Distributed data hiding in multi-cloud storage environment

Leonel Moyou Metcheka, René Ndoundam

HAL Id: hal-03104562

https://hal.sorbonne-universite.fr/hal-03104562v1

Submitted on 9 Jan 2021

RESEARCH   **Open Access**

Distributed data hiding in multi-cloud storage environment

Leonel Moyou Metcheka[1,2,3] and René Ndoundam[1,2,3*]

Abstract

Classical or traditional steganography aims at hiding a secret in cover media such as text, image, audio, video or even in network protocols. Recent research has improved this approach called distributed steganography by fragmenting the secret message and embedding each secret piece into a distinct cover media. The major interest of this approach is to make the secret message detection extremely difficult. However, these file modifications leave fingerprints which can reveal a secret channel to an attacker. Our contribution is a new steganography paradigm transparent to any attacker and resistant to the detection and the secret extraction. Two properties contribute to achieve these goals: the files do not undergo any modification while the distribution of the secret in the multi-cloud storage environment allows us to hide the existence of the covert channel between the communicating parties. Information's are usually hidden inside the cover media. In this work, the covert media is a pointer to information. Therefore the file carries the information without being modified and the only way to access it is to have the key. Experiments show interesting comparison results with remarkable security contributions. The work can be seen as a new open direction for further research in the field.

Keywords: Distributed steganography, Multi-cloud environment, Multimedia files, Data integrity

Introduction

The rapid development of internet leads to an increase dependence in almost all areas of life. In the same way that it promotes communications, it also poses a threat both to users and to state institutions with the spying and theft of information. To tackle this annoyance, two security techniques are used: steganography and cryptography. The main difference between the two is the secret access mechanism. One goes unnoticed while the second is unreadable by transformations [1].

Steganography is mainly used for secret communications by setting up a cover channel [2]. Steganography is the art of writing secret data so that no one except the recipient is aware of the existence of the secret message [3]. Popular steganography techniques hide the secret in digital content such as text, image, video and audio files [4–7]. While other techniques insert the secret in the

network protocols [8]. The study of steganographic systems is evaluated with three main characteristics: capacity, robustness and security. Capacity measures the volume of secret data that is hidden in the cover media. Robustness concerns resistance to destruction or modification of the hidden data (steganogram). While security assesses the ability of an eavesdropper to detect hidden information. Although, security is usually the most desirable characteristic [9].

A successful steganography depends on the carrier medium that does not raise attention [10]. This is why the best carrier choice for steganogram must be popular as well as transparent when inserting the steganogram [11]. Thus, the cloud is an ideal candidate for enabling secret communications. A number of steganographic techniques are used to transfer and secure the data stored in the cloud storage [12]. These techniques generally respond to the problems of confidentiality of user data stored in cloud servers [13].

These methods refer to classical steganography compared to distributed steganography [14] which aims to

* Correspondence: ndoundam@yahoo.com
[1]Team GRIMCAPE, Yaounde, Cameroon
[2]Sorbonne University, IRD, UMMISCO, F-93143 Bondy, France
Full list of author information is available at the end of the article

hide the secret in several covert medium instead of one. The distributed steganography has the merit of making difficult the secret message detection. This is only possible by a meticulous modification of each cover medium. However, these modifications leave fingerprints which can reveal a secret channel by an attacker [15, 16]. Moreover, the simple fact that the attacker knows that there exists an exchange between the two can raise suspicion. When these scenarios are realized the whole steganographic model collapses. Therefore, here is an overview of the usual features of classical steganography:

- The communicating parties are known: the files are sent from one to the other [4, 17, 18];
- The transferred files are altered for secret insertion: the embedding and extraction process use different approaches such as a special character like non breaking space(A0), punctuation marks, word synonyms and linguistic properties [4, 17–19];
- The exchanged files are commonly subject to steg analysis: the covert channel establishment will be detected [15, 16, 20–22]. Consequently, the deletion or modification of the covert media leads to the loss of the entire secret;

In this paper, our contribution consists in proposing a new distributed steganography scheme based on the processed files integrity. Thus, it guarantees that the exchanged files do not undergo any modification. This ensures a good security level with an undetectable secret communication. The technique used exploits file storage in several cloud service providers. This solution is an excellent tool useful for the organizations for data exfiltration in case of espionage or to keep secure the participants shared keys involved in a secret sharing. This work contributions are summarized as follow:

- The communicating parties are not known: there is no direct link between them during the communication process. One uploads files in the cloud storage while the second exploits these files;
- The transferred files are not altered for secret insertion: each file implicitly holds a part of the secret data;
- The exchange files are robust against steg analysis: the proposed technique focuses on maximum resiliency against secret detection and extraction.
- The ability to use any file extension to establish the covert channel while maintaining their integrity.

The rest of the paper is organized as follows: section 2 and section 3 study classical and distributed steganography respectively. The new distributed scheme is presented in section 4. Experimental results are done in section 5 and finally section 6 is devoted to the conclusion.

Classical steganography

In this section, we will revisit some shortcomings related to classical steganography.

In 1984, Simmons presents the interest of steganography through the prisoner's problem [23], where Alice and Bob plan to escape from prison. Their communication goes through the warden who searches for any hidden communication between the two and if he detects one, he will separate them and cause their failure to escape. Then, the two prisoners must use ingenuity to keep their communication undetectable. Alice and Bod must use a channel that is invisible to the warden. This secret channel is known as the covert channel. Figure 1 shows the security model applied to cryptography [24]. This communication conveys secret information in a manner that it can be observed by an adversary. It uses the unsecured regular communication channel.

The general process of steganography achieves unsuspecting communications with two current algorithms: the embedding algorithm which consists in altering the cover media (text, image, video, audio or network protocol) to insert the secret message using a key shared between the parties in communication. This produced a stego media. The second is the extracting algorithm which consists in dissociating the covert message from the media carrier according to the stego media and the secret key. The extraction is said to be reversible when the cover media obtained is identical to the initial.

The main concern of steganography is stealth, because if an attacker, passive or active, can detect the presence of the secret message, from there, he can try to extract it and to decrypt it, if it is encrypted. Thus, steganography techniques must focus on maximum resiliency against detection and extraction. In contrast, two unexpected actions can militate in favour of the detection and extraction of the message. The first is the attacker's ability to know the existence of a communication between Alice and Bob. The stego media is generally sent to Bob via an open or insecure channel. The simple fact that the attacker knows there exists an exchange between the two can raise suspicion. The second action is the attacker's ability to capture and to study in depth the content of the messages exchanged to reveal the existence of a covert channel. The attacker can achieve this objective by performing steg analysis and finally detects or extracts the secret message. The attacker can even go further by deactivating the hidden message so that the recipient cannot extract it and / or modify the hidden message to send incorrect information to the recipient [25].

Fig. 1 The model of a two-party communication using encryption [24]
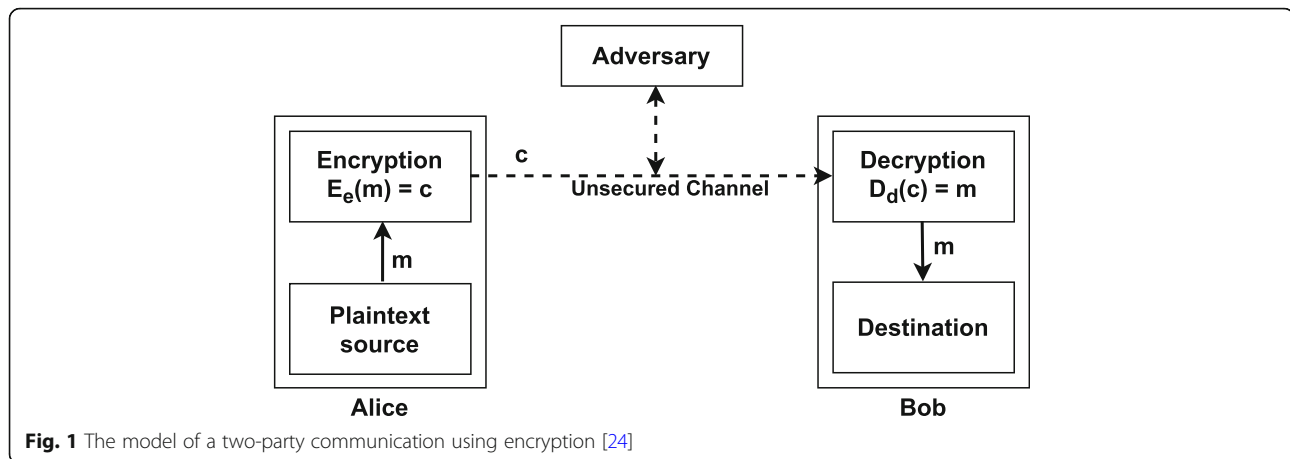
When these scenarios are realized the whole steganographic model collapses. These covert channel detection are motivated and carried out mainly by computer forensic examiners which collect evidence related to a past crime. But also by secret government and corporate services to prevent espionage.

