	learning.docx
$file_5$	network.pptx	scheduling.xlsx	security.pdf
$file_6$	presentation.pptx	thesis.docx	statistics.xlsx
$file_7$	steganography.pdf	video.mp4	steganalysis.pdf

Table 4 Set of files distributed by folder in the cloud environment OneDrive

Folder	Folder $F^{(3)}$	Folder $F^{(2)}$	Folder $F^{(1)}$	Folder $F^{(0)}$
$file_0$	book.pdf	scheduling.xlsx	learning.docx	steganalysis.pdf
$file_1$	article.txt	thesis.docx	cryptanalysis.pdf	cryptography.pdf
$file_2$	presentation.pptx	english.pptx	conference.pptx	network.pptx
$file_3$	exercise.docx	document.pdf	statistics.xlsx	cryptanalysis.pdf

b) if($\tilde{F}_i = L_k^{(j)}$) then

i) $tab[j] = k$;
ii) $i = i + 1$; // next file of the stego folder $\tilde{F}$
iii) go to Step 2;

Folder $\tilde{F}$	presentation.pptx #3	english.pptx #2	statistics.xlsx #1	steganalysis.pdf #0
j	3	2	1	0
	↓	↓	↓	↓
$tab[j]$	2	2	3	0

- Step 3: Compute $m = \sum_{i=0}^{3}(tab[i] \times 4^i) = tab[0] \times 4^0 + tab[1] \times 4^1 + tab[2] \times 4^2 + tab[3] \times 4^3 = 0 \times 4^0 + 3 \times 4^1 + 2 \times 4^2 + 2 \times 4^3 = 172$;
- Step 4: Convert m to binary and get the secret message s:
  $m = 172 = (10101100)_2$, the secret message s = $(10101100)_2$ is retrieved;
- Step 5: Delete the stego folder $\tilde{F}$;

Example 3

In this example for the third approach, the formatted secret message is $s = 10011$ and the base used is $B = 3$. The cloud storage environment presented in the Table 6 consists of 3 folders($F^{(2)}, F^{(1)}, F^{(0)}$), with 3 mutimedia files for each folder.

The steps for the embedding of the secret message are as follows:

- Step 1: Create an empty folder $\tilde{F}$ that represents the stego folder:

The stego folder	Folder $\tilde{F}$
	empty

- Step 2: Convert the secret message $s = 10011$ in base 3: $s = (201)_3$;
- Step 3: For each position i of the secret message s

A) Find the file with index s_i in the folder $F^{(i)}$:

s_2	s_1	s_0
2	0	1
↓	↓	↓
php.xlsx	java.pdf	css.pdf

- Step 4: Select and cut the file $F_{s_i}^{(i)}$ and paste in the stego folder $\tilde{F}$:

The stego folder	Folder $\tilde{F}$
$file_0$	css.pdf
$file_1$	java.pdf
$file_2$	php.xlsx

The steps for the extraction of the secret message are as follows:

the receiver holds the credential of an intermediate cloud account (Dropbox), which contains a copy of the files of the cover folders of the cloud environment Google Drive. Dropbox cloud has for login *user4@gmail.com* and for password *password4*. Table 7 presents the lists of files in the intermediate cloud environment resulting from this copy. The correspondence between the files in the stego folder is made on these lists, because these files had been cut in the cover folders of the cloud environment.

- Step 1: $i = 0$; // first file of the stego folder $\tilde{F}$
- Step2: while($i < n$) // i browse each file in the stego folder $\tilde{F}$(css.pdf, java.pdf, php.xlsx)

A) For each list j in the intermediate cloud environment: $j = 0, 1, 2$

1) For each file k in the list $L^{(j)}$: $k = 0, 1, 2$

a) Compare the file $\tilde{F}_i$ in the stego folder with the file $L_k^{(j)}$;
b) if($\tilde{F}_i = L_k^{(j)}$) then

Table 5 Set of files distributed by list held by the receiver

List	List $L^{(3)}$	List $L^{(2)}$	List $L^{(1)}$	List $L^{(0)}$
$file_0$	book.pdf	scheduling.xlsx	learning.docx	steganalysis.pdf
$file_1$	article.txt	thesis.docx	cryptanalysis.pdf	cryptography.pdf
$file_2$	presentation.pptx	english.pptx	conference.pptx	network.pptx
$file_3$	exercise.docx	document.pdf	statistics.xlsx	cryptanalysis.pdf

