Distributed Data Hiding in A Single Cloud Storage Environment

Stéphane Willy MOSSEBO
Universite de Yaounde 1 Faculte des Sciences
Leonel MOYOU METCHEKA
Universite de Yaounde 1 Faculte des Sciences
Rene NDOUNDAM (ndoundam@yahoo.com)
Universite de Yaounde 1 Faculte des Sciences https://orcid.org/0000-0003-1105-762X

Research Article

Keywords: information, steganography, cloud environment, security, cloud management

Posted Date: May 4th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-455126/v1

Version of Record: A version of this preprint was published at Journal of Cloud Computing on August 21st, 2021. See the published version at https://doi.org/10.1186/s13677-021-00258-2.

Noname manuscript No.
(will be inserted by the editor)

Distributed data hiding in a single cloud storage environment

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

the date of receipt and acceptance should be inserted later

Abstract Information has always constituted a prized commodity over time, with the strong globalization of exchanges and communications. The provenance, acquisition, protection and integrity of information has always asked major questions, with the strong growth of security needs in many areas around the world. Numerous approaches have been proposed with a view mainly to increase information security. One of them is steganography, where the goal is to conceal the very presence of secret information in a media during a communication. However, this one generally modifies the properties of the cover media used to conceal secret information. Thus, this modification constitutes a vulnerability during the detection and extraction of information concealed in the media, by various steganography attacks. To overcome this problem, Moyou and Ndoundam have proposed a distributed method of data hiding in a multi-cloud storage environment that preserves the property of the cover media, by using different types of media as an index of a part of the secret. However, a security problem is revealed when sharing key between the participating entities in the communication, and the process of managing cloud storage environments between the participating entities can turns to be tedious, depending on the number of cloud accounts used in the method. In order to solve these problems, in this paper we

Mossebo Tcheunteu Stéphane Willy
E-mail: smossebo@gmail.com

Leonel Moyou Metcheka
E-mail: leonelmoyou@gmail.com

René Ndoundam
E-mail: 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
[*]Corresponding author

propose a new approach of the method that uses a single cloud storage environment, and strengthens the security of key sharing between the participating entities.

Keywords information · steganography · cloud environment · security · cloud management

1 Introduction

The increasing development of many areas in the world has led to an increase in the protection of information[1,2], because the detention of it constitutes an enormous strategic and economic stake to increase its power. So it has become essential, to set up appropriate techniques to protect it for the person who holds it[3]. Two main techniques are associated: cryptography and steganography.
Cryptography aims at making information incomprehensible to a person who does not possess the key [4], whereas steganography seeks to conceal the very presence of relevant information within several others, without a real interest for an attacker or spy during a communication[5].
Steganography is the art and science of communicating in a way which hides the existence of the communication[6]. It makes use of an unsecured channel between communicating parties and uses several types of files to conceal the secret information (texts,images,sounds,videos)[7–10]. It can be used exclusively or combined with other information protection techniques such as cryptography or watermarking[11]. Studies carried out on it have even used it in network protocols[12].
Four different properties are used to evaluate the performance of a steganographic scheme : capacity, security, perceptibility and robustness[13].

Capacity: is the amount of information that can be concealed in the cover media.

Security: is the inability of an unauthorized user to detect concealed information.

Perceptibility: corresponds to the indistinguishable character of the cover-medium and the stego-medium.

Robustness: is the ability to preserve hidden data in view of changes in the stego-medium.

However, security is the most sought property to achieve undetectable communication[14]. In order to strengthen the protection of the secret, an approach based on distributed steganography[15] is to divide the secret by distributing it between several participants in the communication and each of them conceals the information of their secret entries in separate files. The interest of this approach is to make the detection of the secret particularly difficult for an attacker.
An approach that comes close to this one is the secret sharing[16], which refers to any method for distributing a secret among a group of participants, each of which allocates a share of the secret. Secret sharing is applied for systems that require reinforced security needs with access control by multiple users[17].
However in the approaches presented above, the simple fact of the suspicion of the modification of a medium between the distributed participants, by an attacker can

lead to the deletion of the medium. Thus the different attacks in steganography, in general are based on the changing properties of the cover media when a communication is revealed[18].
In order to solve this problem, Moyou and Ndoundam[19] have proposed a distributed method data hiding in multi-cloud storage environment by considering several types of media as an index of a part of the secret. This method considers beforehand an exchange of files between the participating entities during a communication or meeting, which is susceptible to suspicion. And the number of clouds accounts to manage can be tedious for the participants. Our approach consists to solve these problems by proposing a new method of it, which uses a single cloud storage environment. With the basic hypothesis that the list of files is no longer exchanged between the sender and the receiver when sharing key. This list is accessible by both parties in the cloud storage environment.

The rest of the paper is organized as follows: in section 2, we present the related work on the models of distributed steganography. Section 3 is consecrated to our contribution on a new distribution scheme in a cloud storage environment. In section 4, experimental results are done of our contribution. And finally section 5 is devoted to the conclusion and perspectives of our future work.

2 Presentation of distributed's models in steganography

2.1 Distributed steganography

Distributed steganography is defined as a secret communication distributed between several independent senders and a single receiver[15]. Each sender only knows his own private data and applies a steganographic technique to conceal his private data. The receiver receives the union of their secret inputs, without revealing their information.
Some known examples of distributed steganography are: spies' problem, business investment, government management.

- *Spies' problem:* Military spies are assigned to investigate military allies, and each one detects the force deployment of one country. Now the general deployment of the alliance is only required by the leader of spies, while it is unnecessary to know each country's information. Every spy should protect his own investigation results, and does not know any information about the general deployment.

- *Business investment:* Many competing commercial organizations might jointly invest in a project. The executive branch of the project are only interested in the sum of their profits. Their own earnings should be protected.

- *Government management:* Each subordinate branch of the government has private and secret information about a special affair. The superior officer wants to

know the general status. Their private information should be kept secret for the rest.

More generally, a distributed steganography model is characterized by:

- P_i $(i = 1,2,\ldots,r)$ a set of r participants (senders)
- R the receiver
- m_i $(i = 1,2,\ldots,r)$ the set of input messages corresponding to each participant
- c the cover text and ĉ the stego-text

Each P_i only knows its own entry m_i. All P_i sends the union of their private inputs in a public channel to R, concealing the presence of a communication to others and to an eavesdropper.

The diagram below describes the process of a distributed steganography scheme

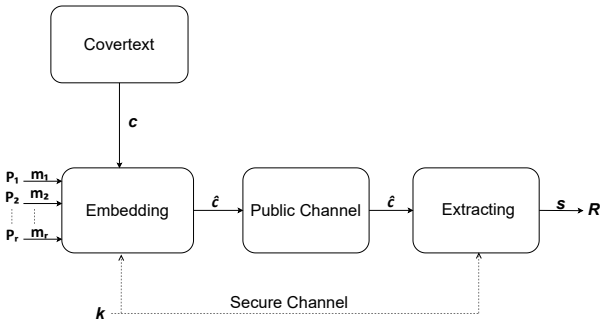

Fig. 1: schema of a distributed steganography process

2.2 Secret sharing

Secret sharing is a method that has known a major advanced over time. In terms of security, sensitive information for applications or resources that must be protected by more than one person[16]. Many applications can use it, in particular when opening a bank safe, controlling missile launches, voting systems, sensitive encryption[20]. The secret sharing scheme divides a secret among a set of participants, in such a way that this set of participants can reconstruct the secret. This scheme breaks down into two phases: the construction-distribution phase of the secret and the reconstruction

phase of the secret.