Some steganography detection tools on image and audio files have been described using advanced statistical tests [26–29] such as higher-order statistics, Markov random fields, linear analysis, wavelet statistics, and much more [20]. Regarding data hiding in network communications, several studies are capable of detecting covert channels while others prevent all secret communications. Goudar's work [30] uses second-order statistical tools such as the adjacency histogram and the normalized adjacency histogram to detect secret communications. While Muawia's work [31] uses crafting and replaying packet tools to disable hidden messages.

Distributed steganography

In this section, we will revisit some shortcomings related to distributed steganography.

Distributed steganography [32, 33] is an improvement of classical steganography which aims at fragmenting a secret and then hidden in several covert media, which makes it more difficult to detect the whole secret message. This is applied when there are various independent senders and only one recipient. Thus the receiver acquires the union of distinct inputs. With the advent of cloud technologies, it is common nowadays for users to hide secret information in a mass of images and stored them in the cloud space. These stego images are then shared with the recipient to extract the secret. In general, these embedding algorithms deal with the issue of distributing the payload in a sequence of images to avoid

detection. This explains the emergence of embedding strategies of payload distribution in multiple images by fusing multiple features to describe image complexity [14]. Other recent strategies are based on the image texture complexity and the distortion distribution as indicator for secure capacity of each cover image [34]. These strategies are applied on single image steganographic algorithms and experiments shown better resistance to modern universal pooled steganalysis compared to existing methods.

Xin Liao et al. [35] have proposed a model of the distributed steganography (see Fig. 2), where n sending parties p_1, p_2, ...,p_n wishes to establish a covert channel with a receiver R. Each party only knows its covert message m_i while no one else apart from the receiver R can retrieve the combination of these secrets messages by receiving it through a public channel. The fact of protecting a secret data through several people is named as secret sharing [36]. Secret sharing schemes require three main phases: generation of the target key, distribution of the share keys to participants and reconstruction of the secret. The target key can thus be obtained from a set of n shares determined automatically by the dealer of the system. The latter distributes a share key to each participant via a private channel. Finally, the system combines the different share keys to access the system from the target key obtained [37]. This principle was originally proposed by Shamir [38] and Blakley [39]. Secret sharing is applied in critical areas requiring access controlled by multiple users such as rocket launching, opening of bank safes, proof correctness of electronic voting systems [40].

Improvements in secret sharing have been proposed for allowing access to a restricted number of participants, called (k, n) secret sharing. The k value is known as the secret share threshold and must be less than or equal to n. As a constraint, the

Fig. 2 The model of distributed steganography [34]
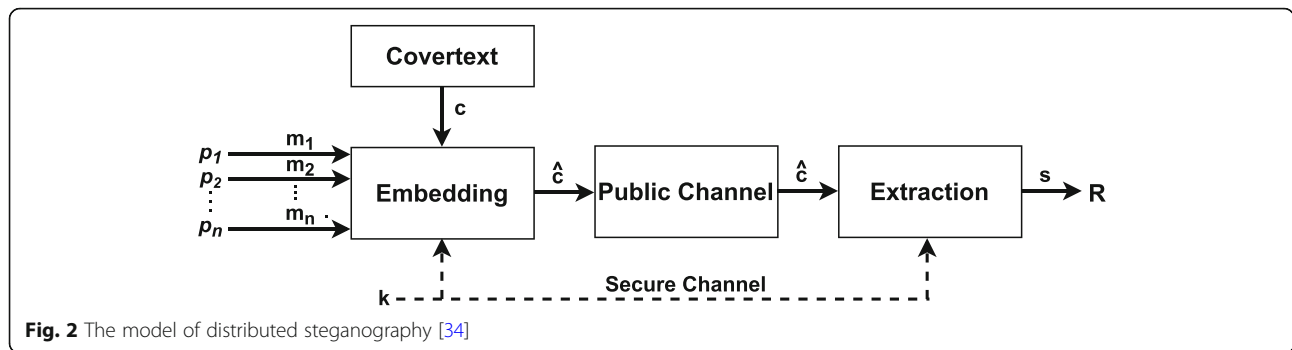
reconstruction of the target key requires at least k participants [41]. One of the approaches which is inspired by this principle is the counting based secret sharing [40], consisting in generating the share keys by replacing at various positions of the secret key one or two 0-bits by a 1-bit. Therefore, the reconstruction of the target key is obtained by adding bit by bit the share keys of the same position, so that the result bit is equal to 1 if the sum is equal to the value of the threshold k and 0 otherwise. This technique requires less computation as a strength, however it generates a reduced number of shares. Several optimizations have been made to guarantee the security of shared keys [42] as well as to enhance the number of generated shares [43, 44]. These schemes are excellent tools useful for cryptographic protocols. Moreover, they are also used in steganography to hide in a distributed manner the share keys of each participant. To this end, methods are provided for concealing the target key in specific covert media such as text [45, 46] or images [47, 48].

Distributed steganography is interesting because it makes detection task more complicated by spreading the secret in various media and store them in random places. However, simply modifying a media to incorporate a secret can raise attention for establishing a covert channel. This can be noted by performing steg analysis [49, 50]. More seriously, if the suspicious is proved to be true, the deletion or modification of a single covert media can lead to the loss of the entire secret.

To illustrate these limits we will take images considered as one of the most used media in steganography. If an algorithm can determine the presence of a secret message in a covert media then the whole steganographic system used is considered broken [28]. However, it is hardly practical for a steganalizer to know the algorithm used in all the variety of existing steganographic system. This is why universal methods emerge named as blind steganalisis, are capable of

detecting any new or unknown embedding algorithm [15, 16]. The process is performed in two main phases: a training phase using original images with features extraction and a second phase for the images classification. This technique detects original images and stego images. It motivates us to develop a steganographic method perfectly undetectable with current steganalized methods by exchanging covert media without modifying them. In addition, the covert channel is capable of handling all types of covert media. Table 1 shows a comparison of steganographic techniques based on two criteria: the covert media type used as well as the modification of the cover media.

Table 1 Steganographic techniques comparaison based on the covert media type and covert media modification

Classical Steganography					
References	**Text**	**Image**	**Audio**	**Video**	**Covert Media Modification**
Liu et al. [51] Lee et al. [19] Ekodeck et al. [4] Khosravi et al. [17]	✓	✗	✗	✗	✓
Sahu et al. [5] Su et al. [52]	✗	✓	✗	✗	✓
Jiang et al. [6] Ali et al. [53]	✗	✗	✓	✗	✓
Pilania et al. [7] Baziyad et al. [54]	✗	✗	✗	✓	✓
Distributed Steganography					
References	Text	Image	Audio	Video	Covert Media Modification
Gutub et al. [45] Gutub et al. [46]	✓	✗	✗	✗	✓
Yang et al. [14] Liao et al. [34] Gutub et al. [47] Gutub et al. [48]	✗	✓	✗	✗	✓
Our proposal	✓	✓	✓	✓	✗

The comparison ends with the positioning of the method to be proposed.

The proposed scheme

The proposed covert channel is a new paradigm, transparent to secret communication between the two parties. The solution uses the cloud storage space to store files. The uploaded files undergo no alteration, the information associated with each of them is their classification order in a list of files. Thereby, the file selection and uploading to the cloud depends on the secret to be shared with the receiver.

The original idea of steganography was proposed by Simmons [23]. The basic idea was to hide secret data inside the cover media to go unnoticed. Related works [4–7] based the design of their steganographic schemes on this idea.