Table 6 Set of files distributed by folder in the cloud environment Google Drive

Folder	Folder $F^{(2)}$	Folder $F^{(1)}$	Folder $F^{(0)}$
$file_0$	laravel.pptx	java.pdf	jquery.pdf
$file_1$	javascript.pdf	database.pdf	css.pdf
$file_2$	php.xlsx	xml.pptx	html.pptx

i) $\boldsymbol{tab[j] = k}$;
ii) $\boldsymbol{i = i + 1}$; // next file of the stego folder $\tilde{\boldsymbol{F}}$
iii) go to Step 2;

Folder $\tilde{\boldsymbol{F}}$	php.xlsx #2	java.pdf #1	css.pdf #0
j	2	1	0
	↓	↓	↓
$\boldsymbol{tab[j]}$	2	0	1

- Step 3: Compute $\boldsymbol{m = \sum_{i=0}^{2}(tab[i] \times 3^i)} = \boldsymbol{tab[0] \times 3^0 + tab[1] \times 3^1 + tab[2] \times 3^2 = 1 \times 3^0 + 0 \times 3^1 + 2 \times 3^2} = 19$;
- Step 4: Convert m to binary and get the secret message s:
  $\boldsymbol{m = 19 = (10011)_2}$, the secret message s = $\boldsymbol{(10011)_2}$ is retrieved;
- Step 5: Delete the stego folder $\tilde{\boldsymbol{F}}$;

Discussion

The different approaches proposed present steganographic schemes of secret distribution in a single cloud storage environment. The cloud environment presents a set of files distributed by folder that allows to conceal a secret message, preserving the integrity of the files that conceal the secret. The set of files distributed by cover folder and the single cloud environment allow to reduce the size of the key and the management of several cloud environments in the approach proposed by Moyou and Ndoundam[13].

Tables 8 and 9 give a comparison between our 3 approaches and Moyou and Ndoundam's approach[13] based on keys, management of the cloud storage environment and receiver elements.

In these tables, the common elements to each approach represent the key shared between the sender and the receiver before the process. In Moyou and Ndoundam's approach[13], the key consists of k lists, n cloud accounts credentials and the base B used. Each list comprising B files. We denote $\boldsymbol{n + k * B + 1}$ elements in the key. In our proposed approaches, a single cloud account credential and base B are the elements of the key. Therefore, the size of the key is reduced to 2 elements in our approaches.

The files duplicated in the same cloud represent the copy of the files that conceal the secret in the same cloud environment. In approach 1, a copy of the files that conceal the secret is made in the cloud environment while in approaches 2 and 3 the files are cut in the cloud environment. These files copied or cutted after a series of correspondence with the encoded secret represent the operations performed by the sender. In Moyou and Ndoundam's approach[13], the operations performed by the sender represent: the encoding of the secret in a base, the partitioning of the secret according to different cloud environments, identification in different cloud environments, transmission of a set of files in different cloud environments based on lists of files in the key. Therefore, if we denote the following elements: a secret message s distributed among n cloud storage environments or cover folders, a base B, k file lists each with B files per list. Moyou and Ndoundam's approach[13] requires a time complexity of $\boldsymbol{O(n * k)}$ on the sender side with n representing the different cloud storage environments, while in our approaches a time complexity of $\boldsymbol{O(n)}$ is required with n representing the different cover folders.