- In the construction-distribution phase of the secret, an algorithm (trusted dealer) allows to share the secret between the different participants and to distribute it through a secure channel.[21]
- In the reconstruction phase, this one consists of access to the structure able to qualify a subset of participants to collaborate and reconstruct the secret by sharing their secrets in a specific way.[22]

Two sets of participants are listed, an authorized set and an unauthorized set has recovered the secret, in this sharing process the key is considered very secure. The system distributes the secret among n participants and recovers the secret by combining their k shares, where k represents the number of keys that will be shared such that $n \geq k$.
This concept of secret sharing results from the independent works of Shamir[23] and Blakley[24]. Shamir's scheme is based on polynomial interpolation while Blakley's scheme is based on geometry.

One of the approaches which is inspired by this principle is the counting based secret sharing[17], the approach consists in counting the one in parallel within the share key to generate the target key. This approach 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[25,26], as well as to enhance the number of generated shares [27]. 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[28,29] or images [30,31].

2.3 Study and limits of the distributed model of Moyou and Ndoundam

The different steganography techniques presented above all have one point in common in their methods, which is the modification of a candidate file (media) to conceal information. This modification presents a fault in view of the different steganographic attacks so they are subjected.
So, in order to get around this problem, Moyou and Ndoundam[19] have proposed an approach where the cover medium carries the information without being modified, the medium is considered as an index to secret information.

In this approach, the following elements are considered :

1. The cover objects represent any file extensions distributed in different cloud storage environments$(c_0, c_1, \ldots, c_{n-1})$, such that $n \geq 2$.

2. The secret message represents any message formats encoded in a specific base.

3. The key constitutes the shared elements between the sender and the receiver during the embedding and extraction of the secret message which are:

   – Cloud environments $c_0, c_1, \ldots, c_{n-1}$.

   – Authentication for access to each cloud account (user's name and password) W_i, 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 at least 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 $B \geq 2$.

Sender and receiver share a list of identical files and cloud access information. The secret is encoded in a specific base and subdivided to a specific blocks number corresponding to the cloud numbers. A list of files is sent to each cloud.
The receiver accesses each cloud account with the access information and reconstructs the secret based on the positions of the files in the cloud.
Concretely, the secret message is encoded in a specific base. At each value of the encoded secret message is associated a file in the list, the associated file carries the information of the value of the encoded secret message. Then the files are deposited in the different cloud storage environments. Each file thus constitutes a pointer to the encoded secret message. Finally, both entities possessing the same list of files. It will suffice to retrieve the positions of the different files in the different cloud accounts, and to retrieve the encoded secret message and so, the initial secret message.

2.3.1 limits of the distributed model of Moyou and Ndoundam

The method of concealing distributed information in a multi-cloud storage environment presented above is a good approach, which solves the problem of changing the properties of the media used to conceal the secret. The use of cloud storage environments ensures the authentication and confidentiality of the files used to conceal the secret, and breaks the communication link between the participating entities. So, the detection and extraction of the secret is more difficult in the eyes of the different attackers. However, the management of cloud accounts can turns to be tedious, if it uses a large number of clouds and the exchange of key (particularly files for cloud environments) between the participating entities can be linked to a suspicion of an attacker. So we propose, to simplify the management of cloud accounts of the scheme presented and to reduce the information of the key shared between the participating entities.

3 Our contribution

The management of clouds of the concealment scheme proposed by Moyou and Ndoundam[19] which can turns to be tedious to manage, for a large number of clouds. And the key sharing between the participating entities containing sensitive files to conceal the secret. We propose a new method of the process embedding and extraction of the secret in 3 approches using a single cloud environment, which presents a set of folders with several files already present in each folder. Files and access information to multiple cloud accounts, are no longer subject to an exchange between the participating entities, when sharing of the key. In our scheme, mainly the access information to a single cloud account and a given base is shared between the sender and the receiver. So the elimination of files that conceals the secret between the sender and the receiver, allows to increase the security of the scheme. And reducing the access information to cloud accounts to a single cloud account, makes the scheme easier to manage.

3.1 Overview

Given a secret message s in base 2, we encode it in a given base B $(\alpha_{n-1}\alpha_{n-2}\dots\alpha_0)_B$, such that $0 \leq \alpha_i < B$. The cloud storage environment consists of a number of folders(at least equal to the size of the secret) plus an additional folder considered as the cover folder. Each folder except the cover folder contains a set of at least B files, so the role is to store the different files to conceal the secret message s. Each file is found once only in all the folders and is unique in its membership folder. The cover folder is initially empty at the beginning of the process and its role is to store the files, which conceal the secret message.

The embedding and extraction algorithms are carried out through the following elements:

1. The cover object represents any file extension located in the cloud environment folders.
2. The cover folder represents a set of files which conceals the secret message.
3. The concealed information represents the secret message encoded in a specific base.
4. The key represents the information shared between the sender and the receiver during an encrypted communication or a meeting.

3.2 Notations and hypothesis

Notations for the embedding and extraction algorithms of the secret are as follows:

- s is the secret message encoded in base 2
- B is the base used to encode the secret, such that $B \geq 2$
- $(\alpha_{n-1}\alpha_{n-2}\ldots\alpha_0)_B$ is the representation in base B of the secret
- $F^{(i)}$ is the i^{th} folder which contains different types of files, such that $0 \leq i < n$
- $F^{(n)}$ is the cover folder which contains the concealed information
- $F_j^{(i)}$: is the j^{th} file in folder number i, $0 \leq i < n$ and $0 \leq j < B$

Hypothesis

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

Each folder i contains at least B files in the cloud storage environment.
The sender and the receiver agree beforehand on an order of classification of the files in the cloud storage environment.
This order can be either:
a classification of files in alphabetical order, either by the date of creation of these files or any other order held as part of the key.

3.3 First approach

In the embedding phase, the sender connects in the cloud storage environment by using the key that he shares with the receiver, it encodes the secret message in a specific base B and browses each position i of the encoded secret message which contains information α_i($0 \leq i < n$ and $0 \leq \alpha_i < B$) and copy the file from folder i of position α_i and paste in the cover folder, this process is performed for each position i of the secret message.

The extraction proceeds as follows, the receiver connects in the cloud storage environment by using the key that he shares with the sender. Open the cover folder and compares each file present with the other files present in the folders i, $0 \leq i < n$. When a match is found, the receiver progressively reconstitutes the values α_i of the positions i of the encoded secret message, then it performs the calculation from base B to base 2 to recover the secret, once the secret message is recovered the files present in the cover folder are deleted for a reinitialisation of the process.

The diagram below describes an overview of the first approach

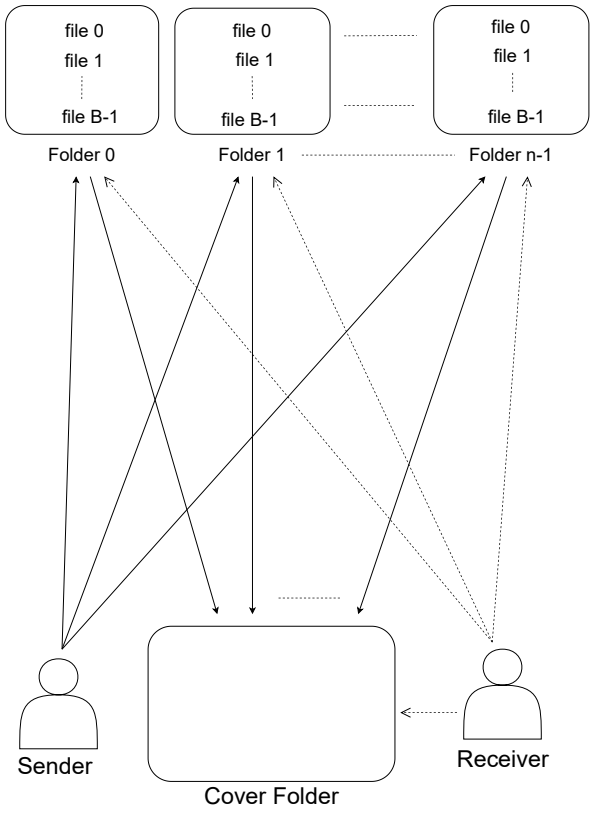

Fig. 2: Overview of a first approach

3.3.1 Embedding

The process of embedding the secret message is carried out as follows:

