Ingénierie des Systèmes d'Information
Vol. 27, No. 6, December, 2022, pp. 915-921
Journal homepage: http://iieta.org/journals/isi

Brute Force Attack on Distributed data Hiding in the Multi-Cloud Storage Environment More Diminutive than the Exponential Computations

Arif Mohammad Abdul*, Arshad Ahmad Khan Mohammad, Matti Kiran Sastry, Jyothi Bankapalli

Department of CSE, GITAM Deemed to be University, Hyderabad 502329, India

Corresponding Author Email: arif.software@gmail.com

https://doi.org/10.18280/isi.270607

Received: 10 August 2022
Accepted: 6 October 2022

Keywords:
security, steganography, brute force attack, multi-cloud storage environment, cover media, encoding

ABSTRACT

Classical steganography is designed to hide data by cover media. Recent approaches fragmented the data and hide them in a distributed manner by embedding each fragmented data into a distinct cover media. This approach makes a secret message extremely difficult for an attacker to detect. However, cover media modification leaves fingerprints that could expose a secret channel to an attacker. To overcome the problem, a novel steganography concept designed by two technical contributions. I). cover media does not undergo any modification, i.e., the cover media act as a pointer to fragmented data. II). A secret message is stored in the multi-cloud storage environment. The approach claimed that it is computationally infeasible for an attacker to detect and extract the hidden message despite of having fully access to the accounts of the different clouds. In this paper, we analysed the security strength of the novel steganography concept and concluded that, attacker can get the secret value stored in multi-cloud storage environment using the brute force attacks more diminutive than exponential computations.

1. INTRODUCTION

The Internet has become very important in today's era. The Internet revolution has resulted in new innovative technologies impacting users' usage from traditional to classical methods. More utilization of the Internet leads to cyber-attacks. Increased demands for online connectivity due to comfortable usage of the latest technologies drastically increase Internet consumption, from individual users to enterprises (transportation, education, tourism, and food). Data is continuously exchanged through internet like text, audio, video, and images in terms of digital NEWS and E-books, watching movies, browsing the net, search engines, and social media (WhatsApp, Twitter, Facebook, emails, etc.). Thus, cyber-attack is a big concern and threat to data security [1].

Growing Internet growth leads to the Cloud [2]. Cloud computing is the ubiquitous model with a distributed computing paradigm. It consists of a large pool of virtualized and dynamically adjustable resources that can be easily useable and accessible. The services of resources are provided to the users' on-demand and pay-per-use basis with fewer management efforts. All utility computing restrictions can efficiently resolve by adopting the cloud. Moreover, cloud computing features are high computing power, providing expensive resources on-demand with less pay, and fewer management efforts. Cloud improves users' computational time, cost-effectiveness, access to data, and higher utilization of services. In addition, data is stored in a distributed manner. Many users fail to secure data while transferring their sensitive information despite utilizing security tools because of unawareness of cyberattacks. The attackers, hackers, adversaries, and intruders are rigorously waiting for an attack. Data security depends on three objectives Confidentiality, Integrity, and Availability, i.e., the CIA. These are the foundation objectives of security. Cryptography and steganography approaches are used to handle an attack on users' sensitive data.

The paper works [3] on steganography and does not require much more mathematical computation like Cryptography. The steganography method hides the secret code in the cover message as plaintext, pixels, audio signal, and video files [4-7] of the least significant bits. This method is characterized by imperceptibility and is responsible for the quality of the image. Robustness uses to extract secret code even though the image is cropping, rotating, or blurring, undetectable/security provides difficulty to steganalysis attack to retrieve the secret code from the cover message, embedding payload characteristic improves the hiding capacity of secret code. In a survey, the research carries on classical and distributed steganography.

The classical approach mainly used the linguistic property to hide the secret code. This scheme achieves unsuspicious transportations using multiple algorithms. One algorithm is used for embedding a secret message in the cover message to produce a stego-message, and another is used for retrieving a secret message from the stego-message, but this approach fails due to a lack of resilience [8].

Distributed steganography approach improves the limitations of the classical scheme by segmenting a secret code and hiding it in different media. Multi-cloud is a multiple public or private cloud service where data is stored and processed. Advantages: A single file fragmented and stored in multiple clouds, difficult to identify a single file, difficult to integrate all the blocks without key, It generates multiple stego-message of one secret code to make it difficult for an attacker and disadvantages: Monitoring multiple clouds for a single file, take high execution time [9, 10]. In a recent survey, a public algorithm was proposed [11], where the receiver is one and communication channels are many to transfer the data. The authentication mechanism's security strength depends

entirely on its key management [12]. The data is shared through multiple media with the receiver, and only the receiver can able to integrate all secret messages. Distributed steganography makes detection tasks more difficult for attackers by breaking the secret message into multiple channels and storing them in random places. The fact of protecting secret messages from hackers is named as secret sharing. Many users fail to secure data while transferring their sensitive information despite utilizing security tools because of unawareness of cyberattacks [13-15].

Classical steganography is designed to hide data by cover media. Recent approaches fragmented the data and hide them in a distributed manner by embedding each fragmented data into a distinct cover media. This approach makes a secret message extremely difficult for an attacker to detect. However, cover media modification leaves fingerprints that could expose a secret channel to an attacker. The author provided an extension of the existing approach [3], i.e., the novel steganography concept, and claimed that it is complicated for an attacker to detect and extract the secret message. It is designed based on two technical contributions. First, cover media does not undergo any modification, i.e., the cover media act as a pointer to fragmented data. Second, a secret message is stored in the multi-cloud storage environment. It claimed that it is complicated for an attacker to detect and extract the secret message.

The aim of the paper is to analyse the designed novel steganography concept to check the security strength against the stored secret value in multi-cloud environment. Thus, the paper performs the brute force attack by assuming that the only unknown thing to attacker is key, as according to Kirchhoff's principles of cryptography. The main aim of the brute force attack to check the whether the stored secrete message in multi-cloud environment can be retrieved by an attacker more diminutive than the exponential computations.