The files duplicated at the receiver represent a copy of a set of files of cloud environment in different lists and in an intermediate cloud environment. In approach 1, no copy of the files is made at the receiver because the files that conceal the secret are duplicated in the cloud environment for matching. In approaches 2 and 3, a copy of the files is made at the receiver for matching because the files have been cut in the cloud environment. In Moyou and Ndoundam's approach[13], the operations performed by the receiver represent: the identification in different cloud environments, the matching of each file in the cloud environments with the lists of the files of the key, the calculation of the formatted secret message. Thus

Table 7 Set of files distributed by list in the intermediate cloud environment held by the receiver

List	List $L^{(2)}$	List $L^{(1)}$	List $L^{(0)}$
$file_0$	laravel.pptx	java.pdf	jquery.pdf
$file_1$	javascript.pdf	database.pdf	css.pdf
$file_2$	php.xlsx	xml.pptx	html.pptx

Table 8 Comparison of our approaches and Moyou and Ndoundam's approach based on the key and management of the cloud storage environment

Approaches	Credentials cloud account	Lists of files	Duplicate files in the same cloud	Duplicate files at the receiver
Moyou and Ndoundam approach [13]	$c_0, c_1, c_2, \ldots, c_{n-1}$	$L^{(0)}, L^{(1)}, \ldots, L^{(k-1)}$	none	none
Approach 1	c_0	none	yes	none
Approach 2	c'_0	$L^{(0)}, L^{(1)}, \ldots, L^{(n-1)}$	none	yes
Approach 3	c_0, c'_0	$L^{(0)}, L^{(1)}, \ldots, L^{(n-1)}$	none	yes

in our approaches, the operations of correspondence and calculation of the formatted secret message being also performed, the main gain lies in the identification of a single cloud account at the receiver.

Security analysis

In this subsection, we present different attack schemes of a spy on the proposed approaches based on two main hypothesis. Hypothesis 1 describes the fact that the spy does not have access to the key and therefore cannot access the cloud environment, while in hypothesis 2 the spy has access to the key and therefore to the cloud environment.

Hypothesis 1:

In the 3 approaches presented, no detection or extraction of a secret is possible, because access to the cloud environment is impossible for the spy.

Hypothesis 2:

In the first approach, the spy has access to the cloud environment and will be able to perform the correspondance between the files of the stego folder and the files of cover folders. This matching is possible, because the files that conceal the secret are duplicated in the cloud environment. For each file listed in the stego folder, the spy will have to perform $\boldsymbol{O(n * B)}$ browse and comparisons in the files of the cover folders in the case of an exhaustive search.

In the second approach, the spy has access to the cloud environment but will not be able to perform the correspondence between the files of the stego folder and the files of cover folders. This matching is not possible because the files that conceal the secret have been cut in the cloud environment. For each file listed in the stego folder, the spy will have to perform $\boldsymbol{O(n * (B-1))}$ browse and comparisons without succes in the files of the cover folders in the case of an exhaustive search. The matching is only performed by the receiver that holds the files of cover folders in different lists.

In the third approach, the spy has access to the cloud environment but will not be able to perform the correspondance between the files of the stego folder and the files of cover folders. This matching is not possible, because the files that conceal the secret have been cut in the cloud environment. For each file listed in the stego folder, the spy will have to perform $\boldsymbol{O(n * (B-1))}$ browse and comparisons without succes in the files of the cover folders in the case of an exhaustive search. The matching is only performed by the receiver that holds the files of cover folders in an intermediate cloud environment.

Conclusion

In this paper, we proposed three steganographic schemes distributed in a single cloud environment, which improves the work proposed by Moyou and Ndoundam[13] on the management of the problems of multi-cloud environments and the large size of the key used in the approach. In this work, the single cloud storage environment presents a set of files distributed by folder allowing to: conceal a secret message while preserving the integrity of the files that conceal the secret message, mask the presence of a communication channel during the process, reduce the size of the key by using a single cloud account credential and no file in the parameters of the key. The experiments showed that for k lists of files, n cloud accounts and a

Table 9 Comparison of our approaches and Moyou and ndoundam's approach based on the number of elements of the key and the elements of the receiver

Approaches	Number of common cloud accounts	Cloud account of the receiver	Number of common lists	Lists of the receiver
Moyou and Ndoundam approach [13]	n	0	k	0
Approach 1	1	0	0	0
Approach 2	1	0	0	n
Approach 3	1	1	0	0

base B used, we denote $n + k * B + 1$ elements in the key for Moyou and Ndoundam's approach[13], while 2 elements are only required in our approaches which are the base used and the credential of a cloud account. The work showed interesting comparisons with simpler operations to be performed by the participants during the embedding and extraction of the secret.