Input:

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

1. Check that the cover folder $F^{(n)}$ is empty and delete all the files in it, if it is not empty.
2. Convert the secret message s to base B such that $s = (\alpha_{n-1}\alpha_{n-2}\ldots\alpha_0)_B$, where $0 \leq \alpha_i < B$.
3. For each position i of the secret message: $i = 0, 1, \ldots, n-1$
   3.1. Find the file with index α_i in the folder $F^{(i)}$
   3.2. Select and copy the file $F_{\alpha_i}^{(i)}$ and paste in the cover folder $F^{(n)}$

End

3.3.2 Extraction

The process of extracting the secret is carried out as follows:

Input:
$\tilde{C}$: Cloud account modified
W: access for authentication in the cloud account
$F^{(0)}, F^{(1)}, \ldots, F^{(n-1)}$: the folders in the cloud account
$F^{(i)}$: is the folder number i in the cloud account
$F^{(n)}$: is the cover 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
B: the base used
$tab[0, \ldots, n-1]$: integer array that retrieves each index α_i of folder number i.
Output:
s: the secret message
Begin

1. $i = 0$ // first file of the cover folder
2. while($i < n$) // i browse each file in the cover folder
   2.1. For each folder j in the cloud account : $j = 0, 1, \ldots, n-1$
      2.2. For each file k in the current folder $F^{(j)}$: $k = 0, 1, \ldots, B-1$
         2.2.1. Compare the file $F_i^{(n)}$ in the cover folder with the file $F_k^{(j)}$
         2.2.2. if($F_i^{(n)} = F_k^{(j)}$) then
            2.2.2.1. $tab[j] = k$
            2.2.2.2. $i = i + 1$ // next file of the cover folder
            2.2.2.3. go to instruction 2.
3. Calculate $m = \sum_{j=0}^{n-1}(tab[j] \times B^j)$
4. Convert m to binary and get the secret message s

5. Delete all files in the cover folder $F^{(n)}$.

End

3.4 Second approach

The information contained in the key are: cloud account access information, the base used to encode the secret and a list of files from each folder contained in the cloud storage environment.

In the embedding phase, the sender connects in the cloud storage environment by using the key that he shares with the receiver, it encodes the secret message in a specific base B and browses each position i of the encoded secret message which contains information α_i($0 \leq i < n$ and $0 \leq \alpha_i < B$) and cut the file from folder i of position α_i and paste in the cover folder, this process is performed for each position i of the secret message.

The extraction proceeds as follows, the receiver connects in the cloud storage environment by using the key that he shares with the sender, it opens the cover folder and compares each file present with the list of files of each folder contained in the key. When a match is found, the receiver progressively reconstitutes the values α_i of the positions i of the encoded secret message, then it performs the calculation from base B to base 2 to recover the secret message, once the secret message is recovered the files present in the cover folder are deleted for a reinitialisation of the process.

The diagram below describes an overview of the second approach

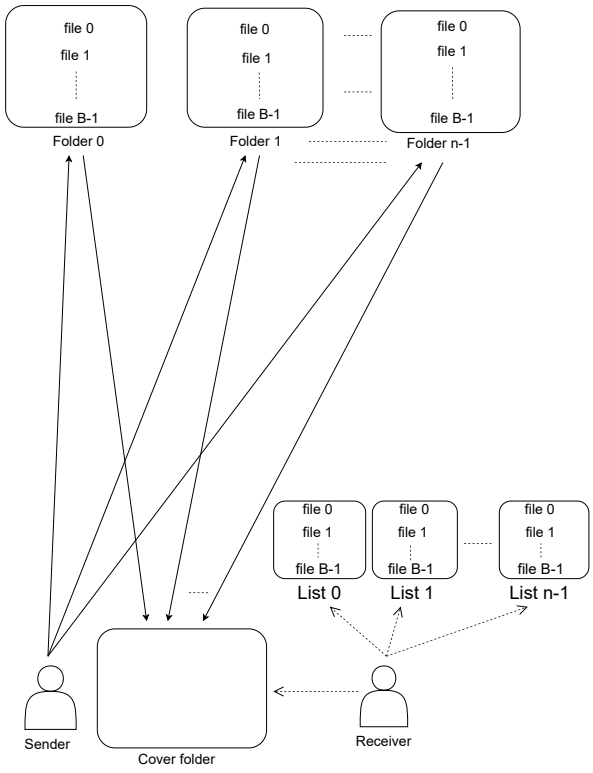

Fig. 3: Overview of a second approach

3.4.1 Embedding

The process of embedding the secret message is carried out as follows:
Input:
C: the cloud account
S: the secret message
B: the base used
$F^{(0)}, F^{(1)}, \ldots, F^{(n-1)}$: the folders in the cloud account
$F^{(i)}$: is the folder number i in the cloud account
$F^{(n)}$: is the cover 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
$L^{(i)}$: is the list number i held by the receiver
$L_j^{(i)}$: $0 \leq i < n$ and $0 \leq j < B$, file j from list number i contained in a list held by the receiver
W: access for authentication in the cloud account
Output:

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

Begin

1. Check that the cover folder $F^{(n)}$ is empty and delete all the files in it, if it is not empty.

2. Convert the secret message s to base B such that $s = (\alpha_{n-1}\alpha_{n-2}\ldots\alpha_0)_B$, where $0 \leq \alpha_i < B$.

3. For each position i of the secret message: $i = 0, 1, \ldots, n-1$
   3.1. Find the file with index α_i in the folder $F^{(i)}$
   3.2. Select and cut the file $F_{\alpha_i}^{(i)}$ and paste in the cover folder $F^{(n)}$

End

3.4.2 Extraction

The process of extracting the secret is carried out as follows:

Input:
$\tilde{C}$: Cloud account modified
W: access for authentication in the cloud account
$F^{(0)}, F^{(1)}, \ldots, F^{(n-1)}$: the folders in the cloud account
$F^{(i)}$: is the folder number i in the cloud account
$F^{(n)}$: is the cover 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
$L^{(i)}$: is the list number i held by the receiver
$L_j^{(i)}$: $0 \leq i < n$ and $0 \leq j < B$, file j from list number i contained in a list held by the receiver
B: the base used
$tab[0, \ldots, n-1]$: integer array that retrieves each index α_i from the folder i.
Output:
S: the secret message
Begin

1. $i = 0$ // first file of the cover folder

2. while($i < n$) // i browse each file in the cover folder
    2.1. For each list $j : j = 0, 1, \ldots, n-1$
        2.2. For each file k in the list $L^{(j)}$: $k = 0, 1, \ldots, B-1$
            2.2.1. Compare the file $F_i^{(n)}$ in the cover folder with the file $L_k^{(j)}$
            2.2.2. if($F_i^{(n)} = L_k^{(j)}$) then
                2.2.2.1. $tab[j] = k$
                2.2.2.2. $i = i + 1$ // next file of the cover folder
                2.2.2.3. go to instruction 2.
3. Calculate $m = \sum_{j=0}^{n-1}(tab[j] \times B^j)$
4. Convert m to binary and get the secret message s
5. Delete all files in the cover folder $F^{(n)}$.