2. DISTRIBUTED DATA HIDING IN THE MULTI-CLOUD STORAGE ENVIRONMENT

Metcheka and Ndoundam [3] use the steganography approach to secure the password. The existing [3] covert channel is the novel steganography concept and claimed that it is complicated for an attacker to detect and extract the secret message. The current system was designed based on two technical contributions. First, cover media does not undergo any modification, i.e., the cover media act as a pointer to fragmented data. Second, a secret message is stored in the multi-cloud storage environment. The cover media selection and uploading into the Cloud are based on the message that communicating entities want to hold secretly. Here, the approach uses files as cover media and is uploaded into the Cloud without any modification. Uploading the files depends on the secret message and key. The Key contains the following information: number of clouds and their order, i.e., $C_0, C_1, C_3 \ldots . C_{n-1}$, and their login credentials, File lists, i.e., $L^0, L^1, L^3 \ldots . , L^{k-1}$, where each list consists of the set of files, say ., $L^0 = L^0{}_1, L^0{}_2, L^0{}_3 \ldots . L^0{}_{B-1}$ and encoding base values (B). Work assumed that the communication entities securely share the key information.

The sender, who wants to store the secret in the Cloud secretly, initially transcodes the secret in a specific base and converts it into K blocks, i.e., b_0, b_1, b_2, b_3 ...., b_{K-1} and each block consists of n values. Each value of the block is substituted with one of the files from the file lists and sent to the Cloud for storing; the same process is followed for other blocks. Each block is assigned a list of files. Further, they are stored in the Cloud. The receiver, which has access to the Cloud, browses them and recovers the files, and additionally constructs the secret by substituting files with values.

2.1 The working procedure of the algorithm is as follows

1. s: The input secret, and transcodes the secret in a specific base B: $B \geq 2$;
2. The secret represents in base B, and is split into blocks of n values.
3. For each secret block
    a. Log in to the first cloud storage
    b. For each value of the block
        i. Substitute value with the specific file
        ii. Store the file in the Cloud
        iii. Open the subsequent cloud storage

Example

The sender and receiver share the key (K), as follows.

$k = number\ of\ clouds\ (n = 4)$,
and their login credentials, the four lists L^0, L^1, L^2 *and* L^3 *and base value, & B=2, or 4 or ,9 or, or 16*

Let's assume that the sender wants to store the 16-bit secret value *s = 1, 111,101,101,000,001* into four clouds, i.e., *n=4*, so that the receiver can securely retrieve the secret using the shared key. The secret stored inside the Cloud depends on the number of clouds, base value for encoding, and list of files, i.e., cover media. The working procedure of secret distribution over distributed clouds securely is explained with examples as follows.

Case 1: Secret *s = 1, 111,101,101,000,001* distribution securely inside distributed environment with encoding base value *B=2*, number of clouds *n=4*, Lists which is shown in Table 1.

1. Transcodes the secret in base ***B=2*, i.e.,** *s =(1111101101000001_2)*
2. Split into blocks 0001 0100 1011 1111, each block consists of 4 values, as *n=4*.
3. Each value of the block is substituted with the specific file refer the paper [3] for file list and substitution concept.
4. The embedding step is to transfer the files to respective clouds.

article.docx, scheduling.xlsx, results.pptx, and cryptography.pdf to the cloud c0.
thesis.docx, scheduling.xlsx, results.pptx, and cryptography.pdf to the cloud c1.
thesis.docx, statistics.xlsx, conference.pptx, and cryptography.pdf to the cloud c2.
thesis.docx, scheduling.xlsx, results.pptx, and cryptography.pdf to the cloud c3.

Case 2: Secret *s = 1, 111,101,101,000,001* distribution securely inside distributed environment with encoding base value *B=4*, number of clouds *n=4*, Lists which is shown in Table 1.

1. Transcodes the secret in base ***B=9*** i.e., *s = (33231001_4)*
2. Split into blocks 1001 3323, each block consists of 4 values, as *n=4*.
3. Each value of the block is substituted with the specific file refer the paper [3] file list and substitution.
4. The embedding step is to transfer the files to their respective clouds.

article.docx and data.xlsx to the cloud c0.
thesis.docx and budget.xlsx to the cloud c1.
thesis.docx and data.xlsx to the cloud c2.
article.docx and data.xlsx to the cloud c3.

Case 3: Secret *s = 1, 111,101,101,000,001* distribution securely inside distributed environment with encoding base value *B=9*, number of clouds *n=4*, Lists which is shown in Table 1.

1. Transcodes the secret in base ***B=9,*** i.e., *s* =(107207_9)
2. Split into blocks 7207 10, each block consists of 4 values, as *n=4*.
3. Each value of the block is substituted with the specific file refer the paper [3] for file list and substitution concept.
4. The embedding step is to transfer the files to their respective clouds.

chapter.docx and scheduling.xlsx to the cloud c0
thesis.docx and statistics.xlsx to the cloud c1.
balanceSheet.docx to the cloud c2.
chapter.docx to the cloud c3.

The work distributed 16-bit secret value *s =1, 111,101,101,000,001* with base *B=2, 4, 9 & 17* into clouds by 16, 8, 6 and 4 values. It is observed that increment in base value requires a smaller number of files necessary for the secret concealment.

Table 1. The four file lists and their index number [3]

	L^0		L^1		L^2		L^3
0	art.docs	0	report.xlsx	0	document.pptx	0	publications.pdf
1	thesis.docs	1	sheet1.xlsx	1	file.pptx	1	linear.pdf
2	article.docs	2	binding.xlsx	2	seminar.pptx	2	bipolar.pdf
3	report.docs	3	gradebook.xlsx	3	conference.pptx	3	document.pdf
4	balancesheet.docs	4	setup.xlsx	4	slides.pptx	4	simulation.pdf
5	chapter1.docs	5	project.xlsx	5	speech.pptx	5	results.pdf
6	index.docs	6	analytics.xlsx	6	shopping.pptx	6	chapter.pdf
7	introduction.docs	7	curves.xlsx	7	bigdata.pptx	7	conference.pdf
8	chapter2.docs	8	quotation.xlsx	8	accounts.pptx	8	passport.pdf
9	journal.docs	9	finance.xlsx	9	security.pptx	9	card.pdf
10	editor.docs	10	classes.xlsx	10	marketing.pptx	10	introduction.pdf
11	lesson.docs	11	sheet2.xlsx	11	resume.pptx	11	contacts.pdf
12	book.docs	12	phonebook.xlsx	12	animation.pptx	12	awards.pdf
13	news.docs	13	salaries.xlsx	13	slides2.pptx	13	code.pdf
14	letter.docs	14	employees.xlsx	14	movie.pptx	14	human.pdf
15	meeting.docs	15	bill.xlsx	15	symposium.pptx	15	learning.pdf

2.2 Security analysis

1. The author claims that the proposed approach does not leave any suspicious items to the attacker as files do not modify before uploading into the Cloud.
2. If an attack cannot access the Cloud, could not compute the key, and further cannot launch the attack
3. Assume that an attacker can access the clouds. He gets the file lists. Authors claim that attacker cannot be able to set up the key by viewing file lists. As he must find the cloud order in which the secret is distributed, he must convert the files to index values. Authors claim that the attacker needs to perform the *B!*K!*n!* computations to find the secret message and this computation is exponential, where,

B=base encoding value,
K=number of files in list,
n=number of clouds

3. BRUTE FORCE ATTACK ON DISTRIBUTED DATA HIDING IN THE MULTI-CLOUD STORAGE ENVIRONMENT

Paper claimed that an attack by an adversary who can fully access the accounts of the different clouds does not have access to the key, so the attacker needs $B! * K! * n!$ Computations to get the secret, and this computation is exponential. We analysed the performance of the work [3] and concluded as follows.