The original contribution is the proposal of a steganographic scheme where the files used as covert media carry the information without being modified. The cover media is a pointer to secret data. This secret data is found in the key. This key consists of the following sets: the list of clouds and login credentials, the lists of files and the base used. The sender and the receiver exchange this key before initiating their secret communication. The keys exchange can be done in a physical meeting or using encrypted communications. As a concrete example, a secret agent C is employed by his origin country A. During a physical meeting in the country's government agency A, the agency officials give to the secret agent a removable memory which contains the key. Country A sends the secret agent in the country B. During his spy mission in country B, the secret agent sends confidential information from country B to country A.

Overview

The communication model proposed in Fig. 3, shows the sender and the receiver sharing identical lists of documents. The sender transcodes the secret in a specific base and group them into k blocks of n values: b_0, b_1, ..., b_{k-1}. For each value of a given bloc, the files at this index value are sent to the cloud. The process is repeated for every secret block. To each block is assigned a new list. The receiver having the same access to each cloud, browses them to recover the saved files. Thereafter, he reconstructs the secret from these file positions found in the lists. The originality of this scheme is the opponent disability to make any communication link between the sender and the receiver.

The covert channel model
The covert object
These are any extension files, selected to be dropped in multiple clouds storage environment. The cloud storage providers used are named as: c_0, c_1, ..., c_{n-1}, $n \geq 2$.

The embedded message
Any message format can be concealed. The preliminary step requires that the secret be encoded in a given base.

The key
Three elements are shared between the sender and the receiver:

- The cloud order c_0, c_1, ..., c_{n-1};
- The authentication accounts (user name and password) for cloud access named as: w_0, w_1, ..., w_{n-1};
- A set of disjointed lists $L^{(0)}$, $L^{(1)}$, ..., $L^{(k-1)}$, where each list i contains exactly B files: $L_0^{(i)}, L_1^{(i)}, ..., L_{B-1}^{(i)}$, $i =$
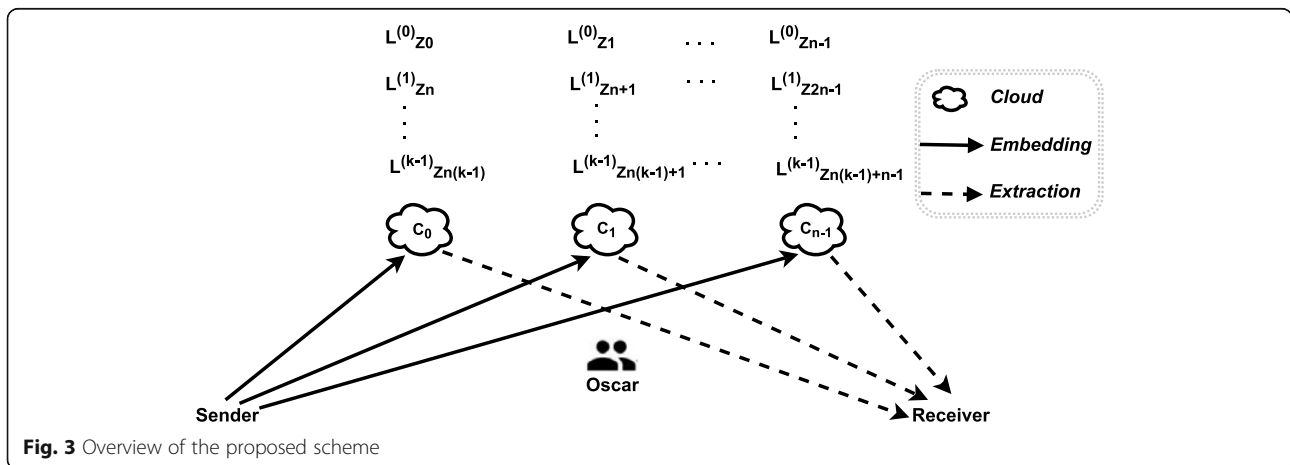
Fig. 3 Overview of the proposed scheme

0,1, ..., k-1. These files can take any format type such as text, image, audio, video, application, archive, ...

- The base B such that: $|L^{(0)}| = |L^{(1)}| = ... = |L^{(k-1)}| = B$.

Notations and hypothesis

These notations are useful for the embedding algorithm as well as for the secret extraction:

- s: the input secret formatted in base *2* or *10*;
- B: the base used such that: $B \geq 2$;
- $(z_{q-1} ... z_1 z_0)_B$: is the secret representation in base B;
- *Mat [i]*: is the ith block of the secret;
- *Mat [i,j]*: is the value at position j of the block number i;
- n: is the number of clouds handled;
- k: is the secret bloc number or the number of lists used;
- $L^{(i)}$: is the ith files list. Each secret block uses a distinct list i, $0 \leq i \leq k-1$;
- $L_j^{(i)}$: is the jth file in the list number i, $0 \leq i \leq k-1$ and $0 \leq j \leq B-1$;

Here are hypothesis of the proposed scheme:

- Lists:$L^{(0)}, L^{(1)}, ..., L^{(k-1)}$ are disjointed:
  i_1, i_2 designating Lists, $0 \leq i_1, i_2 \leq k-1$,
  j_1, j_2 designating files, $0 \leq j_1, j_2 \leq B-1$,
  if $i_1 \neq i_2$ then $L_{j_1}^{(i_1)} \neq L_{j_2}^{(i_2)}$.

Embedding algorithm

This is done by performing the following steps in the sender side. The corresponding flowchart is presented in Fig. 4.

1) The secret is converted in base B
2) The secret representation in base B is split in block of n values;
3) For each secret block;
   a) Open the first cloud storage with the associated user name and password;
   b) For each value of the block.
      i. Find in the list the file having these value index;
      ii. Send to the cloud the related file ;
      iii. Open the next cloud storage with the associated user name and password.
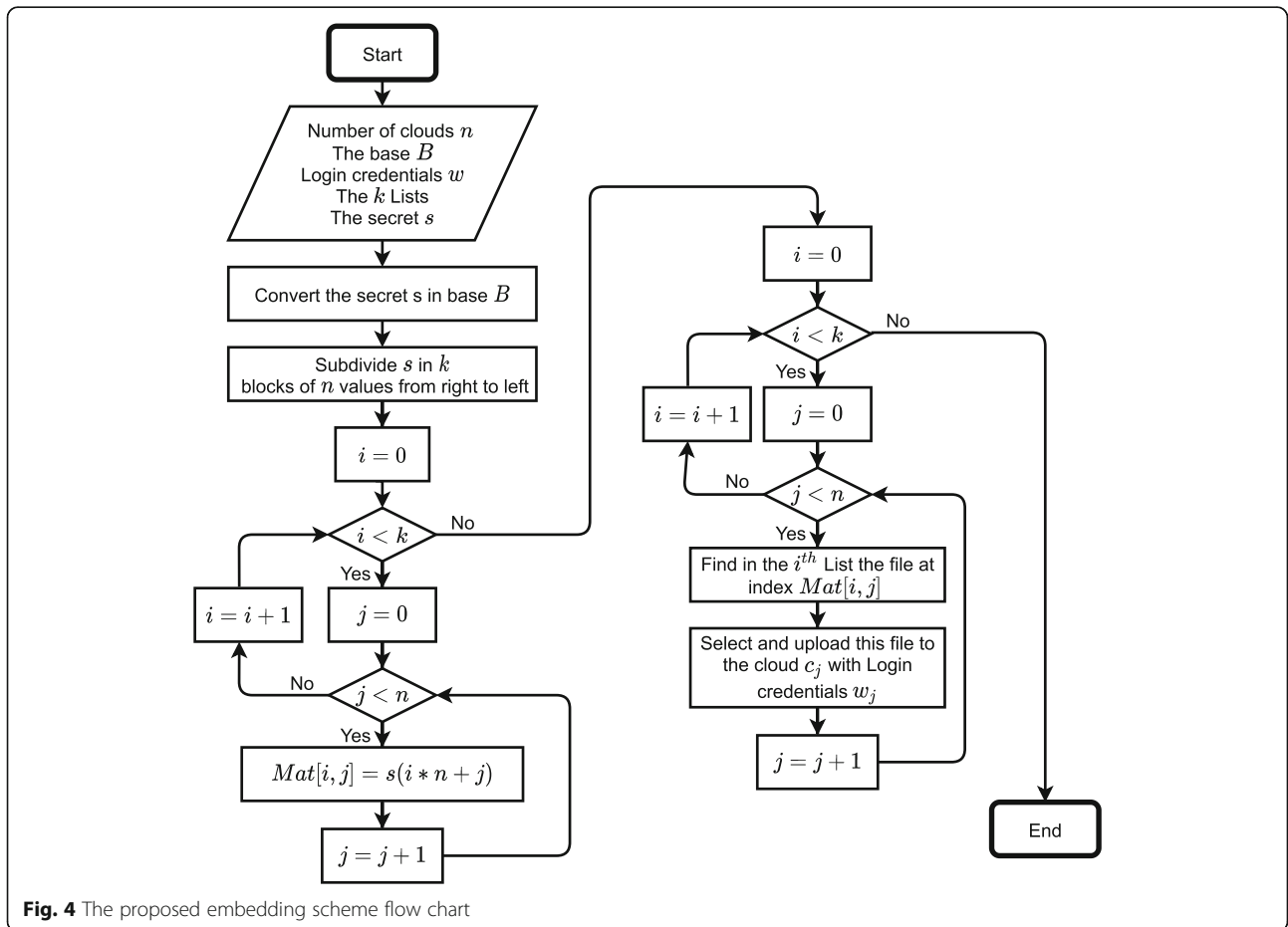
Fig. 4 The proposed embedding scheme flow chart