End

3.5 Third approach

The receiver uses an intermediate cloud account, which contains the same information of the cloud account of the participating entities. The information contained in the key are: the access information to the cloud account of the participating entities and the base used to encode the secret.

In the embedding phase, the sender connects in the cloud storage environment by using the key that he shares with the receiver, it encodes the secret message in a specific base B and browses each position i of the encoded secret message which contains information α_i($0 \leq i < n$ and $0 \leq \alpha_i < B$) and cut the file from folder i of position α_i and paste in the cover folder, this process is performed for each position i of the secret message.

The extraction proceeds as follows, the receiver connects to the cloud environment by using the information contained in the key. Then it opens the cover folder and compares each file present with the files of each folder contained in the intermediate cloud environment. When a match is found, the receiver progressively reconstitutes the values α_i of the positions i of the encoded secret message, then it performs the calculation from base B to base 2 to recover the secret message, once the secret message is recovered the files present in the cover folder are deleted for a reinitialisation of the process.

The diagram below describes an overview of the third approach

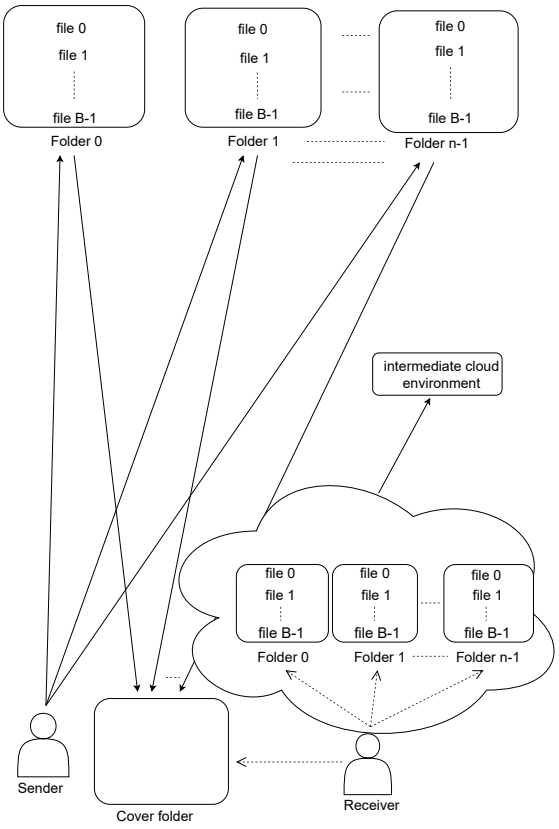

Fig. 4: Overview of a third approach

3.5.1 Embedding

The process of embedding the secret message is carried out as follows:

Input:

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

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

Begin

1. Check that the cover folder $F^{(n)}$ is empty and delete all the files in it, if it is not empty.

2. Convert the secret message s to base B such that $s = (\alpha_{n-1}\alpha_{n-2}\ldots\alpha_0)_B$, where $0 \leq \alpha_i < B$.

3. For each position i of the secret message: $i = 0, 1, \ldots, n-1$
   3.1. Find the file with index α_i in the folder $F^{(i)}$
   3.2. Select and cut the file $F_{\alpha_i}^{(i)}$ and paste in the cover folder $F^{(n)}$

End

3.5.2 Extraction

The process of extracting the secret is carried out as follows:

Input:
$\tilde{C}$: Cloud account modified
W: access for authentication in the cloud account
$\tilde{W}$: access for authentication in the intermediate cloud account
$F^{(0)}, F^{(1)}, \ldots, F^{(n-1)}$: the folders in the cloud account
$\tilde{F}^{(0)}, \tilde{F}^{(1)}, \ldots, \tilde{F}^{(n-1)}$: the folders in the intermediate cloud account
$F^{(n)}$: is the cover folder in the cloud account
$F^{(i)}$: is the folder number i in the cloud account
$\tilde{F}^{(i)}$: is the folder number i in the cloud intermediate account
$F_j^{(i)}$: $0 \leq i < n$ and $0 \leq j < B$, file j from folder number i in cloud account
$\tilde{F}_j^{(i)}$: $0 \leq i < n$ and $0 \leq j < B$, file j from folder number i in the intermediate cloud account
B: the base used
$tab[0, \ldots, n-1]$: integer array that retrieves each index α_i from the folder i.
Output:
S: the secret message
Begin

1. $i = 0$ // first file of the cover folder

2. while($i < n$) // i browse each file in the cover folder
   2.1. For each folder j in the intermediate cloud account : $j = 0, 1, \ldots, n-1$
      2.2. For each file k in the current folder $\tilde{F}^{(j)}$: $k = 0, 1, \ldots, B-1$
         2.2.1. Compare the file $F_i^{(n)}$ in the cover folder with the file $\tilde{F}_k^{(j)}$
         2.2.2. if($F_i^{(n)} = \tilde{F}_k^{(j)}$) then
            2.2.2.1. $tab[j] = k$
            2.2.2.2. $i = i + 1$ // next file of the cover folder
            2.2.2.3. go to instruction 2.
3. Calculate $m = \sum_{j=0}^{n-1}(tab[j] \times B^j)$
4. Convert m to binary and get the secret message s
5. Delete all files in the cover folder $F^{(n)}$.

End

3.6 Time Complexity

In this sub-section, we calculate the time complexity of the 3 proposed approaches. We have a secret s distributed between n folders, each folder contains at least B files. For the embedding of the secret message in the worst case, the file that corresponds to the index α_i of the secret s, in folder i is the file at position number B in the folder. So, this corresponds to browse B files in the folder i. In total, we have n folders and B files browsed by folder, which gives $O(n * B)$ for this case.

For the extraction of the secret message in the worst case, the file that corresponds to index i of the cover folder corresponds to the last file of the last folder $n-1$ which gives $O(n * B)$, in total we have n files in the cover folder and each file corresponds to $O(n * B)$, which gives $O(n^2 * B)$ for this case.

4 Experimentation

In this section we present an evaluation of the hidden bits capacity of our proposed scheme and the execution through 3 examples. Then we present a discussion with the works of Moyou and Ndoundam[19] and finally a security analysis of our proposed scheme.

4.1 Evaluation of hidden bits capacity

The idea is to make an estimate of the number of bits hidden in the cloud storage environment. Each folder in the cloud storage environment 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 account, we have B^n possibilities, so the number of hidden bits is:

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

4.2 Examples

We describe on 3 examples the process of embedding and extraction of the secret of each proposed approach above. The cloud environment used is Google Drive with 3 different configurations for each approach of our proposed scheme. The username is *user* with for password *userpwd*.

4.2.1 Example 1

$s = 1010100, B = 8$

Cloud storage environment configuration: The files are classified in ascending alphabetical order in each folder. The secret message is $s = 1010100$ and the base used is $B = 8$. The cloud storage environment used has 4 folders, with 3 folders that contain 8 files, and an additional folder considered as the cover folder.

The secret message $s = (124)_8$ corresponds to 3 folders $F^{(2)}$,$F^{(1)}$,$F^{(0)}$ for the respective positions $2,1,0$ of the secret message. An additional folder $F^{(3)}$ is considered as the cover folder, which gives a total of 4 folders. 8 files are used for the folders $F^{(2)}$,$F^{(1)}$,$F^{(0)}$, because the base used is 8.

Table below presents the folders $F^{(2)}$,$F^{(1)}$,$F^{(0)}$ in the cloud storage environment

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

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

1. Check that the cover folder $F^{(3)}$ is empty and delete all the files in it, if it is not empty.

   table below presents the cover folder $F^{(3)}$ after step 1

The cover folder	Folder $F^{(3)}$
	empty

2. Encoded the secret s in base 8, $s = (124)_8$.
3. For each position i of the secret s: $i = 0, 1, 2$
   3.1. Find the file with the index α_i in the folder $F^{(i)}$.