Lemma: If an attacker gets access to the clouds can get the secret less than the *B!*K!*n!* Computations, where $B =$ *base encoding value* , $K =$ number of lists used; & *n=number of clouds*.

Proof: According to Kirchhoff's principles of cryptography, the only unknown thing to the attacker is the key, apart from the algorithm, and the key size is public. We assume that the existing algorithm is known to the attacker, and key contents are also known to the attacker i.e., The existing algorithm key contents are "file lists, the base value, the set of clouds and their login credentials." But attacker does not know the values of the key content. The only unknown thing to the attacker here is the file lists and the base value. Figure 1 shows the flowchart of the brute-force attack, the explanation of the brute force attack is explained with the following examples.

Case 1 base value *B=2, n=4,* and secret *s =1, 111,101,101,000,001*

We assume that the attacker got access to the Clouds and gets the data stored in the Cloud. He got the files stored by the existing approach as follows.

From Cloud C0	article.docx, scheduling.xlsx, results.pptx and cryptography.pdf;
From Cloud C1	thesis.docx, scheduling.xlsx, results.pptx and cryptography.pdf
From Cloud C2	thesis.docx, statistics.xlsx, conference.pptx and cryptography.pdf
From Cloud C3	thesis.docx, scheduling.xlsx, results.pptx and cryptography.pdf

Then attacker can easily convert the above-accessed data into lists as follows, by taking all the 0th ordered files are stored in List 0 and all the 1st ordered files are stored in List 1, and all the 2nd ordered files are stored in List 2, and all the 3rd ordered files are stored in List 2. Which is given as follows.

List 0	article.docx, thesis.docx, thesis.docx, thesis.docx
List 1	scheduling.xlsx, scheduling.xlsx, statistics.xlsx, scheduling.xlsx
List 2	results.pptx, results.pptx, conference.pptx, results.pptx
List 3	cryptography.pdf, cryptography.pdf, cryptography.pdf, cryptography.pdf

The attacker aims to convert the given secret code to plain text. To do this attacker needs to compute the correct base values and suitable replacement of files to corresponding values. An attacker assumes that the possible base values are one of the 2,4,9,17. The attacker apply all the possible combinations (brute force) to convert the given secret code to plain text. The cryptanalysis approach is explained as follows:

a. The attacker starts with base value 2. The possible values are 0 and 1. Then attacker convert files to either 0 or 1.

List 0 replacement

In list 0 as thesis.docx appears in three places, it must represent with same value, i.e., either 0 or 1.

The possible values of List 0 are {(1,0,0,0) (0,1,1,1)}.

List 1 replacement

In list 1 as scheduling.xlsx appears three places, it must be represented with the same value, i.e., either o or 1.

The possible values in List 1 are {(1,1,0,1) (0,0,1,0)}.

List 2 replacement

In list 2, as results.pptx appears three places, it must be represented with same value i.e., either o or 1.

The possible values in List 2 are {(1,1,0,1) (0,0,1,0)}.

List 3 replacement

In list 3, as cryptography.pdf appears in four places, it must be either 1 or 0.

The possible values in List 3 are {(1,1,1,1) (0,0,0,0)}

The final values are {(1,0,0,0) (0,1,1,1)}, {(1,1,0,1) (0,0,1,0)}, {(1,1,0,1) (0,0,1,0)}, {(1,1,1,1) (0,0,0,0)}.

Table 2. List value to secret code generation brute force

I.No	L0	L1	L2	L3	Final code
1	1,0,0,0	1,1,0,1	1,1,0,1	1,1,1,1	1000110111011111
2	1,0,0,0	1,1,0,1	1,1,0,1	0,0,0,0	1000110111010000
3	1,0,0,0	1,1,0,1	0,0,1,0	1,1,1,1	1000110100101111
4	1,0,0,0	1,1,0,1	0,0,1,0	0,0,0,0	1000110100100000
5	1,0,0,0	0,0,1,0	1,1,0,1	1,1,1,1	**1000001011011111 Matched**
6	1,0,0,0	0,0,1,0	1,1,0,1	0,0,0,0	1000001011010000
7	1,0,0,0	0,0,1,0	0,0,1,0	1,1,1,1	1000001000101111
8	1,0,0,0	0,0,1,0	0,0,1,0	0,0,0,0	1000001000100000
9	0,1,1,1	1,1,0,1	1,1,0,1	1,1,1,1	0111110111011111
10	0,1,1,1	1,1,0,1	1,1,0,1	0,0,0,0	0111110111010000
11	0,1,1,1	1,1,0,1	0,0,1,0	1,1,1,1	0111110100101111
12	0,1,1,1	1,1,0,1	0,0,1,0	0,0,0,0	0111110100100000
13	0,1,1,1	0,0,1,0	1,1,0,1	1,1,1,1	0111001011011111
14	0,1,1,1	0,0,1,0	1,1,0,1	0,0,0,0	0111001011010000
15	0,1,1,1	0,0,1,0	0,0,1,0	1,1,1,1	0111001000101111
16	0,1,1,1	0,0,1,0	0,0,1,0	0,0,0,0	0111001000100000

Table 2 shows that within 5 computations, attackers can get the secret value. In the worst-case attacker can get a secret value at the 16th computation. This value is very much less than $B! * K! * n!$.

Case 2 base value *B=4*, n=4, and secret *s =1, 111,101,101,000,001*

We assume that the attacker got access to the Clouds and gets the data stored in the Cloud. He got the files stored by the existing approach as follows

From cloud C0	article.docx and data.xlsx
From cloud C1	thesis.docx and budget.xlsx
From cloud C2	thesis.docx and data.xlsx
From cloud C3	article.docx and data.xlsx

Then he converted it into a list as follows: all the 0th ordered files are stored in List0, and all the 1st ordered files are stored in List 1. There is no List 2 and List 3, as no files are presented in order 2 and 3. The list arrangement is given as follows.