This work is part of the research of a distributed steganography paradigm using the concept of indirection on different multimedia files. Future improvements of the scheme will be to take no element in the key and to propose other more robust schemes in case of access of a spy in the cloud environment.

Acknowledgements
This work was supported by UMMISCO and the University of Yaounde I.

Authors' contributions
René Ndoundam conceived, designed and directed this research. Mossebo Tcheunteu Stéphane Willy and Leonel Moyou Metcheka have investigated, implemented and wrote the paper. All authors reviewed and approved the final manuscript.

Funding
This work has no funding.

Availability of data and materials
No data or models were generated during the study. However, a code wrote in C language was used to distribute the secret in the cloud handled.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 22 April 2021 Accepted: 14 July 2021
Published online: 21 August 2021

References

1. Litman J (2000) Information privacy/information property. Stanf Law Rev 52(5):1283–1313
2. Malhotra NK, Kim SS, Agarwal J (2004) Internet users' information privacy concerns (iuipc): The construct, the scale, and a causal model. Inf Syst Res 15(4):336–355
3. Gupta R, Gupta S, Singhal A (2014) Importance and techniques of information hiding: A review. arXiv preprint arXiv:1404.3063
4. AbuTaha M, Farajallah M, Tahboub R, Odeh M (2011) Survey paper: cryptography is the science of information security. Intl J Comput Sci Secur (IJCSS) 5(3). http://scholar.ppu.edu/handle/123456789/121
5. Kahn D (1996) The history of steganography. In: International Workshop on Information Hiding. Springer, Berlin. pp 1–5
6. Sharma S, Gupta A, Trivedi MC, Yadav VK (2016) Analysis of different text steganography techniques: a survey. In: 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT). IEEE. pp 130–133. https://doi.org/10.1109/cict.2016.34
7. Morkel T, Eloff JH, Olivier MS (2005) An overview of image steganography. In: Proceedings of the ISSA 2005 New Knowledge Today Conference, 29 June - 1 July 2005, Balalaika Hotel, Sandton, South Africa. ISSA, University of Pretoria, South Africa
8. Singh P (2016) A comparative study of audio steganography techniques. Int Res J Eng Technol (IRJET) 3(4):581-585
9. Mstafa RJ, Elleithy KM, Abdelfattah E (2017) Video steganography techniques: taxonomy, challenges, and future directions. In: 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT). IEEE. pp 1–6. https://doi.org/10.1109/lisat.2017.8001965
10. Koptyra K, Ogiela MR (2020) Distributed steganography in pdf files—secrets hidden in modified pages. Entropy 22(6):600
11. Bhattacharyya S (2011) A survey of steganography and steganalysis technique in image, text, audio and video as cover carrier. J Glob Res Comput Sci 2(4):1-16
12. Ahmed OM, Abduallah WM (2017) A review on recent steganography techniques in cloud computing. Acad J Nawroz Univ 6(3):106–111
13. Metcheka LM, Ndoundam R (2020) Distributed data hiding in multi-cloud storage environment. J Cloud Comput 9(1):1–15
14. Liao X, Wen Q.-y., Shi S (2011) Distributed steganography. In: 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE. pp 153–156. https://doi.org/10.1109/iihmsp.2011.20
15. Araujo II, Kazemian H, et al (2019) Enhancement of capacity, detectability and distortion of bmp, gif and jpeg images with distributed steganography. Int J Comput Netw Inf Secur (IJCNIS) 11(11):21–27
16. Zhang X, Wang S (2004) Steganography using multiple-base notational system and human vision sensitivity. IEEE Signal Process Lett 12(1):67–70
17. Wang H, Wang S (2004) Cyber warfare: steganography vs. steganalysis. Commun ACM 47(10):76–82
18. Liao X, Qin Z, Ding L (2017) Data embedding in digital images using critical functions. Signal Process Image Commun 58:146–156
19. Chan C-K, Cheng L-M (2004) Hiding data in images by simple lsb substitution. Pattern Recog 37(3):469–474
20. Liao X, Yu Y, Li B, Li Z, Qin Z (2019) A new payload partition strategy in color image steganography. IEEE Trans Circ Syst Video Technol 30(3):685–696
21. Liao X, Yin J (2018) Two embedding strategies for payload distribution in multiple images steganography. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. pp 1982–1986. https://doi.org/10.1109/icassp.2018.8462384
22. Yang J, Liao X (2020) An embedding strategy on fusing multiple image features for data hiding in multiple images. J Vis Commun Image Represent 71:102822
23. Liao X, Yin J, Chen M, Qin Z (2020) Adaptive payload distribution in multiple images steganography based on image texture features. IEEE Trans Dependable Secure Comput 1:1–1
24. Ito M, Saito A, Nishizeki T (1989) Secret sharing scheme realizing general access structure. Electron Commun Jpn (III Fundam Electron Sci) 72(9):56–64
25. Beimel A (2011) Secret-sharing schemes: A survey. In: International Conference on Coding and Cryptology. Springer, Berlin. pp 11–46
26. Gutub A, Al-Juaid N, Khan E (2019) Counting-based secret sharing technique for multimedia applications. Multimed Tools Appl 78(5):5591–5619
27. Gutub A, AlKhodaidi T (2020) Smart expansion of target key for more handlers to access multimedia counting-based secret sharing. Multimedia Tools Appl 79(25/26):1–29
28. AlKhodaidi T, Gutub A (2020) Trustworthy target key alteration helping counting-based secret sharing applicability. Arab J Sci Eng 45(4):3403–3423
29. Al-Qurashi A, Gutub A (2018) Reliable secret key generation for counting-based secret sharing. J Comput Sci Comput Math 8(4):87–101
30. Gutub A, Al-Qurashi A (2020) Secure shares generation via m-blocks partitioning for counting-based secret sharing. J Eng Res 8(3):91-117
31. Al-Ghamdi M, Al-Ghamdi M, Gutub A (2019) Security enhancement of shares generation process for multimedia counting-based secret-sharing technique. Multimedia Tools Appl 78(12):16283–16310
32. Gutub AA-A, Alaseri KA (2019) Refining arabic text stego-techniques for shares memorization of counting-based secret sharing. J King Saud Univ-Comput Inf Sci. https://doi.org/10.1016/j.jksuci.2019.06.014
33. Gutub A, Alaseri K (2020) Hiding shares of counting-based secret sharing via arabic text steganography for personal usage. Arab J Sci Eng 45(4):2433–2458
34. Alaseri K, Gutub A (2018) Int J Res Dev Organ (IJRDO) J Comput Sci Eng 4:1–17
35. Gutub A, Al-Ghamdi M (2019) Image based steganography to facilitate improving counting-based secret sharing. 3D Res 10(1):6
36. Gutub A, Al-Ghamdi M (2020) Hiding shares by multimedia image steganography for optimized counting-based secret sharing. Multimedia Tools Appl 79(11/12):1–35