The following table describes the process

α_2	α_1	α_0
1	2	4
↓	↓	↓
article.txt	french.pdf	learning.docx

4. Select and copy each file found $F_{\alpha_i}^{(i)}$ and paste in the cover folder $F^{(3)}$ as shown in table below.

the cover folder	folder $F^{(3)}$
$file_0$	article.txt
$file_1$	french.pdf
$file_2$	learning.docx

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

1. $i = 0$ // first file of the cover folder $F^{(3)}$(article.txt)
2. while($i < n$) // i browse each file in the cover folder $F^{(3)}$(article.txt, french.pdf, learning.docx)
   2.1. For each folder j in the cloud account : $j = 0, 1, 2$
   2.2. For each file k in the current folder $F^{(j)}$: $k = 0, 1, \ldots, 7$
   2.2.1. Compare the file $F_i^{(3)}$ in the cover folder with the file $F_k^{(j)}$
   2.2.2. if($F_i^{(3)} = F_k^{(j)}$) then
   2.2.2.1 $tab[j] = k$
   2.2.2.2. $i = i + 1$ // next file of the cover folder $F^{(3)}$
   2.2.2.3. go to instruction 2.

The table below is the correspondence found, allowing to update the table tab

	learning.docx #2	french.pdf #1	article.txt #0
j	0	1	2
	↓	↓	↓
$tab[j]$	4	2	1

3. Calculate $m = \sum_{j=0}^{2}(tab[j] \times 8^j) = tab[0] \times 8^0 + tab[1] \times 8^1 + tab[2] \times 8^2 = 4 \times 8^0 + 2 \times 8^1 + 1 \times 8^2 = 84$.
4. Convert m to binary and get the secret message s
   $m = 84 = (1010100)_2$, the secret message $s = (1010100)_2$ is retrieved.
5. Delete all files in the cover folder.

4.2.2 Example 2

$s = 10101100, B = 4$

Cloud storage environment configuration: The files are classified by creation date ascending in each folder. The secret message is $s = 10101100$ and the base used is $B = 4$. The cloud storage environment used has 5 folders, with 4 folders that contain 4 files and an additional folder considered as the cover folder.

The secret message $s = (2230)_4$ corresponds to 4 folders $F^{(3)}$,$F^{(2)}$,$F^{(1)}$,$F^{(0)}$ for the respective positions $3,2,1,0$ of the secret. An additional folder $F^{(4)}$ is considered as the cover folder, which gives a total of 5 folders. 4 files are used for the folders $F^{(3)}$,$F^{(2)}$,$F^{(1)}$,$F^{(0)}$, because the base used is 4.

Table below presents the folders $F^{(3)}$,$F^{(2)}$,$F^{(1)}$,$F^{(0)}$ in the cloud storage environment

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

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

1. Check that the cover folder $F^{(4)}$ is empty and delete all the files in it, if it is not empty.

   table below presents the cover folder $F^{(4)}$ after step 1

The cover folder	Folder $F^{(4)}$
	empty

2. Encoded the secret s in base 4, $s = (2230)_4$.

3. For each position i of the secret s
   3.1. Find the file with the index α_i in the folder $F^{(i)}$.

The following table describes the process

α_3	α_2	α_1	α_0
2	2	3	0
↓	↓	↓	↓
presentation.pptx	english.pptx	statistics.xlsx	steganalysis.pdf

4. Select and cut each file found $F_{\alpha_i}^{(i)}$ and paste in the cover folder $F^{(4)}$ as shown in table below.

the cover folder	folder $F^{(4)}$
$file_0$	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 list of files by folder of the cloud environment. The correspondence between the files in the cover folder is made on this list, because these files had been cut in the cloud storage environment.

The table below shows the 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

1. $i = 0$ // first file of the cover folder(steganalysis.pdf)

2. while($i < n$) // i browse each file in the cover folder(steganalysis.pdf, statistics.xlsx, english.pptx, presentation.pptx)
    2.1. For each list j: $j = 0,1,2,3$
        2.2. For each file k in the list $L^{(j)}$: $k = 0,1,2,3$
            2.2.1. Compare the file $F_i^{(4)}$ in the cover folder with the file $L_k^{(j)}$
            2.2.2. if($F_i^{(4)} = L_k^{(j)}$) then
                2.2.2.1. $tab[j] = k$
                2.2.2.2. $i = i + 1$ // next file of the cover folder
                2.2.2.3. go to instruction 2.

The table below is the correspondence found allowing to update the table tab

	presentation.pptx #3	english.pptx #2	statistics.xlsx #1	steganalysis.pdf #0
j	3	2	1	0
	↓	↓	↓	↓
$tab[j]$	2	2	3	0

3. Calculate $m = \sum_{i=0}^{3}(tab[i] \times 4^i) = tab[0] \times 4^0 + tab[1] \times 4^1 + tab[2] \times 4^2 + tab[3] \times 4^3 = 0 \times 4^0 + 3 \times 4^1 + 2 \times 4^2 + 2 \times 4^3 = 172$.
4. Convert m to binary and get the secret message s
   $m = 172 = (10101100)_2$, the secret message s = $(10101100)_2$ is retrieved.
5. Delete all files from the cover folder.

4.2.3 Example 3

$s = 10011, B = 3$

Cloud storage environment configuration: The files are classified by increasing size in each folder. The One Drive intermediate cloud account is used in this configuration, the secret message is $s = 10011$ and the base used is $B = 3$.

The secret message $s = (201)_3$ corresponds to 3 folders $F^{(2)}$,$F^{(1)}$,$F^{(0)}$ for the respective positions $2,1,0$ of the secret. An additional folder $F^{(3)}$ is considered as the cover folder, which gives a total of 4 folders. 3 files are used for the folders $F^{(2)}$,$F^{(1)}$,$F^{(0)}$, because the base used is 3.

Table below presents the folders $F^{(2)}$,$F^{(1)}$,$F^{(0)}$ in the cloud storage environment

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

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

1. Check that the cover folder $F^{(3)}$ is empty and delete all the files in it, if it is not empty.

   table below presents the cover folder $F^{(3)}$ after step 1

The cover folder	Folder $F^{(3)}$
	empty

2. Encoded the secret s in base 3, $s = (201)_3$.
3. For each position i of the secret s
   3.1. Find the file with the index α_i in the folder $F^{(i)}$.

   The following table describes the process

α_2	α_1	α_0
2	0	1
↓	↓	↓
php.xlsx	java.pdf	css.pdf

4. Select and cut each file found $F_{\alpha_i}^{(i)}$ and paste in the cover folder $F^{(3)}$ as shown in table below.

The cover folder	Folder $F^{(3)}$
$file_0$	java.pdf
$file_1$	css.pdf
$file_2$	php.xlsx

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

The receiver holds the Google Drive cloud account information in a One Drive cloud account. The correspondence between the files in the cover folder is made on this account, because these files had been cut in the Google Drive cloud environment.

The table below shows the information contained in the One Drive cloud account held by the receiver.

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

1. $i = 0$ // first file of the cover folder(java.pdf)
2. while($i < n$) // i browse each file in the cover folder(java.pdf, css.pdf, php.xlsx)
   2.1. For each folder j in the intermediate cloud account : $j = 0, 1, 2$
      2.2. For each file k in the current folder $\tilde{F}^{(j)}$: $k = 0, 1, 2$
         2.2.1. Compare the file $F_i^{(3)}$ in the cover folder with the file $\tilde{F}_k^{(j)}$
         2.2.2. if($F_i^{(3)} = \tilde{F}_k^{(j)}$) then
            2.2.2.1. $tab[j] = k$
            2.2.2.2. $i = i + 1$ // next file of the cover folder
            2.2.2.3. go to instruction 2.