List 0	article.docx, thesis.docx, thesis.docx, article.docx
List 1	data.xlsx, budget.xlsx, data.xlsx, data.xlsx

In cryptanalysis, the attacker aims to convert the given secret code to plain text. To do this attacker needs to compute the correct base values and suitable replacement of files to corresponding values. Assumed that the possible base values are 2,4,9,17. The attacker applies all the possible combinations (brute force) to convert the given secret code to plain text. The cryptanalysis approach is explained as follows:

a. The attacker starts with base value 2.

The possible values are 0 and 1

List 0 replacement

In list 0 as article.docx appears in two places, and thesis.docx appears in two places, they must be replaced by the same values value, i.e., either 0 or 1.

The possible values of List 0 are {(1,0,0,1) (0,1,1,0)}.

List 1 replacement

In list 1 as data.xlsx appears three places, it must be represented with the same value, i.e., either 0 or 1.

The possible values in List 1 are {(1,0,1,1) (0,1,0,0)}.

The final values are {(1,0,0,1) (0,1,1,0)} {(1,0,1,1) (0,1,0,0)}.

Table 3 shows that the attacker tries to get the secret value by combining list 0 and list 1 to get list 2 but fail's to found secret value.

Table 3. List value to secret code generation brute force

Iterations	L0	L1	L2	
1	1,0,0,1	1,0,1,1	10011011	
2	1,0,0,1	0,1,0,0	10010100	**No Match**
3	0,1,1,0	1,0,1,1	01101011	
4	0,1,1,0	0,1,0,0	01100100	

b. Attacker increases the base value 4. The possible values are 0, 1, 2, 3.

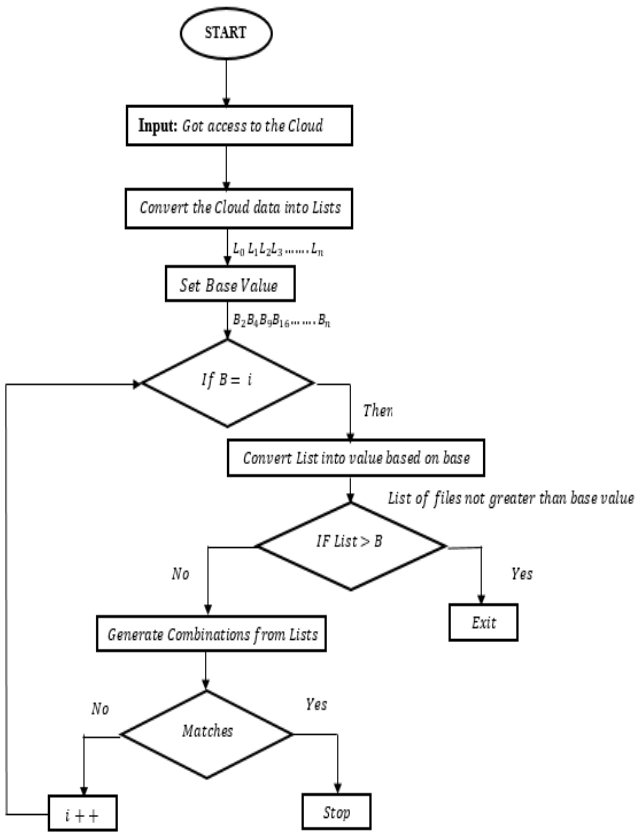

Figure 1. Brute force analysis flowchart

The list representation

List 1: - {(0,1,1,0) (0,2,2,0) (0,3,3,0) (1,0,0,1) (1,2,2,1) (1,3,3,1) (2,0,0,2) (2,1,1,2) (2,3,3,2) (3,0,0,3) (3,1,1,3) (3,2,2,3)}

List 2: - {(0,1,0,0) (0,2,0,0) (0,3,0,0) (1,0,1,1) (1,2,1,1) (1,3,1,1) (2,0,2,2) (2,1,2,2) (2,3,2,2) (3,0,3,3) (3,1,3,3) (3,2,3,3)}

The final values are {(0,1,1,0) (0,2,2,0) (0,3,3,0) (1,0,0,1) (1,2,2,1) (1,3,3,1) (2,0,0,2) (2,1,1,2) (2,3,3,2)(3,0,0,3) (3,1,1,3) (3,2,2,3)} {(0,1,0,0) (0,2,0,0) (0,3,0,0) (1,0,1,1) (1,2,1,1) (1,3,1,1) (2,0,2,2) (2,1,2,2) (2,3,2,2) (3,0,3,3) (3,1,3,3) (3,2,3,3)}.