The embedding algorithm is as follows:

Input

C: the set of clouds;

W: the set of cloud authentication accounts;

L: the set of files lists;

B: the base;

s: the secret message;

Output

C': the cloud lists after file storage;

Begin

1. Convert the secret s to base B such that:$s = (z_{q-1}...z_1 z_0)_B$, where $0 \le z_i \le B\text{-}1$;

2. Split s in k blocks of length n stored in the matrix *Mat* such that :

   $Mat[i,j] = s[(i \times n) + j], 0 \le i \le k\text{-}1$ and $0 \le j \le n\text{-}1$;

3. For each block of *Mat* : $0 \le i \le k\text{-}1$:

   3.1. For each value $Mat[i,j]$: $j = 0,1,...,n-1$:

      3.1.1. $u = Mat[i,j]$;

      3.1.2. Find in the ith list $L^{(i)}$ the files at index $Mat[i,j]$;

      3.1.3. Select and upload to the cloud c_j the file $L_u^{(i)}$.

End

Extraction algorithm

This is done by performing the following steps in the receiver side. The corresponding flowchart is presented in Fig. 5.

1. Browse each managed cloud storage;
   a. Authenticate with the user name and password;
   b. Recover the files found in the list shared between the sender and the receiver;
   c. Determine the indexes of these files;
   d. Sort the indexes in the numbering order of the lists where they come from;
   e. Store the indexes by column in a matrix;
2. Use the secret base value B to retrieve the secret in decimal stored in the matrix;
3. Transcode the secret from the decimal representation to binary;
4. Delete from the clouds all the files used to recover the secret.

The extraction algorithm is as follows:

Input

C: the set of clouds;

W: the set of cloud authentication accounts;

L: the set of filelists;

B: the base;

Output

s: the secret message;

Begin

1. For each cloud c_j , $j = 0, 1, ..., n-1$:

   1.1. Extract to the cloud c_j the files contained in the lists $L^{(0)}, L^{(1)}, ..., L^{(k-1)}$;

   1.2. Find the indexes associated to these files named as:$z_0, z_1, ..., z_{k-1}$;

   1.3. For each block $i = 0, 1, ..., k-1$:

      1.3.1. $Mat[i,j] = z_i$;

2. Compute $m = \sum_{i=0}^{k-1} \sum_{j=0}^{n-1} (Mat[i,j] \times B^{(i \times n)+j})$;

3. Convert m to binary and get the secret message s.

4. Remove from the clouds all the files used to recover the secret.

End

Time complexity analysis

In this subsection, we investigate the time complexity analysis of the proposed steganographic scheme. We assume that we have a secret s to distribute between n cloud storages using the base value B. We also assume that the secret s is hidden using k file lists, each containing B files. The proposed embedding scheme converts the secret s in base B in $O(log_B(s))$. The subdivision of s into blocks and the choice of files to be stored in the cloud are done in $O(n*k)$. Consequently, the embedding scheme time complexity is $O(n*k)$.

In the proposed extraction scheme, we assume that the cloud contains m files. The files extraction from the cloud contained in the lists is done in $O(m*k*B)$. Moreover, the secret is converted to decimal in $O(n*k)$. Finally, the secret decimal value conversion to base 2 is done in $O(log_2 (s))$. Therefore, the time complexity of the secret extraction scheme is $O(m*k*B)$.

Evaluation

In order to assess the performance of the proposed method, a theoretical estimation of the hidden bits in multi-cloud storage environment is given. Then experiments will show in detail the steps necessary to realize the covert channel. Finally, discussion and security analysis are performed on the proposed scheme.

Hidden secret bits estimation in the clouds storage

The focus here is to hide secret bits in a set of n cloud. Each cloud embeds a value in base B, and this value can vary from 0 to $B\text{-}1$, so B possibilities. Then for a set of n clouds, there is B^n possibilities. So, for a secret with k blocks, the number of hidden bits is:

$$k \times log_2(B^n) = k \times n \times log_2(B).$$

Example

To describe our proposed data hiding scheme, simple numerical examples are detailed below. In this examples, s = 1, 111,101,101,000,001 a 16-bit secret and the number of managed clouds is set to four ($n = 4$). The storage providers used and their respective IDs are: SugarSync (c_0), Dropbox (c_1), OneDrive (c_2) and Google Drive (c_3). Table 2 shows associated cloud login credentials. The four lists $L^{(0)}$, $L^{(1)}$, $L^{(2)}$ and $L^{(3)}$ used to embed the secret are presented in Table 3. Then, four scenarios are highlighted with the base taking these successive values: $B = 2$, $B = 4$, $B = 9$ and $B = 17$. Each case presents the secret distribution between these cloud storage environments and explains in detail how the secret is embedded and extracted using file lists. Note that the file lists, the base value, the set of clouds and their login credentials are the key, shared between the sender and the receiver.
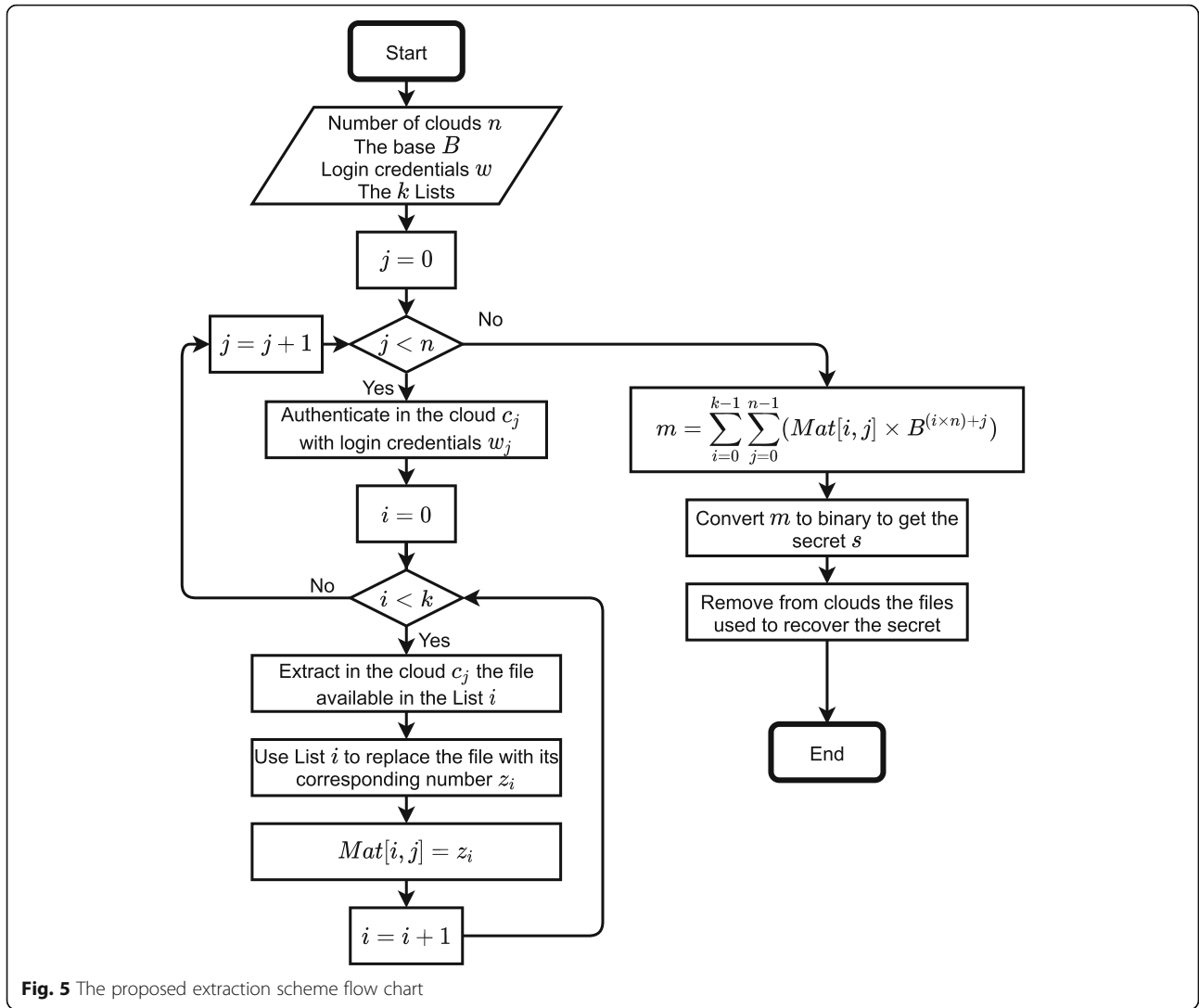
Fig. 5 The proposed extraction scheme flow chart