The table below is the correspondence found allowing to update the table tab

	php.xlsx #2	css.pdf #1	java.pdf #0
j	2	0	1
	↓	↓	↓
$tab[j]$	2	1	0

3. Calculate $m = \sum_{i=0}^{2}(tab[i] \times 3^i) = tab[0] \times 3^0 + tab[1] \times 3^1 + tab[2] \times 3^2 = 1 \times 3^0 + 0 \times 3^1 + 2 \times 3^2 = 19$.
4. Convert m to binary and get the secret message s
   $m = 19 = (10011)_2$, the secret message s = $(10011)_2$ is retrieved.
5. Delete all files from the cover folder.

4.3 Discussion

Our approach presented above, is a technique of hiding information in a cloud computing storage environment that solves the problems of cloud management and key management in Moyou and Ndoundam's method[19], by proceeding as follows:

- The management of cloud accounts is reduced to a single cloud account, whatever the size of the secret.
- Files that conceal information are no longer exchanged between the participating entities.

The table below provides an assessment of the criteria of Moyou and Ndoundam's method[19] and of our contribution.

Concealment technique	Moyou and Ndoundam method[19]	Our Contribution
Difficulty of process	Tedious (for a very large number of clouds)	Easy (because it uses only one cloud)
Security	High	Higher(because no file exchange is carried out)
Imperceptibility	High	High

In this table, we note the following elements:

- The difficulty of the process in Moyou and Ndoundam's method[19], is linked to a division of the secret message requiring a large number of clouds. The disadvantage of using a large number of clouds is linked to the cost (not all cloud accounts are free) and also to the difficulty of managing these accounts. Our contribution, solves the problem by using a single cloud account (easier to manage) whatever the size of the secret.
- The security of the process in Moyou and Ndoundam's method[19] is high, because it uses a cloud environment, and hides the secret message on several file extensions. But the sharing of these files between the participating entities can be problematic, if they are linked to a suspicion of an attacker. Our contribution, also solves the problem by removing the sharing of these files in the key.
- The imperceptibility in both cases remains high, because the files which conceal the secret message in both methods carry no information, and are considered as index of the secret, which reveals the non perceptible character of the secret in these files.

4.4 Security analysis

In this section we present the security of our proposed scheme, according to the different attacks so it can be subjected by an attacker. Two hypothesis are considered depending on the information held by the attacker:

Hypothesis 1:

The attacker does not hold any information of the key shared between the sender and the receiver. In this case the attack is not possible by the attacker, because he does not have access to the cloud account and to the files contained in the cloud environment.

Hypothesis 2:

The attacker holds some or all of the information contained in the key. In this case the attacker has access to the cloud account and can perform an exhaustive search between the files in the cover folder and the folders containing at least B files to recover the secret. The cover folder contains n files and each folder contains at least B files, so the search for the attacker will be performed in $n \times B^n$ for the n folders in the cloud environment.

However the choice of the base to encode the secret also influences the search of the secret for the attacker. Because, we can notice that for a given secret message the more the base increases, the fewer folders used and the more the base decreases, the more folders used in the cloud environment. Thus the attacker's task will be more or less exhautive during the search for the secret.
On the other hand, the process of embedding and extraction of the secret will be exhaustive to be carried out by the participants, for a small base and therefore more secure and for a large base, the process will be less exhaustive and therefore less secure for any attacker.
So, depending on the security needs of the participants and information available in the cloud environment, the base will be chosen small or large.

5 Conclusion

In this article, we have presented a new method of information hiding in a cloud computing storage environment based on Moyou and Ndoundam's work. This information hiding scheme uses a single cloud environment, whatever the size of the secret with multiple folders containing a set of files. Thus the management of the cloud account is simpler compared to Moyou and Ndoundam's method. The files that conceal the secret information are no longer subject to exchange by the participating entities, hence improving security.
Analysis of this scheme shows that, the choice of base influences the number of files that must be present in the cloud environment, which can make the task of detection and extraction of the secret difficult for attackers.
In future work, we will seek to apply our work in distributed communications in one or more cloud environments and also in the secret sharing techniques.

Competing interests

The authors declare that they have no competing interests.

Author's 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.

Acknowledgements

This work was supported by UMMISCO and the University of Yaounde 1.

Additional Files

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.

References

1. Jessica Litman. Information privacy/information property. *Stanford Law Review*, pages 1283–1313, 2000.
2. Naresh K Malhotra, Sung S Kim, and James Agarwal. Internet users' information privacy concerns (iuipc): The construct, the scale, and a causal model. *Information systems research*, 15(4):336–355, 2004.
3. Richa Gupta, Sunny Gupta, and Anuradha Singhal. Importance and techniques of information hiding: A review. *arXiv preprint arXiv:1404.3063*, 2014.
4. Mohammed AbuTaha, Mousa Farajallah, Radwan Tahboub, and Mohammad Odeh. Survey paper: cryptography is the science of information security. 2011.
5. David Kahn. The history of steganography. In *International Workshop on Information Hiding*, pages 1–5. Springer, 1996.
6. Sabu M Thampi. Assistant professor, department of computer science & engineering, lbs college of engineering, kasaragod, kerala-671542, s. *India—Information Hiding Techniques: A Tutorial Review‖, ISTE-STTP on Network Security & Cryptography, LBSCE*, 2004.
7. Shivani Sharma, Avadhesh Gupta, Munesh Chandra Trivedi, and Virendra Kumar Yadav. Analysis of different text steganography techniques: a survey. In *2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)*, pages 130–133. IEEE, 2016.
8. Tayana Morkel, Jan HP Eloff, and Martin S Olivier. An overview of image steganography. In *ISSA*, volume 1, 2005.
9. Palwinder Singh. A comparative study of audio steganography techniques. *International Research Journal of Engineering and Technology (IRJET)*, 3(4), 2016.
10. Ramadhan J Mstafa, Khaled M Elleithy, and Eman Abdelfattah. Video steganography techniques: taxonomy, challenges, and future directions. In *2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pages 1–6. IEEE, 2017.
11. R Gayathri and V Nagarajan. Secure data hiding using steganographic technique with visual cryptography and watermarking scheme. In *2015 International Conference on Communications and Signal Processing (ICCSP)*, pages 0118–0123. IEEE, 2015.
12. Józef Lubacz, Wojciech Mazurczyk, and Krzysztof Szczypiorski. Principles and overview of network steganography. *IEEE Communications Magazine*, 52(5):225–229, 2014.
13. SN Wawale and Prof A Dasgupta. Review of data hiding techniques. *International Journal for Advance Research in Engineering and Technology*, 2(2):32–37, 2014.
14. Ira S Moskowitz, LiWu Chang, and Richard E Newman. Capacity is the wrong paradigm. In *Proceedings of the 2002 workshop on New security paradigms*, pages 114–126, 2002.
15. Xin Liao, Qiao-yan Wen, and Sha Shi. Distributed steganography. In *2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 153–156. IEEE, 2011.
16. Amos Beimel. Secret-sharing schemes: a survey. In *International Conference on Coding and Cryptology*, pages 11–46. Springer, 2011.
17. Adnan Gutub, Nouf Al-Juaid, and Esam Khan. Counting-based secret sharing technique for multimedia applications. *Multimedia Tools and Applications*, 78(5):5591–5619, 2019.