Table 4. List value to secret code generation brute force

{(0,1,1,0,0,1,0,0)(0,1,1,0,0,2,0,0)(0,1,1,0,0,3,0,0)(0,1,1,0,1,0,1,1)(0,1,1,0,1,2,1,1)(0,1,1,0,1,3,1,1)(0,1,1,0,2,0,2,2)(0,1,1,0,2,1,2,2)(0,1,1,0,2,3,2,2)(0,1,1,0,3,0,3,3)(0,1,1,0,3,1,3,3)(0,1,1,0,3,2,3,3)} **Not Match**
{(0,2,2,0,0,1,0,0)(0,2,2,0,0,2,0,0)(0,2,2,0,0,3,0,0)(0,2,2,0,1,0,1,1)(0,2,2,0,1,2,1,1)(0,2,2,0,1,3,1,1)(0,2,2,0,2,0,2,2)(0,2,2,0,2,1,2,2)(0,2,2,0,2,3,2,2)(0,2,2,0,3,0,3,3)(0,2,2,0,3,1,3,3)(0,2,2,0,3,2,3,3)} **Not Match**
{(0,3,3,0,0,1,0,0)(0,3,3,0,0,2,0,0)(0,3,3,0,0,3,0,0)(0,3,3,0,1,0,1,1)(0,3,3,0,1,2,1,1)(0,3,3,0,1,3,1,1)(0,3,3,0,2,0,2,2)(0,3,3,0,2,1,2,2)(0,3,3,0,2,3,2,2)(0,3,3,0,3,0,3,3)(0,3,3,0,3,1,3,3)(0,3,3,0,3,2,3,3)} **Not Match**
{(1,0,0,1,0,1,0,0)(1,0,0,1,0,2,0,0)(1,0,0,1,0,3,0,0)(1,0,0,1,1,0,1,1)(1,0,0,1,1,2,1,1)(1,0,0,1,1,3,1,1)(1,0,0,1,2,0,2,2)(1,0,0,1,2,1,2,2)(1,0,0,1,2,3,2,2)(1,0,0,1,3,0,3,3)(1,0,0,1,3,1,3,3)**(1,0,0,1,3,2,3,3)**} **Match found**
{(1,2,2,1,0,1,0,0)(1,2,2,1,0,2,0,0)(1,2,2,1,0,3,0,0)(1,2,2,1,1,0,1,1)(1,2,2,1,1,2,1,1)(1,2,2,1,1,3,1,1)(1,2,2,1,2,0,2,2)(1,2,2,1,2,1,2,2)(1,2,2,1,2,3,2,2)(1,2,2,1,3,0,3,3)(1,2,2,1,3,1,3,3)(1,2,2,1,3,2,3,3)} **Not Match**
{(1,3,1,1,0,1,0,0)(1,3,1,1,0,2,0,0)(1,3,1,1,0,3,0,0)(1,3,1,1,1,0,1,1)(1,3,1,1,1,2,1,1)(1,3,1,1,1,3,1,1)(1,3,1,1,2,0,2,2)(1,3,1,1,2,1,2,2)(1,3,1,1,2,3,2,2)(1,3,1,1,3,0,3,3)(1,3,1,1,3,1,3,3)(1,3,1,1,3,2,3,3)} **Not Match**
{(2,0,0,2,0,1,0,0)(2,0,0,2,0,2,0,0)(2,0,0,2,0,3,0,0)(2,0,0,2,1,0,1,1)(2,0,0,2,1,2,1,1)(2,0,0,2,1,3,1,1)(2,0,0,2,2,0,2,2)(2,0,0,2,2,1,2,2)(2,0,0,2,2,3,2,2)(2,0,0,2,3,0,3,3)(2,0,0,2,3,1,3,3)(2,0,0,2,3,2,3,3)} **Not Match**
{(2,1,1,2,0,1,0,0)(2,1,1,2,0,2,0,0)(2,1,1,2,0,3,0,0)(2,1,1,2,1,0,1,1)(2,1,1,2,1,2,1,1)(2,1,1,2,1,3,1,1)(2,1,1,2,2,0,2,2)(2,1,1,2,2,1,2,2)(2,1,1,2,2,3,2,2)(2,1,1,2,3,0,3,3)(2,1,1,2,3,1,3,3)(2,1,1,2,3,2,3,3)} **Not Match**
{(2,3,3,2,0,1,0,0)(2,3,3,2,0,2,0,0)(2,3,3,2,0,3,0,0)(2,3,3,2,1,0,1,1)(2,3,3,2,1,2,1,1)(2,3,3,2,1,3,1,1)(2,3,3,2,2,0,2,2)(2,3,3,2,2,1,2,2)(2,3,3,2,2,3,2,2)(2,3,3,2,3,0,3,3)(2,3,3,2,3,1,3,3)(2,3,3,2,3,2,3,3)} **Not Match**
{(3,0,3,3,0,1,0,0)(3,0,3,3,0,2,0,0)(3,0,3,3,0,3,0,0)(3,0,3,3,1,0,1,1)(3,0,3,3,1,2,1,1)(3,0,3,3,1,3,1,1)(3,0,3,3,2,0,2,2)(3,0,3,3,2,1,2,2)(3,0,3,3,2,3,2,2)(3,0,3,3,3,0,3,3)(3,0,3,3,3,1,3,3)(3,0,3,3,3,2,3,3)} **Not Match**
{(3,1,3,3,0,1,0,0)(3,1,3,3,0,2,0,0)(3,1,3,3,0,3,0,0)(3,1,3,3,1,0,1,1)(3,1,3,3,1,2,1,1)(3,1,3,3,1,3,1,1)(3,1,3,3,2,0,2,2)(3,1,3,3,2,1,2,2)(3,1,3,3,2,3,2,2)(3,1,3,3,3,0,3,3)(3,1,3,3,3,1,3,3)(3,1,3,3,3,2,3,3)} **Not Match**
{(0,3,3,0,0,1,0,0)(0,3,3,0,0,2,0,0)(0,3,3,0,0,3,0,0)(0,3,3,0,1,0,1,1)(0,3,3,0,1,2,1,1)(0,3,3,0,1,3,1,1)(0,3,3,0,2,0,2,2)(0,3,3,0,2,1,2,2)(0,3,3,0,2,3,2,2)(0,3,3,0,3,0,3,3)(0,3,3,0,3,1,3,3)(0,3,3,0,3,2,3,3)} **Not Match**
{(3,2,3,3,0,1,0,0)(3,2,3,3,0,2,0,0)(3,2,3,3,0,3,0,0)(3,2,3,3,1,0,1,1)(3,2,3,3,1,2,1,1)(3,2,3,3,1,3,1,1)(3,2,3,3,2,0,2,2)(3,2,3,3,2,1,2,2)(3,2,3,3,2,3,2,2)(3,2,3,3,3,0,3,3)(3,2,3,3,3,1,3,3)(3,2,3,3,3,2,3,3)} **Not Match**

Table 4 shows that the attacker gets the secret value while working with base value 4, i.e., (1,0,0,1,3,2,3,3).

c. Attackers increase the base value 9. The possible values are 0, 1, 2, 3,4,5,6,7,8.

The list representation

List 0: - chapter.docx, thesis.docx, balancesheet.docx, chapter.docx

List 1: - Scheduling.xlsx, statistics.xlsx

Starts with base value 2. The possible values are 0, 1.

List 0 has more than two possible, so base 2 computation is not possible; then move to base value 4.

base value 4. The possible values are 0, 1, 2, 3.

List 0: - {(0,1,2,0) (0,1,3,0) (0,2,1,0) (0,2,3,0) (0,3,1,0) (0,3,2,0) (1,2,3,1) (1,2,0,1) (1,0,2,1) (1,0,3,1) (1,3,0,1) (1,3,2,1) (2,0,1,2) (2,0,3,2) (2,1,0,2) (2,1,3,2) (2,3,0,2) (2,3,1,2) (3,0,1,3) (3,0,2,3) (3,1,0,3) (3,1,2,3) (3,2,0,3) (3,2,1,3)}

List 1: - {(0,1) (0,2) (0,3) (1,0) (1,2) (1,3) (2,0) (2,1) (2,3) (3,0) (3,1) (3,2)}

The final values are {(0,1,2,0) (0,1,3,0) (0,2,1,0) (0,2,3,0) (0,3,1,0) (0,3,2,0) (1,2,3,1) (1,2,0,1) (1,0,2,1) (1,0,3,1) (1,3,0,1) (1,3,2,1) (2,0,1,2) (2,0,3,2) (2,1,0,2) (2,1,3,2) (2,3,0,2) (2,3,1,2) (3,0,1,3) (3,0,2,3) (3,1,0,3) (3,1,2,3) (3,2,0,3) (3,2,1,3) (0,1) (0,2) (0,3) (1,0) (1,2) (1,3) (2,0) (2,1) (2,3) (3,0) (3,1) (3,2)}. **No Match Found**

base value 9. The possible values are 0,1,2,3,4,5,6,7,8.