Case 1: s = 1,111,101,101,000,001, n = 4 and B = 2

Let's follow these steps to embed the secret key:

- Step 1: The secret is already represented in base 2, $s = (1111101101000001)_2$;
- Step 2: The secret is subdivided into groups of 4 bits, because of the four clouds available. From right to left this gives 4 blocks: 0001 0100 1011 1111;
- Step 3: For each block, each bit is linked to a distinct cloud in the order c_0, c_1, c_2 and c_3:

c_0	c_1	c_2	c_3
1	0	0	0
0	0	1	0
1	1	0	1
1	1	1	1

Bloc #0				Bloc #1				Bloc #2				Bloc #3			
0	0	0	1	0	1	0	0	1	0	1	1	1	1	1	1
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
c_3	c_2	c_1	c_0	c_3	c_2	c_1	c_0	c_3	c_2	c_1	c_0	c_3	c_2	c_1	c_0

- Step 4: Bits of the same cloud are grouped together. Therefore, the secret parts of each cloud are:

Table 2 The set of 4 cloud service providers handled and their login credentials

Code	Cloud Name	Login	Password
c_0	SugarSync	userlogin0@gmail.com	User-pwd0
c_1	Dropbox	userlogin1@gmail.com	User-pwd1
c_2	OneDrive	userlogin2@gmail.com	User-pwd2
c_3	Google Drive	userlogin3@gmail.com	User-pwd3

Table 3 The four file lists and their index number

	(a) $L^{(1)}$		(b) $L^{(2)}$		(c) $L^{(3)}$		(d) $L^{(4)}$
0	thesis.docx	0	scheduling.xlsx	0	conference.pptx	0	dataHiding.pdf
1	article.docx	1	statistics.xlsx	1	results.pptx	1	cryptography.pdf
2	balanceSheet.docx	2	budget.xlsx	2	slideshow.pptx	2	deepLearning.pdf
3	report.docx	3	data.xlsx	3	marketing.pptx	3	linearAlgebra.pdf
4	meeting.docx	4	bill.xlsx	4	management.pptx	4	dataScience.pdf
5	opportunities.docx	5	Evaluation.xlsx	5	slides.pptx	5	publications.pdf
6	lesson.docx	6	gradebook.xlsx	6	animation.pptx	6	sourceCode.pdf
7	chapter.docx	7	stocks.xlsx	7	overview.pptx	7	dataAnalysis.pdf
8	introduction.docx	8	simulation.xlsx	8	speech.pptx	8	modelingLife.pdf
9	tutorial.docx	9	project.xlsx	9	seminar.pptx	9	cloudComputing.pdf
10	redaction.docx	10	analysis.xlsx	10	symposium.pptx	10	masterDegree.pdf
11	news.docx	11	curves.xlsx	11	resume.pptx	11	bachelorDegree.pdf
12	book.docx	12	quotation.xlsx	12	shopping.pptx	12	bithdayCerticate.pdf
13	exercise.docx	13	finance.xlsx	13	accounts.pptx	13	passport.pdf
14	anthem.docx	14	classes.xlsx	14	clinical.pptx	14	human.pdf
15	journal.docx	15	salaries.xlsx	15	aviation.pptx	15	contacts.pdf
16	editor.docx	16	phonebook.xlsx	16	audition.pptx	16	awards.pdf

- Step 5: The four lists of Table 3 are used to hide the four secret blocks. Hide respectively the 1st, 2nd, 3rd and 4th row of the matrix obtained in step 4 with the list $L^{(0)}$, $L^{(1)}$, $L^{(2)}$ and $L^{(3)}$. More specifically, each value is replaced by the file having this index in the corresponding list. These files act as pointers to the data to be kept secret. The stego files to be uploaded in each cloudare allocated as follows:

List	Cloud c_0	Cloud c_1	Cloud c_2	Cloud c_3
$L^{(0)}$	article.docx	thesis.docx	thesis.docx	thesis.docx
$L^{(1)}$	scheduling.xlsx	scheduling.xlsx	statistics.xlsx	scheduling.xlsx
$L^{(2)}$	results.pptx	results.pptx	conference.pptx	results.pptx
$L^{(3)}$	cryptography.pdf	cryptography.pdf	cryptography.pdf	cryptography.pdf

- Step 6: The last embedding step is to transfer the files article.docx, scheduling.xlsx, results.pptx and cryptography.pdf to the cloud c_0; thesis.docx, scheduling.xlsx, results.pptx and cryptography.pdf to the cloud c_1; thesis.docx, statistics.xlsx, conference.pptx and cryptography.pdf to the cloud c_2; thesis.docx, scheduling.xlsx, results.pptx and cryptography.pdf to the cloud c_3.

Let's follow these steps to extract the secret:

- Step 1: The files of each cloud are compared to those available in the four lists $L^{(0)}$, $L^{(1)}$, $L^{(2)}$ and $L^{(3)}$.

When the names are identical, these files are retrieved and sorted in ascending order of list numbering. The files extracted by cloud and by list are as follows:

Cloud	$L^{(0)}$	$L^{(1)}$	$L^{(2)}$	$L^{(3)}$
c_0	article.docx	scheduling.xlsx	results.pptx	cryptography.pdf
c_1	thesis.docx	scheduling.xlsx	results.pptx	cryptography.pdf
c_2	thesis.docx	statistics.xlsx	conference.pptx	cryptography.pdf
c_3	thesis.docx	scheduling.xlsx	results.pptx	cryptography.pdf

- Step 2: The files in each list are then replaced by their number. The sequence of each cloud obtained is:

Cloud	$L^{(0)}$	$L^{(1)}$	$L^{(2)}$	$L^{(3)}$
c_0	1	0	1	1
c_1	0	0	1	1
c_2	0	1	0	1
c_3	0	0	1	1

- Step 3: Each binary sequence belonging to a cloud is stored in column inside a matrix called *Mat*:

$$Mat = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

- Step 4: Compute m, the secret in decimal using the base value (B = 2). The variables i and j respectively scan the rows and columns of the matrix *Mat*. The conversion is done as follows:

$$m = \sum_{i=0}^{3} \sum_{j=0}^{3} Mat[i,j] \times 2^{(i \times 4)+j}$$

$$= 1*2^0 + 0*2^1 + 0*2^2 + 0*2^3 + 0*2^4 + 0*2^5 + 1*2^6 + 0*2^7 +$$

$$1*2^8 + 1*2^9 + 0*2^{10} + 1*2^{11} + 1*2^{12} + 1*2^{13} + 1*2^{14} + 1*2^{15}$$

$$= 1 + 64 + 256 + 512 + 2\,048 + 4\,096 + 8\,192 + 16\,384 + 32\,768$$

$$= 64,321$$

- Step 5: The secret s is obtained by converting m to base 2: $(64321)_{10} = (1111101101000001)_2$
- Step 6: All the files retrieved in step 1 of extraction are removed from the cloud storage.