37. Tan S, Li B (2012) Targeted steganalysis of edge adaptive image steganography based on lsb matching revisited using b-spline fitting. IEEE Signal Process Lett 19(6):336–339
38. Tang W, Li H, Luo W, Huang J (2014) Adaptive steganalysis against wow embedding algorithm. In: Proceedings of the 2nd ACM Workshop on Information Hiding and Multimedia Security. pp 91–96. https://doi.org/10.1145/2600918.2600935
39. Wu S, Zhong S, Liu Y (2018) Deep residual learning for image steganalysis. Multimedia Tools Appl 77(9):10437–10453
40. Chiew KL, Pieprzyk J (2010) Blind steganalysis: A countermeasure for binary image steganography. In: 2010 International Conference on Availability, Reliability and Security. IEEE. pp 653–658. https://doi.org/10.1109/ares.2010.66
41. Munteanu VI, Şandru C, Petcu D (2014) Multi-cloud resource management: cloud service interfacing. J Cloud Comput 3(1):1–23
42. Georgios C, Evangelia F, Christos M, Maria N (2021) Exploring cost-efficient bundling in a multi-cloud environment. Simul Model Pract Theory 111:102338
43. Heilig L, Lalla-Ruiz E, Voß S (2016) A cloud brokerage approach for solving the resource management problem in multi-cloud environments. Comput Ind Eng 95:16–26

Publisher's Note