Table 5. List 0 representation

{(0,1)(0,2)(0,3)(0,4)(0,5)(0,6)(0,7)(0,8)**(1,0)**(1,2)(1,3)(1,4)(1,5)(1,6)(1,7)(1,8)(2,0)(2,1)(2,3)(2,4)(2,5)(2,6)(2,7)(2,8)(3,0)(3,1)(3,2)(3,4)(3,5)(3,6)(3,7)(3,8)(4,0)(4,1)(4,2)(4,3)(4,5)(4,6)(4,7)(4,8)(5,0)(5,1)(5,2)(5,3)(5,4)(5,6)(5,7)(5,8)(6,0)(6,1)(6,2)(6,3)(6,4)(6,5)(6,7)(6,8)(7,0)(7,1)(7,2)(7,3)(7,4)(7,5)(7,6)(7,8)(8,0)(8,1)(8,2)(8,3)(8,5)(8,6)(8,7)} **Match Found**
{(0,1,2,0)(0,1,3,0)(0,1,4,0)(0,1,5,0)(0,1,6,0)(0,1,7,0)(0,1,8,0)(0,2,1,0)(0,2,3,0)(0,2,4,0)(0,2,5,0)(0,2,6,0)(0,2,7,0)(0,2,8,0)(0,3,1,0)(0,3,2,0)(0,3,4,0)(0,3,5,0)(0,3,6,0)(0,3,7,0)(0,3,8,0)(0,4,1,0)(0,4,2,0)(0,4,3,0)(0,4,5,0)(0,4,6,0)(0,4,7,0)(0,4,8,0)(0,5,1,0)(0,5,2,0)(0,5,3,0)(0,5,4,0)(0,5,6,0)(0,5,7,0)(0,5,8,0)(0,6,1,0)(0,6,2,0)(0,6,3,0)(0,6,4,0)(0,6,5,0)(0,6,7,0)(0,6,8,0)(0,7,1,0)(0,7,2,0)(0,7,3,0)(0,7,4,0)(0,7,5,0)(0,7,6,0)(0,7,8,0)(0,8,1,0)(0,8,2,0)(0,8,3,0)(0,8,4,0)(0,8,5,0)(0,8,6,0)(0,8,7,0)(1,0,2,1)(1,0,3,1)(1,0,4,1)(1,0,5,1)(1,0,6,1)(1,0,7,1)(1,0,8,1)(1,2,0,1)(1,2,3,1)(1,2,4,1)(1,2,5,1)(1,2,6,1)(1,2,7,1)(1,2,8,1)(1,3,0,1)(1,3,2,1)(1,3,4,1)(1,3,5,1)(1,3,6,1)(1,3,7,1)(1,3,8,1)(1,4,0,1)(1,4,2,1)(1,4,3,1)(1,4,5,1)(1,4,6,1)(1,4,7,1)(1,4,8,1)(1,5,0,1)(1,5,2,1)(1,5,3,1)(1,5,4,1)(1,5,6,1)(1,5,7,1)(1,5,8,1)(1,6,0,1)(1,6,2,1)(1,6,3,1)(1,6,4,1)(1,6,5,1)(1,6,7,1)(1,6,8,1)(1,7,0,1)(1,7,2,1)(1,7,3,1)(1,7,4,1)(1,7,5,1)(1,7,6,1)(1,7,8,1)(1,8,0,1)(1,8,2,1)(1,8,3,1)(1,8,4,1)(1,8,5,1)(1,8,6,1)(1,8,7,1)(2,0,1,2)(2,0,3,2)(2,0,4,2)(2,0,5,2)(2,0,6,2)(2,0,7,2)(2,0,8,2)(2,1,0,2)(2,1,3,2)(2,1,4,2)(2,1,5,2)(2,1,6,2)(2,1,7,2)(2,1,8,2)(2,3,0,2)(2,3,1,2)(2,3,4,2)(2,3,5,2)(2,3,6,2)(2,3,7,2)(2,3,8,2)(2,4,0,2)(2,4,1,2)(2,4,3,2)(2,4,5,2)(2,4,6,2)(2,4,7,2)(2,4,8,2)(2,5,0,2)(2,5,1,2)(2,5,32)(2,5,4,2)(2,5,6,2)(2,5,7,2)(2,5,8,2)(2,6,0,2)(2,6,1,2)(2,6,3,2)(2,6,4,2)(2,6,5,2)(2,6,7,2)(2,6,8,2)(2,7,0,2)(2,7,1,2)(2,7,3,2)(2,7,4,2)(2,7,5,2)(2,7,6,2)(2,7,8,2)(2,8,0,2)(2,8,1,2)(2,8,3,2)(2,8,4,2)(2,8,5,2)(2,8,6,2)(2,8,7,2)(3,0,1,3)(3,0,2,3)(3,0,4,3)(3,0,5,3)(3,0,6,3)(3,0,7,3)(3,0,8,3)(3,1,0,3)(,1,2,3)(3,1,4,3)3,1,5,3)(3,1,6,3)(3,1,7,3)(3,1,8,3)(3,2,0,3)(3,2,1,3)(3,2,4,0)(3,2,5,3)(3,2,6,3)(3,2,7,3)(3,2,8,3)(3,4,0,3)(3,4,1,3)(3,4,2,3)(3,4,5,3)(3,4,6,3)(3,4,7,3)(3,4,8,3)(3,5,0,3)(3,5,1,3)(3,5,2,3)(3,5,4,3)(3,5,6,3)(3,5,7,3)(3,5,8,3)(3,6,0,3)(3,6,1,3)(3,6,2,3)(3,6,4,3)(3,6,5,3)(3,6,7,3)(3,6,8,3)(3,7,0,3)(3,7,1,3)(3,7,2,3)(3,7,4,3)(3,7,5,3)(3,7,6,3)(3,7,8,3)(3,8,0,3)(3,8,1,3)(3,8,2,3)(3,8,4,3)(3,8,5,3)(3,8,6,3)(3,8,7,3)(4,0,1,4)(4,0,2,4)(4,0,3,4)(4,0,5,4)(4,0,6,4)(4,0,7,4)(4,0,8,4)(4,1,0,4)(4,1,2,4)(4,1,3,4)(4,1,5,4)(4,1,6,4)(4,1,7,4)(4,1,8,4)(4,2,0,4)(4,2,1,4)(4,2,3,4)(4,2,5,4)(4,2,6,4)(4,2,7,4)(4,2,8,4)(4,3,0,4)(4,3,1,4)(4,3,2,4)(4,3,5,4)(4,3,6,4)(4,3,7,4)(4,3,8,4)(4,5,0,4)(4,5,1,4)(4,5,2,4)(4,5,3,4)(4,5,6,4)(4,5,7,4)(4,5,8,4)(4,6,0,4)(4,6,1,4)(4,6,2,4)(4,6,3,4)(4,6,5,4)(4,6,7,4)(4,6,8,4)