18. Andreas Westfeld and Andreas Pfitzmann. Attacks on steganographic systems. In *International workshop on information hiding*, pages 61–76. Springer, 1999.
19. Leonel Moyou Metcheka and René Ndoundam. Distributed data hiding in multi-cloud storage environment. *Journal of Cloud Computing*, 9(1):1–15, 2020.
20. Sorin Iftene. Secret sharing schemes with applications in security protocols. *Sci. Ann. Cuza Univ.*, 16:63–96, 2006.
21. Douglas R Stinson. Decomposition constructions for secret-sharing schemes. *IEEE Transactions on Information Theory*, 40(1):118–125, 1994.
22. Huaxiong Wang and Duncan S Wong. On secret reconstruction in secret sharing schemes. *IEEE Transactions on Information Theory*, 54(1):473–480, 2008.
23. Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
24. George Robert Blakley. Safeguarding cryptographic keys. In *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pages 313–318. IEEE, 1979.
25. Maimoona Al-Ghamdi, Manal Al-Ghamdi, and Adnan Gutub. Security enhancement of shares generation process for multimedia counting-based secret-sharing technique. *Multimedia Tools and Applications*, 78(12):16283–16310, 2019.
26. Adnan Gutub and Taghreed AlKhodaidi. Smart expansion of target key for more handlers to access multimedia counting-based secret sharing. *Multimedia Tools and Applications*, pages 1–29, 2020.
27. Taghreed AlKhodaidi and Adnan Gutub. Trustworthy target key alteration helping counting-based secret sharing applicability. *Arabian Journal for Science and Engineering*, pages 1–21, 2020.
28. Adnan Gutub and Khaled Alaseri. Hiding shares of counting-based secret sharing via arabic text steganography for personal usage. *Arabian Journal for Science and Engineering*, pages 1–26, 2019.
29. Adnan Abdul-Aziz Gutub and Khaled Aydh Alaseri. Refining arabic text stego-techniques for shares memorization of counting-based secret sharing. *Journal of King Saud University-Computer and Information Sciences*, 2019.
30. Adnan Gutub and Maimoona Al-Ghamdi. Hiding shares by multimedia image steganography for optimized counting-based secret sharing. *Multimedia Tools and Applications*, pages 1–35, 2020.
31. Adnan Gutub and Maimoona Al-Ghamdi. Image based steganography to facilitate improving counting-based secret sharing. *3D Research*, 10(1):6, 2019.