Case 2: s = 1,111,101,101,000,001, n = 4 and B = 4
In this case, the secret s and the number of clouds n remain unchanged. The base considered is B = 4. Let's follow these steps to embed the secret:

- Step 1: The secret is converted to base 4:

$$(1111101101000001)_2 = (64321)_{10} = (33231001)_4;$$

- Step 2: The secret is subdivided into groups of 4 values, because of the four clouds available. From right to left this gives 2 blocks: 1001 3323
- Step 3: For each block, each value is linked to a distinct cloud in the order c_0, c_1, c_2 and c_3:

Bloc #0				Bloc #1			
1	0	0	1	3	3	2	3
↓	↓	↓	↓	↓	↓	↓	↓
c_3	c_2	c_1	c_0	c_3	c_2	c_1	c_0

- Step 4: Values of the same cloud are grouped together. Therefore, the secret parts of each cloud are:

c_0	c_1	c_2	c_3
1	0	0	1
3	2	2	3

- Step 5: Two lists of Table 3 are used to hide the two secret blocks. Hide respectively the 1st and 2nd row of the matrix obtained in step 4 with the list $L^{(0)}$ and $L^{(1)}$. Each value is replaced by the file having this index in the corresponding list. The stego files to be uploaded in the clouds are allocated as follows:

List	Cloud c_0	Cloud c_1	Cloud c_2	Cloud c_3
$L^{(0)}$	article.docx	thesis.docx	thesis.docx	article.docx
$L^{(1)}$	data.xlsx	budget.xlsx	data.xlsx	data.xlsx

- Step 6: The last embedding step is to transfer the files *article.docx* and *data.xlsx* to the cloud c_0; *thesis.docx* and *budget.xlsx* to the cloud c_1; *thesis.docx* and *data.xlsx* to the cloud c_2; *article.docx* and *data.xlsx* to the cloud c_3.

Let's follow these steps to extract the secret:

- Step 1: The files of each cloud are compared to those available in the two lists $L^{(0)}$ and $L^{(1)}$. When the names are identical, these files are retrieved and sorted in ascending order of list numbers. The files extracted by cloud and by list is as follows:

Cloud	$L^{(0)}$	$L^{(1)}$
c_0	article.docx	data.xlsx
c_1	thesis.docx	budget.xlsx
c_2	thesis.docx	data.xlsx
c_3	article.docx	data.xlsx

- Step 2: The files in each list are then replaced by their number. The sequence of each cloud obtained is:

Cloud	$L^{(0)}$	$L^{(1)}$
c_0	1	3
c_1	0	2
c_2	0	3
c_3	1	3

- Step 3: Each sequence belonging to a cloud is stored in column inside the matrix *Mat*:

$$Mat = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 3 & 2 & 3 & 3 \end{pmatrix}$$

- Step 4: Compute m, the secret in decimal using the base value (B = 4). The conversion is done as follows:

$$m = \sum_{i=0}^{1} \sum_{j=0}^{3} Mat[i,j] \times 4^{(i \times 4)+j}$$

$$= 1*4^0 + 0*4^1 + 0*4^2 + 0*4^3 + 3*4^4 + 2*4^5 + 3*4^6 + 3*4^7$$

$$= 1 + 64 + 768 + 2\,048 + 12\,288 + 49\,152$$

$$= 64,321$$

- Step 5: The secret s is obtained by converting m to base 2: $s = (64321)_{10} = (1111101101000001)_2$
- Step 6: All the files retrieved in step 1 of extraction are removed from the cloud storage.

Case 3: s = 1,111,101,101,000,001, n = 4 and B = 9
In this third case, the secret s and the number of clouds n remain unchanged. The base considered is B = 9. Let's follow these steps to embed the secret:

- Step 1: The secret is converted to base 9:

$$(1111101101000001)_2 = (64321)_{10} = (107207)_9;$$

- Step 2: The secret is subdivided into groups of 4 values, because of the four clouds available. From right to left this gives 2 blocks: 7207 10.
- Step 3: For each block, each value is linked to a distinct cloud in the order c_0, c_1, c_2 and c_3:

Bloc #0			Bloc #1		
7	2	0	7	1	0
↓	↓	↓	↓	↓	↓
c_3	c_2	c_1	c_0	c_1	c_0

- Step 4: Values of the same cloud are grouped together. Therefore, the secret parts of each cloud are:

c_0	c_1	c_2	c_3
7	0	2	7
0	1		

- Step 5: Two lists of Table 3 are used to hide the two secret blocks. Hide respectively the 1st and 2nd row of the matrix obtained in step 4 with the list $L^{(0)}$ and $L^{(1)}$. Each value is replaced by the file having this index in the corresponding list. The stego files to be uploaded in the clouds are allocated as follows:

List	Cloud c_0	Cloud c_1	Cloud c_2	Cloud c_3
$L^{(0)}$	chapter.docx	thesis.docx	balanceSheet.docx	chapter.docx
$L^{(1)}$	scheduling.xlsx	statistics.xlsx		

- Step 6: The last embedding step is to transfer the files *chapter.docx* and *scheduling.xlsx* to the cloud c_0; *thesis.docx* and *statistics.xlsx* to the cloud c_1; *balanceSheet.docx* to the cloud c_2 and *chapter.docx* to the cloud c_3.

Let's follow these steps to extract the secret:

- Step 1: The files of each cloud are compared to those available in the two lists $L^{(0)}$ and $L^{(1)}$. When the names are identical, these files are retrieved and sorted in ascending order of list numbers. The files extracted by cloud and by list is as follows:

Cloud	$L^{(0)}$	$L^{(1)}$
c_0	chapter.docx	scheduling.xlsx
c_1	thesis.docx	statistics.xlsx
c_2	balanceSheet.docx	
c_3	chapter.docx	

- Step 2: The files in each list are then replaced by their number. The sequence of each cloud obtained is:

Cloud	$L^{(0)}$	$L^{(1)}$
c_0	7	0
c_1	0	1
c_2	2	
c_3	7	

- Step 3: Each sequence belonging to a cloud is stored in column inside the matrix *Mat*. Empty entries in

the matrix are replaced by zeros. The resulting matrix looks like this:

$$Mat = \begin{pmatrix} 7 & 0 & 2 & 7 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

- Step 4: Compute m, the secret in decimal using the base value (B = 9). The conversion is done as follows:

$$m = \sum_{i=0}^{1} \sum_{j=0}^{3} Mat[i, j] \times 9^{(i \times 4)+j}$$

$$= 7*9^0 + 0*9^1 + 2*9^2 + 7*9^3 + 0*9^4 + 1*9^5 + 0*9^6 + 0*9^7$$

$$= 7 + 162 + 5\,103 + 59\,049$$

$$= 64,321$$

- Step 5: The secret s is obtained by converting m to base 2:
$$s = (64321)_{10} = (1111101101000001)_2$$

- Step 6: All the files retrieved in step 1 of extraction are removed from the cloud storage.

Case 4: s = 1,111,101,101,000,001, n = 4 and B = 17

In this last case, the secret s and the number of clouds n remain unchanged. The base considered is B = 17. Let's follow these steps to embed the secret:

- Step 1: The secret is converted to base 17:

$$(1111101101000001)_2 = (64321)_{10} = (D19A)_{17};$$

- Step 2: The secret is subdivided into groups of 4 values, because of the four clouds available. This gives one block: D19A.
- Step 3: For each block, each value is linked to a distinct cloud in the order c_0, c_1, c_2 and c_3:

Bloc #0			
D	1	9	A
↓	↓	↓	↓
c_3	c_2	c_1	c_0

- Step 4: Values of the same cloud are grouped together. Letters in the block are also replaced by their equivalent: D = 13 and A = 10. Therefore, the secret parts of each cloud are:

c_0	c_1	c_2	c_3
10	9	1	13

- Step 5: As the subdivision gave one block, just one list is used. Hide the vector value obtained in step 4 with the list $L^{(0)}$. The stego files to be uploaded in the clouds are allocated as follows:

List	Cloud c_0	Cloud c_1	Cloud c_2	Cloud c_3
$L^{(0)}$	redaction.docx	tutorial.docx	article.docx	exercise.docx

- Step 6: The last embedding step is to transfer the files *redaction.docx* to the cloud c_0; *tutorial.docx* to the cloud c_1; *article.docx* to the cloud c_2 and *exercise.docx* to the cloud c_3.