(4,7,0,4)(4,7,1,4)(4,7,2,4)(4,7,3,4)(4,7,5,4)(4,7,6,4)(4,7,8,4)(4,8,0,4)(4,8,1,4)(4,8,2,4)(4,8,3,4)(4,8,5,4)(4,8,6,4)(4,8,7,4)(5,0,1,5)(5,0,2,5)(5,0,3,5)(5,0,4,5)(5,0,5,5)(5,0,6,5)(5,0,7,5)(5,1,0,5)(5,1,2,5)(5,1,3,5)(5,1,4,5)(5,1,6,5)(5,1,7,5)(5,1,8,5)(5,2,0,5)(5,2,1,5)(5,2,3,5)(5,2,4,5)(5,2,6,5)(5,2,7,5)(5,2,8,5)(5,3,0,5)(5,3,1,5)(5,3,2,5)(5,3,4,5)(5,3,6,5)(5,3,7,5)(5,3,8,5)(5,4,0,5)(5,4,1,5)(5,4,2,5)(5,4,3,5)(5,4,6,5)(5,4,7,5)(5,4,8,5)(5,6,0,5)(5,6,1,5)(5,6,2,5)(5,6,3,5)(5,6,6,5)(5,6,7,5)(5,6,8,5)(5,7,0,5)(5,7,1,5)(5,7,2,5)(5,7,3,5)(5,7,4,5)(5,7,6,5)(5,7,8,5)(5,8,0,5)(5,8,1,5)(5,8,2,5)(5,8,3,5)(5,8,4,5)(5,8,6,5)(5,8,7,5)(6,0,1,6)(6,0,2,6)(6,0,3,6)(6,0,4,6)6,0,5,6)(6,0,7,6)(6,0,8,6)(6,1,0,6)(6,1,2,6)(6,1,3,6)(6,1,4,6)(6,1,5,6)(6,1,7,6)(6,1,8,6)(6,2,0,6)(6,2,1,6)(6,2,3,6)(6,2,4,6)(6,2,5,6)(6,2,7,6)(6,2,8,6)(6,3,0,6)(6,3,1,6)(6,3,2,6)(6,3,4,6)(6,3,5,6)(6,3,7,6)(6,3,8,6)(6,4,0,6)(6,4,1,6)(6,4,2,6)(6,4,3,6)(6,4,5,6)(6,4,7,6)(6,4,8,6)(6,5,0,6)(6,5,1,6)(6,5,2,6)(6,5,3,6)(6,5,4,6)(6,5,7,6)(6,5,8,6)(6,7,0,6)(6,7,1,6)(6,7,2,6)(6,7,3,6)(6,7,4,6)(6,7,5,6)(6,7,8,6)(6,8,0,6)(6,8,1,6)(6,8,2,6)(6,8,3,6)(6,8,4,6)(6,8,5,6)(6,8,7,6)(7,0,1,7)(7,0,2,7)(7,0,3,7)(7,0,4,7)(7,0,5,7)(7,0,6,7)(7,0,8,7)(7,1,0,7)(7,1,2,7)(7,1,3,7)(7,1,4,7)(7,1,5,7)(7,1,6,7)(7,1,8,7)**(7,2,0,7)(**7,2,1,7)(7,2,3,7)(7,2,4,7)(7,2,5,7)(7,2,6,7)(7,2,8,7)(7,3,0,7)(7,3,1,7)(7,3,2,7)(7,3,4,7)(7,3,5,7)(7,3,6,7)(7,3,8,7)(7,4,0,7)(7,4,1,7)(7,4,2,7)(7,4,3,7)(7,4,5,7)(7,4,6,7)(7,4,8,7)(7,5,0,7)(7,5,1,7)(7,5,2,7)(7,5,3,7)(7,5,4,7)(7,5,6,7)(7,5,8,7)(7,6,0,7)(7,6,1,7)(7,6,2,7)(7,6,3,7)(7,6,4,7)(7,6,5,7)(7,6,8,7)(8,0,1,8)(8,0,2,8)(8,0,3,8)(8,0,4,8)(8,0,5,8)(8,0,6,8)(8,0,7,8)(8,1,0,8)(8,1,2,8)(8,1,3,8)(8,1,4,8)(8,1,5,8)(8,1,6,8)(8,1,7,8)(8,2,0,8)(8,2,1,8)(8,2,3,8)(8,2,4,8)(8,2,5,8)(8,2,6,8)(8,2,7,8)(8,3,0,8)(8,3,1,8)(8,3,2,8)(8,3,4,8)(8,3,5,8)(8,3,6,8)(8,3,7,8)(8,4,0,8)(8,4,1,8)(8,4,2,8)(8,4,3,8)(8,4,5,8)(8,4,6,8)(8,4,7,8)(8,5,0,8)(8,5,1,8)(8,5,2,8)(8,5,3,8)(8,5,5,8)(8,5,6,8)(8,5,7,8)(8,6,0,8)(8,6,1,8)(8,6,2,8)(8,6,3,8) (8,6,5,8)(8,6,5,8)(8,6,7,8)(8,7,0,8)(8,7,1,8)(8,7,2,8)(8,7,3,8) (8,7,5,8)(8,7,5,8)(8,7,6,8)

Table 6. List 1 representation

{(0,1)(0,2)(0,3)(0,4)(0,5)(0,6)(0,7)(0,8)**(1,0)**(1,2)(1,3)(1,4)(1,5)(1,6)(1,7)(1,8)(2,0)(2,1)(2,3)(2,4)(2,5)(2,6)(2,7)(2,8)(3,0)(3,1)(3,2)(3,4)(3,5)(3,6)(3,7)(3,8)(4,0)(4,1)(4,2)(4,3)(4,5)(4,6)(4,7)(4,8)(5,0)(5,1)(5,2)(5,3)(5,4)(5,6)(5,7)(5,8)(6,0)(6,1)(6,2)(6,3)(6,4)(6,5)(6,7)(6,8)(7,0)(7,1)(7,2)(7,3)(7,4)(7,5)(7,6)(7,8)(8,0)(8,1)(8,2)(8,3)(8,5)(8,6)(8,7)} **Match Found**

Tables 5 and 6 represent the possible **List 0** and **List 1** indexed file values. If we combine all the potential List 5 and 6, values, then we get the solution. All the combinations are only 36,288, in which the attacker gets the secret value, and this computation is very much less than B! *k! *n!

