A Proposed Steganographic Method in Digital Media

Mr. Wisam A. Shukur
University of Baghdad
College of education / Ibn Al-Haitham

Abstract

The aim of this research is designing and implementing proposed steganographic method. The proposed steganographic method don't use a specific type of digital media as a cover but it can use all types of digital media such as audio, all types of images, video and all types of files as a cover with the same of security, accuracy and quality of original data, considering that the size of embedded data must be smaller than the size of a cover. The proposed steganographic method hides embedded data at digital media without any changing and affecting the quality of the cover data. This means, the difference rate between cover before hiding operation and stego is zero. The proposed steganographic method hides embedded data at various locations in cover irregularly or randomly, whereas the locations of cover for information hiding are not constant, this property will increase the level of security for proposed method. In the proposed method, the sender needs sending a file that has small size via any communication channel and that considered as a key while sending a cover is not necessary to recipient if both agree about downloading it from the internet before sending a file. The contents of this file are invaluable for an attacker. The programming language that used in programming this proposed method is C++ language. Steganographically, The proposed steganographic method is strong and robust. It is possible classifying this proposed method as a public key steganography system and substitution system at the same time.

1. Introduction

The more information is placed in the public's reach on the internet, the more owners of such information need to protect themselves from unwanted surveillance, theft, false representation and reproduction; they can use information hiding to protect themselves [1].

Today, it seems natural to use binary files with certain degree of irrelevancy and redundancy to hide data. Digital images, videos, and audio tracks are ideal for this purpose[2].

Steganography is an ancient art of conveying messages in a secret way that only the receiver knows the existence of the message [3].

Steganography literally means "covered writing", and is usually interpreted to mean hiding information in other information [4].

The embedded data should be invisible and inaudible to a human observer. Its goal isn't to restrict or regulate access to the host signal, but rather to ensure that the embedded data remain inviolated and recoverable [1].The term "cover-object" is used to describe the original, innocent message, data, audio, still video and so on. Information to be hidden in the cover data is known as the "embedded" data. The "stego-object" is the data containing both the cover and the embedded data.

2. Information Hiding Classification

Information Hiding can be classified into two types, Steganography and digital watermarking as shown in figure (1).

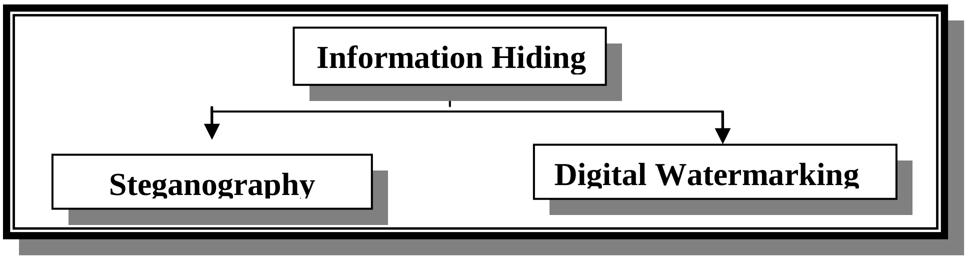


Figure (1): Information Hiding Classification

2.1 Steganography

Steganography is a Greek word which means "covered writing". It is a process that involves hiding a message in an appropriate covers for example an image, an audio file, video and so on. The cover can be sent to a recipient without anyone else knowing that it contains a hidden message. Steganography literally means "covered message" and involves transmitting secret messages through seemingly innocuous files. The goal is that not only does the message remain hidden, but also that a hidden message was even sent goes undetected [1]. Most applications of steganography follow one general principle, as illustrated in figure (2). Sender, who wants to share a secret message m with recipient randomly chooses (using the private random) a harmless message C called cover object, which can be transmitted to Recipient without raising suspicion, and embeds the secret message into C, probably by using key K, called stego-key. Sender therefore changes the cover C to a stego-object S. This must be done in a very careful way, so that third party, knowing only the apparently harmless message S, can not detect existence of the message [5].