Let's follow these steps to extract the secret:

- Step 1: The files of each cloud are compared to those available in the list $L^{(0)}$. When the names are identical, these files are retrieved and sorted in the order in which the lists were created. The files extracted by cloud and by list is as follows:

Cloud	$L^{(0)}$
c_0	redaction.docx
c_1	tutorial.docx
c_2	article.docx
c_3	exercise.docx

- Step 2: The files in each list are then replaced by their number. The sequence of each cloud obtained is:

Cloud	$L^{(0)}$
c_0	10
c_1	9
c_2	1
c_3	13

- Step 3: The sequence stored in column in the matrix *Mat*. The resulting vector looks like this:

$$Mat = (10 \ 9 \ 1 \ 13)$$

- Step 4: Compute m, the secret in decimal using the base value (B = 17). The conversion is done as follows:

$$m = \sum_{i=0}^{0} \sum_{j=0}^{3} Mat[i, j] \times 17^{(i \times 4)+j}$$

$$= 10*17^0 + 9*17^1 + 1*17^2 + 13*17^3$$

$$= 10 + 153 + 289 + 63\,869$$

$$= 64,321$$

- Step 5: The secret s is obtained by converting m to base 2:

$$s = (64321)_{10} = (1111101101000001)_2$$

- Step 6: All the files retrieved in step 1 of extraction are removed from the cloud storage.

Discussion

In this paper, we propose a steganographic scheme for secret distribution resistant to detection. Compared to related work, secret extraction assumes that files do not undergo any modification when distributing the secret in multi-cloud storage environment, by hiding the existence of the covert channel between the communicating parties. As examples 1 to 4 show, the data distributed in the clouds decrease as the value of the chosen base increases. We have respectively distributed 16, 8, 6 and 4 values with base $B = 2$, 4, 9 and 17 in examples 1 to 4, considering a fixed secret size and clouds number. We also observe that the base value choice has an impact on the number and the size of the file lists necessary for the secret dissimulation. Each list must contain at least B files and the number of lists must be identical to the blocks number obtained after splitting the secret message represented in base B. In example 1, four lists are used, then in examples 2 and 3 only two lists are used and finally a single list in the last example. Moreover, the files available in the lists are not fully used. The base value indicates the number of files to manage both for secret insertion and extraction. Indeed, only 2, 4, 9 and 17 files are respectively taken into account in each of the lists of examples 1 to 4. An additional argument that argues in favour of the proposed scheme security is the ability to use any file extension to establish the covert channel such as images (.png, .jpg, ...), binary files (.exe, .bin, ...) text files (.txt, .doc, ...), and so on. In this case,

the *word, excel, powerpoint* and *pdf* files were used while maintaining their integrity. Another important note should be raised when using the multi-cloud storage environment. These can be multiple accounts available from a single or multiple storage providers. In the examples of this work we used accounts from four different providers name as: SugarSync, Dropbox, OneDrive and Google Drive.

Ultimately, a comparison of these examples reveals that example 4 has a better distribution of the secret in the four chosen clouds. Only one file is deposited in each cloud, which is not the case in the other examples. This also facilitates the secret extraction by reducing the search time for files available in the lists. In general, the embedding program takes as input the base value, the number of clouds and the directories containing the lists of files, then outputs the files to be uploaded to each cloud in directories. While the extraction program takes as input a set of files from the clouds, the base value, the clouds number and the lists, then outputs the secret. For a better distribution of the secret, an optimal choice must be made on the following two parameters: the clouds number and the base value.

Security analysis

In the literature, some techniques [4, 17–19] use a special character insertion to hide information like: space, ASCII code character *A0*, characters coloring, text justification. In our method, no special character is used. Thus, by just observing the file content, there will be no suspicious items. Therefore, the file content doesn't attract the adversary attention.

Security analysis is done by considering two attacks hypothesis:

Hypothesis 1:an attack by an adversary who doesn't have the ability to access the cloud accounts. This adversary has the following limits:
- He doesn't know the key (the cloud user name and password, the files lists, the base);
- He doesn't know the clouds storage content;
- He doesn't know that a secret communication is taking place by observing only the files transfer between the clouds. Nothing can reveal the secret communication existence because there is no addition of special information in the exchange files.

Hence, it does not attract the adversary's attention. In short, he can't do anything.

Hypothesis 2: an attack by an adversary who can partially or fully access the accounts of the different clouds. This adversary has the following limits:

• He doesn't know the key (the files lists and the base). If the adversary gets the file lists by accessing the cloud accounts, he must find the correct order of the secret distribution in the clouds as well as the numbering of the lists and files contained in these lists. Therefore, he must perform $B! * k! * n!$ permutations in a case of exhaustive search for a successful attack. Unfortunately, this number of permutations is exponential.

• He can't make a link between Alice and Bob. The only connection between Alice and Bod is during the key exchange. After that, there is no direct communication between them. In the proposed steganographic scheme, there is only communication between each party and the cloud. As presented in the Fig. 6, the secret channel doesn't make a direct connection between Alice and Bob. Thus, the usual security model as mentioned in the Fig. 1 is broken.

Just like Hypothesis 1, he can't do anything.

Conclusion

In this paper a new steganography paradigm, transparent to any attacker and resistant to the detection and the secret extraction was proposed. Two properties contribute to achieve these goals: the files do not undergo any modification while the distribution of the secret in the multi-cloud storage environment allows us to hide the existence of the covert channel between the communicating parties. Related works hide usually information inside the covert media. In this work, the covert media is a pointer to information. Therefore the file carries the information without being modified and the only way to access it is to have the key. The experiments carried out have shown that the secret distribution in the clouds decreases as the value of the chosen base increases. Moreover, the base value choice has an impact on the number and the size of the file lists necessary for the secret dissimulation. An additional argument that argues in favour of the proposed scheme security is the ability to use any file extension to establish the covert channel while maintaining their integrity. Another important note should be raised when using the multi-cloud storage environment. These can be multiple accounts available from a single or multiple storage providers.

The paper shows interesting comparison results with remarkable security contributions. The work can be seen as a new open direction for further distributed stego research. Future work will consist in improving the scheme by proposing optimal parameters allowing a better distribution of the secret. We will also study the robustness of the proposed technique in the face of very large secret data. We are also motivated to design new steganographic schemes resistant to detection by preserving the shared files integrity.
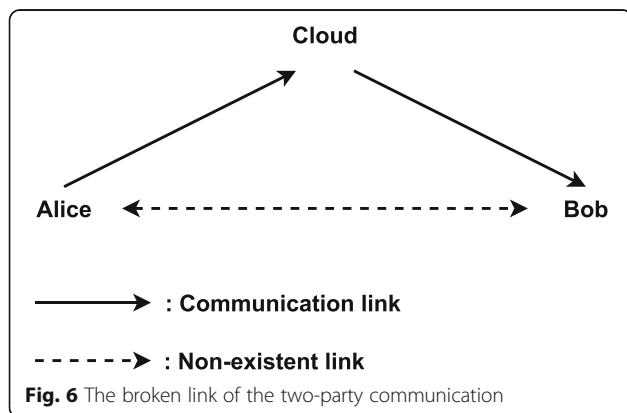
Author details
[1]Team GRIMCAPE, Yaounde, Cameroon. [2]Sorbonne University, IRD, UMMISCO, F-93143 Bondy, France. [3]Department of Computer Science, University of Yaounde I, P.o. Box 812, Yaounde, Cameroon.