Attackers increase the base value 16. The possible values are 0, 1, 2, 3,4,5,6,7,8

The list representation

List 0: - redaction.docx tutorial.docx article.docx exercise.docx

Starts with base value 2. The possible values are 0, 1.

List 0 has more than two possible so base 2 computation is not possible then move to base value 4.

base value 4. The possible values are 0, 1, 2, 3 but results not match.

base value 9. The possible values are 0,1,2,3,4,5,6,7,8 but results not match

base value 16. The possible values are 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 and results match.

4. DISCUSSION

From the above computation, it is concluded that if an attacker got access to the Cloud and got the stored files, then he could compute the stored secret more diminutive than the exponential computations, i.e., very much less than $B! * K! * n!$. The following things must be incorporated into existing "distributed data hiding in the multi-cloud storage environment" to make brute force attack computationally exponential or equal to $B! * K! * n!$

1. Increasing the encoding base value must increase the number and the size of the file lists necessary for the secret dissimulation. It is observed that the increasing encoding base value decreases the number and the size of the file lists necessary for the secret dissimulation.
2. Multiple encoding must be used for the same secret message so that brute force attacks are computationally exponential. It is observed that only one base encoding technique is used for secretly distributing the secret message into multiple clouds.
3. Each Cloud must store the files of different lists, which are the results of different encoding techniques.

5. CONCLUSION

Distributed data hiding in the multi-cloud storage environment is a novel steganography concept designed with two technical contributions. I). cover media does not undergo any modification, i.e., the cover media act as a pointer to fragmented data. II). A secret message is stored in the multi-cloud storage environment. It claimed that it is complicated for an attacker to detect and extract the secret message. The paper analysed the security strength of the novel steganography concept and concluded that it is vulnerable to brute force attacks more diminutive than the exponential computations, i.e., very much less than $B! * K! * n!$. Further, the approach can be enhanced by three technical contributions. 1). Increasing the encoding base value must increase the file lists' number and size. 2). Multiple encoding must be used for the same secret message. 3). Each Cloud must store the files of different lists, resulting from different encoding techniques.

REFERENCES

[1] Muhammad, A.L., Riaz, A.S., Syed, R.H. (2020). Security analysis of network anomalies mitigation schemes in IoT networks. IEEE Access, 8: 43355-43374. https:// doi.org/10.1109/ACCESS.2020.2976624

[2] Katal, A., Dahiya, S., Choudhury, T. (2022). Energy efficiency in cloud computing data center: A survey on hardware technologies. Cluster Computing, 25: 675-705. https://doi.org/10.1007/s10586-021-03431-z

[3] Metcheka, L.M., Ndoundam, R. (2020). Distributed data hiding in multi-cloud storage environment. Journal of Cloud Computing: Advances, Systems and Applications, 68: 2-15. https://doi.org/10.1186/s13677-020-00208-4

[4] Ekodeck, S.G.R, Ndoundam, R. (2016). PDF steganography based on Chinese remainder theorem. Journal of Information Security and Applications, 29: 1-15. http://doi.org/10.1016/j.jisa.2015.11.008

[5] Sahu, A.K., Swain, G. (2020). Reversible image steganography using dua-layer LSB matching. Sensing Imaging, 21(1): 1. http:// doi.org/10.1007/s11220-019-0262-y

[6] Jiang, S.Z., Ye, D.P., Huang, J.Q., Shang, Y.Y., Zheng, Z.Y. (2020). Smart steganography: Light-weight generative audio steganography model for smart embedding application. Journal Netw Comput Appl., 165(7): 102689. https://doi.org/10.1016/j.jnca.2020.102689

[7] Pilania, U., Gupta, P. (2020). Analysis and implementation of IWT-SVD scheme for video steganography. Micro-Electronics and Telecommunication Engineering. 106: 153-162. https://doi.org/10.1007/978-981-15-2329-8_16

[8] Jackson, J.T., Gunsch, G.H., Claypoole, R.L., Lamont, G.B. (2003). Blind steganography detection using a computational immune system: A work in progress. Int J. Digit Evid., 4(1): 19.

[9] Koikara, R., Deka, D.J., Gogoi, M., Das, R. (2015). A novel distributed image steganography method based on block-dct. In: Advanced Computer and Communication Engineering Technology, 423-435. https:// doi.org/10.1007/978-3-319-07674-4_42

[10] Fendi, Wibisurya, A., Faisal. (2017). Distributed steganography using five-pixel pair differencing and modulus function. Procedia Comput Sci., 116: 334-341. https://doi.org/10.1016/j.procs.2017.10.085

[11] Liao, X., Wen, Q.Y., Shi, S. (2011). Distributed steganography. In: 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE, pp. 153-156. https://doi.org/10.1109/IIHMSP.2011.20

[12] Abdul, A.M., Mohammad, A.A.K., Reddy, V.K. (2022). Enhancing security of mobile cloud computing by trust- and role-based access control. Scientific Programming, 2022, Article ID 9995023. https://doi.org/10.1155/2022/9995023

[13] Chabbi, S., Boudour, R., Semchedine, F. (2020). A secure cloud password and secure authentication protocol for electronic NFC payment between ATM and smartphone. Ingénierie des Systèmes d'Information, 25(2): 139-152. https://doi.org/10.18280/isi.250201

[14] Yadav, A., Ritika, Garg, M.L. (2019). Monitoring based security approach for cloud computing. Ingénierie des Systèmes d'Information, 24(6): 611-617. https://doi.org/10.18280/isi.240608

[15] Mohammad, A. A. K., Mahmood, A. M., Vemuru, S. (2016). Secure energy efficient routing in manets by considering packet dropping reasons. Ponte International Journal of Science and Research, 72(11).