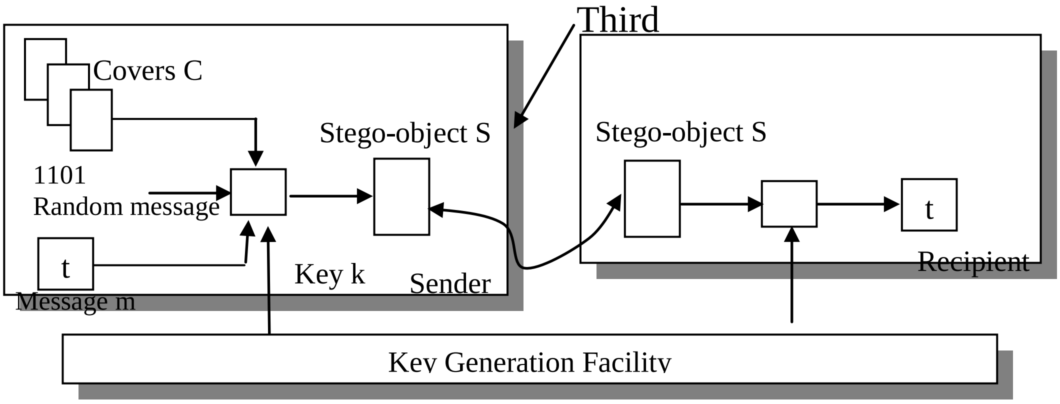


Figure (2): General Principle of Steganography

2.2 The Basic Model of Steganography System

Figure (3) shows the basic model of Steganography system, this model is called the embedding model. The input cover represents the untreated original data, Emb the ones which will be embedded into cover by function F_E. The resulting data called stego contain the message Emb. The operation F_E-1 extract the embedded data to Emb* and also produce an output Cover*. Naturally, Emb* should be equal to Emb and in most cases Cover* is the same as stego [6].

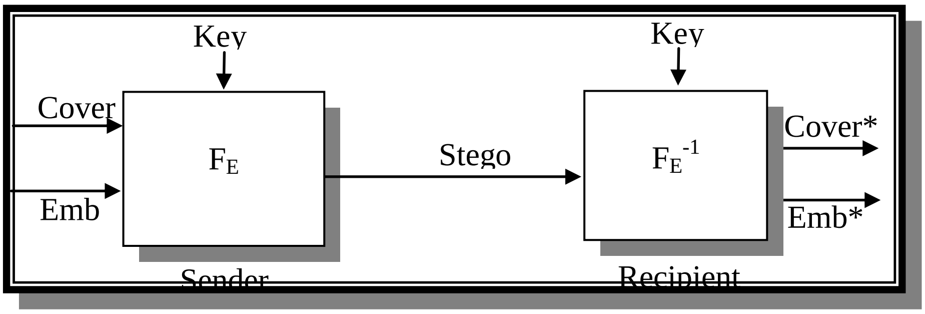


Figure (3): The Embedding Model

2.3 Categories of Steganography

There are several approaches classifying the steganographic systems. One could categorize them according to the type of covers used for secret communication. A classification according to the cover modifications applied in the embedding process is another possibility [4]. Steganographic methods can be divided into six categories**:**

1. **Substitution Systems**: substitute redundant parts of a cover with a secret message.

2. **Transform Domain Techniques:** embed secret information in a transform space of the signal (e.g. in frequency domain);
3. **Spread Spectrum Techniques**: adopt ideas from spread spectrum communication.
4. **Statistical Methods**: encode information by changing several statistical properties of cover and use hypothesis testing in the extraction process;
5. **Distortion Techniques:** store information by signal distortion and measure the deviation from the original cover in the decoding step;
6. **Cover Generation Methods**: encode information in the way a cover for secret communication is created.

3. The Proposed Steganographic Method

Many factors must be involved in the design of a good steganographic system, such as security, accuracy and capacity of storage for hidden data at digital media. The proposed steganographic method deals with these factors successfully. Whereas it can hide a large amount of information in digital media without exposing the covert communication to risk, while one of steganographic principles is if embedded data in the cover-media are large, then the risk of exposing the covert communication also increases at the same time. The proposed steganographic method embeds data from any type of digital media in any type of digital media. The cover and embedded data must be represented digitally. The proposed method works on media completely included the header of it without affecting the quality of the media.

3.1 The Architecture of Proposed Steganographic Method

The proposed method is a data-hiding steganographic method that uses digital media as a cover for embedded data. Its goal is to prevent the detection of a secret message that is embedded at digital media. The proposed steganographic method consists of two stages that are: 1. Embedding Stage 2. Extracting Stage, each stage will discuss and explain in the following sections of this research. The Architecture of proposed steganographic method is shown in figure (4).