Fig. 6 The broken link of the two-party communication

References
1. Mazurczyk W, Wendzel S, Zander S, Houmansadr A, Szczypiorski K (2016) Information hiding in communication networks: fundamental, Mechanisms, Applications, and Countermeasures. Wiley
2. Chang C-C, Kieu TD (2010) A reversible data hiding scheme using complementary embedding strategy. Inf Sci 180(16):3045–3058
3. Por LY, Delina B (2008) Information hiding: a new approach in text steganography. In: WSEAS international conference. In: Proceedings. Mathematics and computers in science and engineering, vol 7. World Scientific and Engineering Academy and Society
4. Ekodeck SGR, Ndoundam R (2016) Pdf steganography based on chinese remainder theorem. J Inform Security Appl 29:1–15

5.   Sahu AK, Swain G (2020) Reversible image steganography using dual-layer lsb matching. Sensing Imaging 21(1):1
6.   Jiang S, Ye D, Huang J, Shang Y, Zheng Z (2020) Smartsteganogaphy: Lightweight generative audio steganography model for smart embedding application. J Netw Comput Appl 102689
7.   Pilania U, Gupta P (2020) Analysis and implementation of IWT-SVD scheme for video steganography. In: Micro-Electronics and Telecommunication Engineering. Springer, Singapour, pp 153–162
8.   Singh N, Bhardwaj J, Raghav G (2017) Network steganography and its techniques: A survey. Int J Comput Appl 174:2
9.   Moskowitz IS, Chang L, Newman RE (2002) Capacity is the wrong paradigm. In: Proceedings of the 2002 Workshop on New Security Paradigms, pp 114–126
10. Sajedi H, Jamzad M (2010) Bss: Boosted steganography scheme with cover image preprocessing. Expert Syst Appl 37(12):7703–7710
11. Kopiczko P, Mazurczyk W, Szczypiorski K (2013) Stegtorrent: a steganographic method for the p2p file sharing service. In: 2013 IEEE security and privacy workshops, p 151 157 IEEE
12. El-Latif AAA, Abd-El-Atty B, Elseuofi S, Khalifa HS, Alghamdi AS, Polat K, Amin M (2020) Secret images transfer in cloud system based on investigating quantum walks in steganography approaches. Physica A: Stat Mech Appl 541:123687
13. Ali M, Khan SU, Vasilakos AV (2015) Security in cloud computing: Opportunities and challenges. Inf Sci 305:357–383
14. 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
15. Luo X-Y, Wang D-S, Wang P, Liu F-L (2008) A review on blind detection for image steganography. Signal Process 88(9):2138–2157
16. Wu S, Zhong S, Liu Y (2018) Deep residual learning for image steganalysis. Multimed Tools Appl 77(9):10437–10453
17. Khosravi B, Khosravi B, Khosravi B, Nazarkardeh K (2019) A new method for pdf steganography in justified texts. J Inf Secur Appl 45:61–70
18. Malik A, Sikka G, Verma HK (2017) A high capacity text steganography scheme based on lzw compression and color coding. Engineering Science and Technology. Int J 20(1):72–79
19. Lee I-S, Tsai W-H (2010) A new approach to covert communication via pdf files. Signal Process 90(2):557–565
20. Bhattacharyya S (2011) A survey of steganography and steganalysis technique in image, text, audio and video as cover carrier. J Global Res Comput Sci 2(4)
21. Wang Z, Chen M, Yang Y, Lei M, Dong Z (2020) Joint multi-domain feature learning for image steganalysis based on cnn. EURASIP J Image Video Process 2020(1):1–12
22. Qiao T, Retraint F, Cogranne R, Zitzmann C (2015) Steganalysis of jsteg algorithm using hypothesis testing theory. EURASIP J Inf Secur 2015(1):1–16
23. Simmons GJ (1984) The prisoners' problem and the subliminal channel. In: advances in cryptology. Springer, Boston, pp 51–67
24. Menezes AJ, Katz J, Van Oorschot PC, Vanstone SA (1996) Handbook of Applied Cryptography. CRC press
25. Jackson JT, Gunsch GH, Claypoole RL, Lamont GB (2003) Blind steganography detection using a computational immune system: a work in progress. Int J Digit Evid 4(1):19
26. Farid H, Lyu S (2003) Higher-order wavelet statistics and their application to digital forensics. In: 2003 Conference on computer vision and pattern recognition workshop, vol 8. IEEE, pp 94–94
27. Farid H (2001) Detecting steganographic messages in digital images. Report TR2001–412. Dartmouth College, Hanover
28. Fridrich J, Goljan M (2002) Practical steganalysis of digital images: state of the art. In: security and Watermarking of Multimedia Contents IV, vol 4675. International Society for Optics and Photonics, pp 1–13
29. Ozer H, Avcibas I, Sankur B, Memon ND (2003) Steganalysis of audio based on audio quality metrics. In: Security and Watermarking of Multimedia Contents V, vol 5020, pp 55–66 International Society for Optics and Photonics
30. Goudar R, Patil A (2012) Packet length based steganography detection in transport layer. Int J Sci Res Publ 2:12
31. Elsadig MA, Fadlalla YA (2018) Packet length covert channels crashed. J Comput Sci Comput Math 8(4):55–62
32. Koikara R, Deka DJ, Gogoi M, Das R (2015) A novel distributed image steganography method based on block-dct. In: Advanced Computer

and Communication Engineering Technology. Springer, Cham, pp 423–435
33. Wibisurya A et al (2017) Distributed steganography using five pixel pair differencing and modulus function. Procedia Comput Sci 116:334–341
34. 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
35. 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
36. Ito M, Saito A, Nishizeki T (1989) Secret sharing scheme realizing general access structure. Electron Commun Jpn (Part III: Fundamental Electronic Science) 72(9):56–64
37. Iftene S (2006) Secret sharing schemes with applications in security protocols. Sci Ann Cuza Univ 16:63–96
38. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
39. Blakley GR (1979) Safeguarding cryptographic keys. In: 1979 international workshop on managing requirements knowledge (MARK). IEEE, pp 313–318
40. Gutub A, Al-Juaid N, Khan E (2019) Counting-based secret sharing technique for multimedia applications. Multimed Tools Appl 78(5):5591–5619
41. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M. (1995) Proactive secret sharing or: how to cope with perpetual leakage. In: annual international cryptology conference, pp. 339-352. Springer
42. Al-Ghamdi M, Al-Ghamdi M, Gutub A (2019) Security enhancement of shares generation process for multimedia counting-based secret-sharing technique. Multimed Tools Appl 78(12):16283–16310
43. Gutub A, AlKhodaidi T (2020) Smart expansion of target key for more handlers to access multimedia counting-based secret sharing. Multimed Tools Appl 1:29
44. AlKhodaidi T, Gutub A (2020) Trustworthy target key alteration helping counting-based secret sharing applicability. Arab J Sci Eng 1:21
45. Gutub A, Alaseri K (2019) Hiding shares of counting-based secret sharing via arabic text steganography for personal usage. Arab J Sci Eng 1:26
46. 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
47. Gutub A, Al-Ghamdi M (2019) Image based steganography to facilitate improving counting-based secret sharing. 3D Res (1):6
48. Gutub A, Al-Ghamdi M (2020) Hiding shares by multimedia image steganography for optimized counting-based secret sharing. In: Multimedia Tools and Applications, pp 1–35
49. Li B, He J, Huang J, Shi YQ (2011) A survey on image steganography and steganalysis. J Inf Hiding Multimedia Signal Process 2(2):142–172
50. Karampidis K, Kavallieratou E, Papadourakis G (2018) A review of image steganalysis techniques for digital forensics. J Inf Secur Appl 40:217–235
51. Liu T-Y, Tsai W-H (2007) A new steganographic method for data hiding in microsoft word documents by a change tracking technique. IEEE Trans Inf Forensic Secur 2(1):24–30
52. Su W, Ni J, Hu X, Fridrich J (2020) Image steganography with symmetric embedding using gaussianmarkov random field model. IEEE Trans Circuits Syst Video Technol
53. Ali AH, George LE, Mokhtar MR (2020) An adaptive high capacity model for secure audio communication based on fractal coding and uniform coefficient modulation. Circuits Syst Signal Process 39(10):5198–5225
54. Baziyad M, Rabie T, Kamel I (2020) Directional pixogram: a new approach for video steganography. In: 2020 advances in science and engineering technology international conferences (ASET), Dubai, United Arab Emirates, pp 1–5

Publisher's Note