Figures

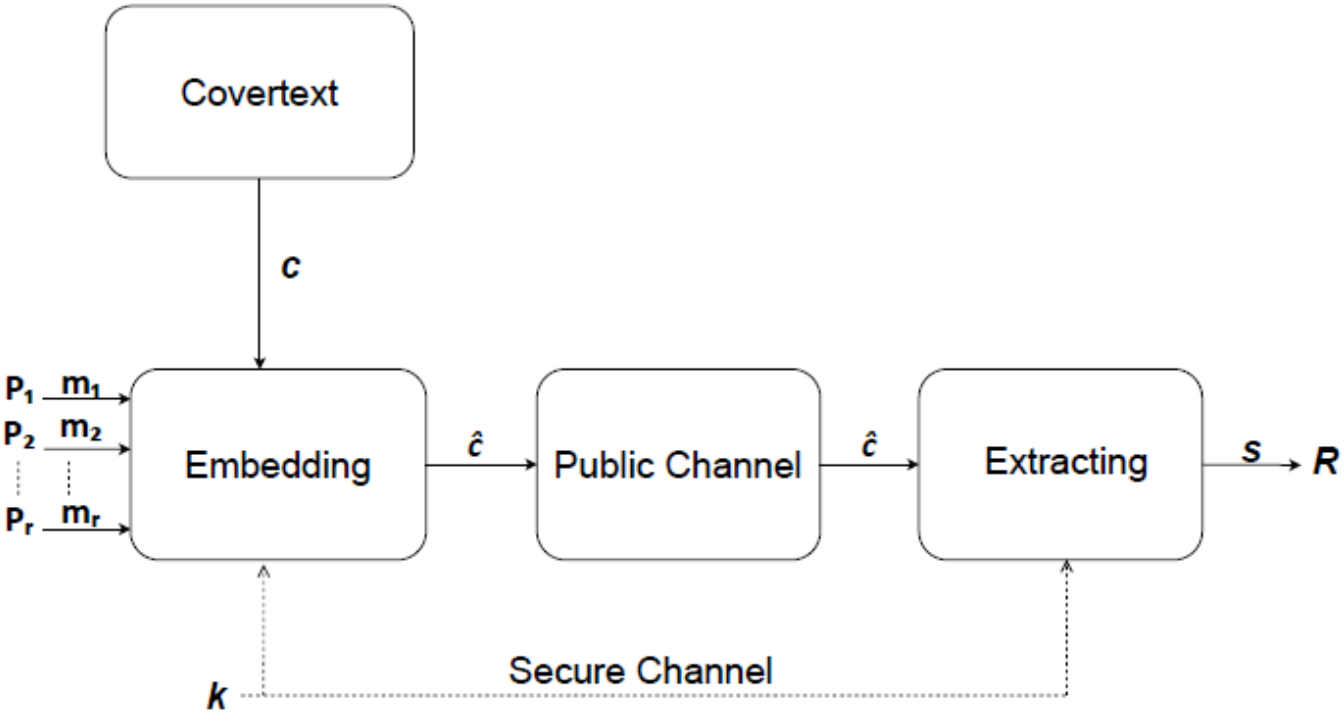

Figure 1

Schema of a distributed steganography process

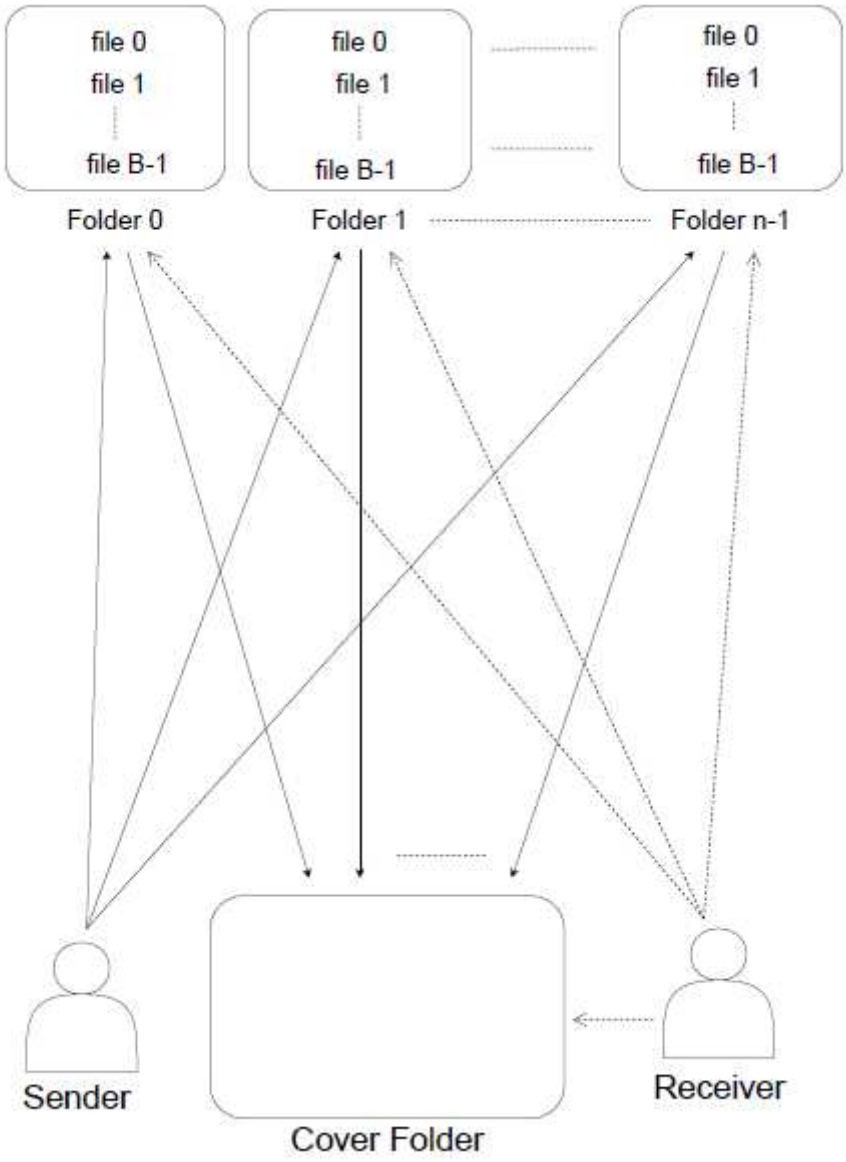

Figure 2

Overview of a first approach

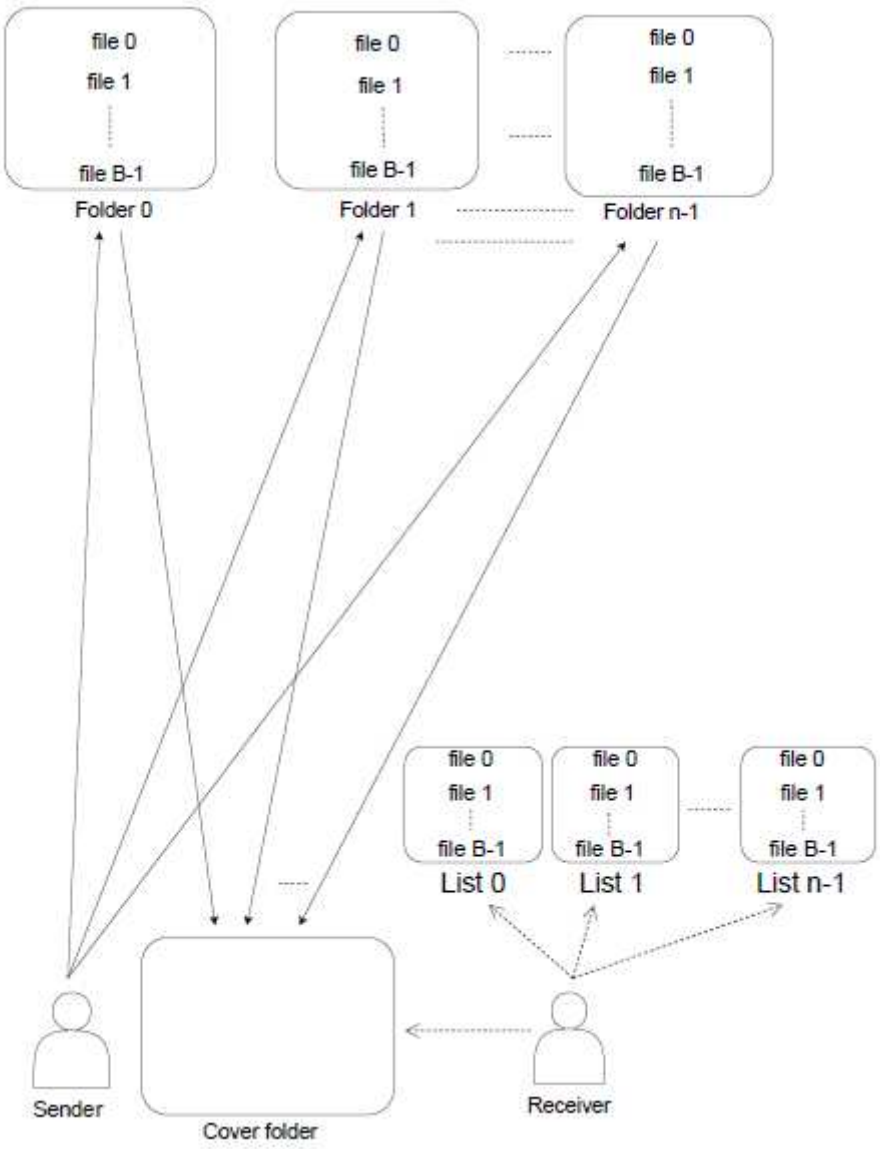

Figure 3

Overview of a second approach

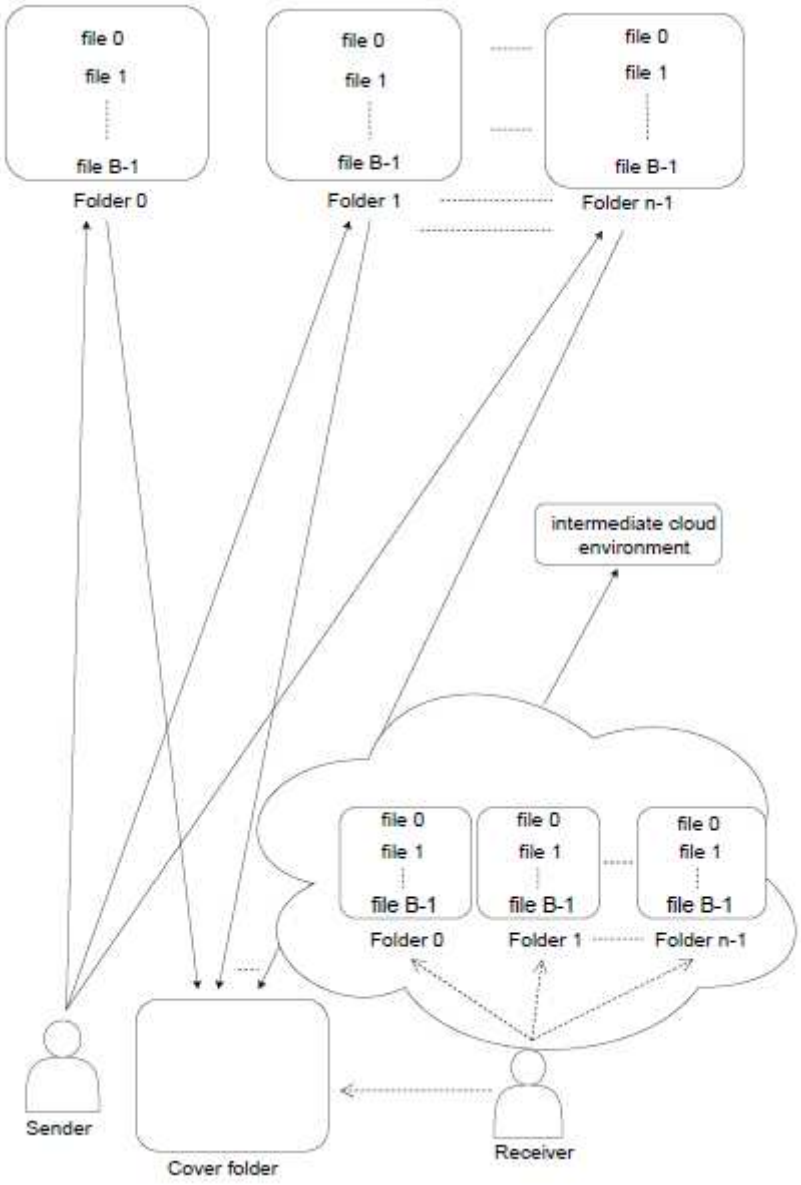

Figure 4

Overview of a third approach