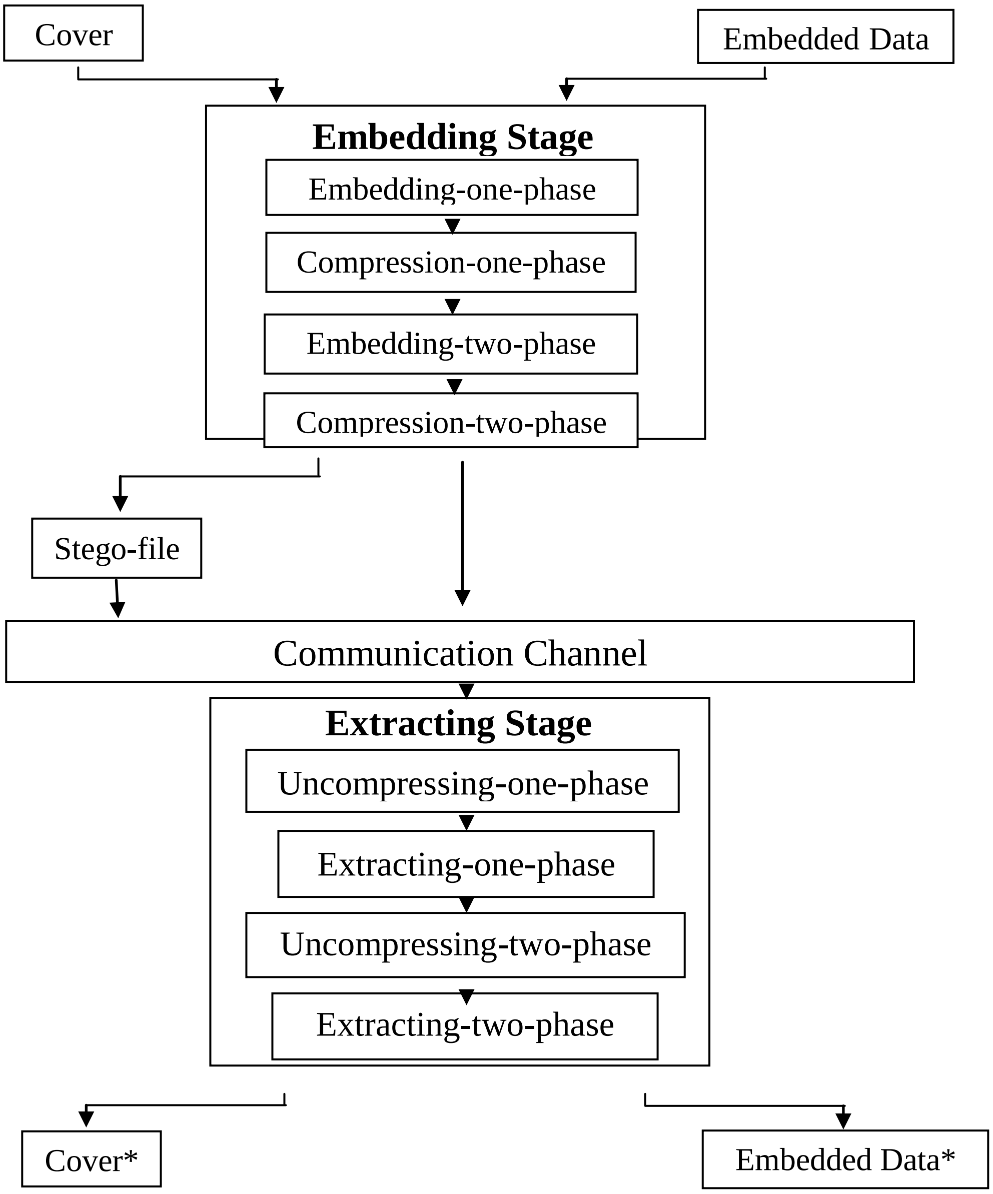


Figure (4): The Architecture of the proposed method

3.2 Embedding Stage

The proposed steganographic method hides the existence of message by transmitting information through various locations at digital media. For all of the steganographic methods, the most important and fundamental requirement is undetectability. The hidden message should not be detected by any other people. The proposed steganographic method can be used to cloak hidden data in different types of images, audio, video and even all types of files. The result of information hidden within a cover image is a stego-image, and the result of information hidden within a video is a stego-video and so forth. In this stage, all operations of information hiding must be performed by a sender completely. This stage consists of four phases that are: 1. Embedding-One-Phase 2.Compression-One-Phase 3.Embedding-Two-Phase 4.Compression-Two-Phase, these phases are shown in figure (4) and each one of them will explain in the next sections of this research.

3.2.1 Embedding-One-Phase

In this phase, the cover and embedded data will be selected; both must be compatible. The cover and embedded data must be represented digitally, the cover is divided into blocks; each block has size (8) bytes and it will store in two-dimension array, the number of rows equals (8) and the number of columns equals (8). The hiding process starts with dividing an array into two halves vertically, the first half is left half (LH) of array and the second half is right half (RH) of array. The embedded data are hided in right half (RH) of array, the map1 file is the output of this phase that contains addresses of hiding locations for embedded data in right half (RH) of array, such as sequences of (4, 5, 6, 7) values. This file will hide in left half (LH) of array via embedding-two-phase. The hiding algorithm of this phase is shown in figure (5).

3.2.2 Compression-One-Phase

In this phase, the output of previous phase that is "map1 file" will reduce according to compression table that is shown in table (1). The nature and type of map1 file's data gives the capability to compress contents of this file because the redundant contents. The contents of map1 file are sequences of (4, 5, 6, 7) values. Applying the compression table will reduce the size of file to 1/3 of the total size of file. All sequences of (4, 5, 6, 7) values in this file will substitute with new single letter; all single letters will store in "table-map1-file" then represent this new file digitally. All operations of this phase are shown in figure (6). The compression operation for map1 file increases the speed of map1 file hiding in the right half (RH) of array that will perform in the next phase.

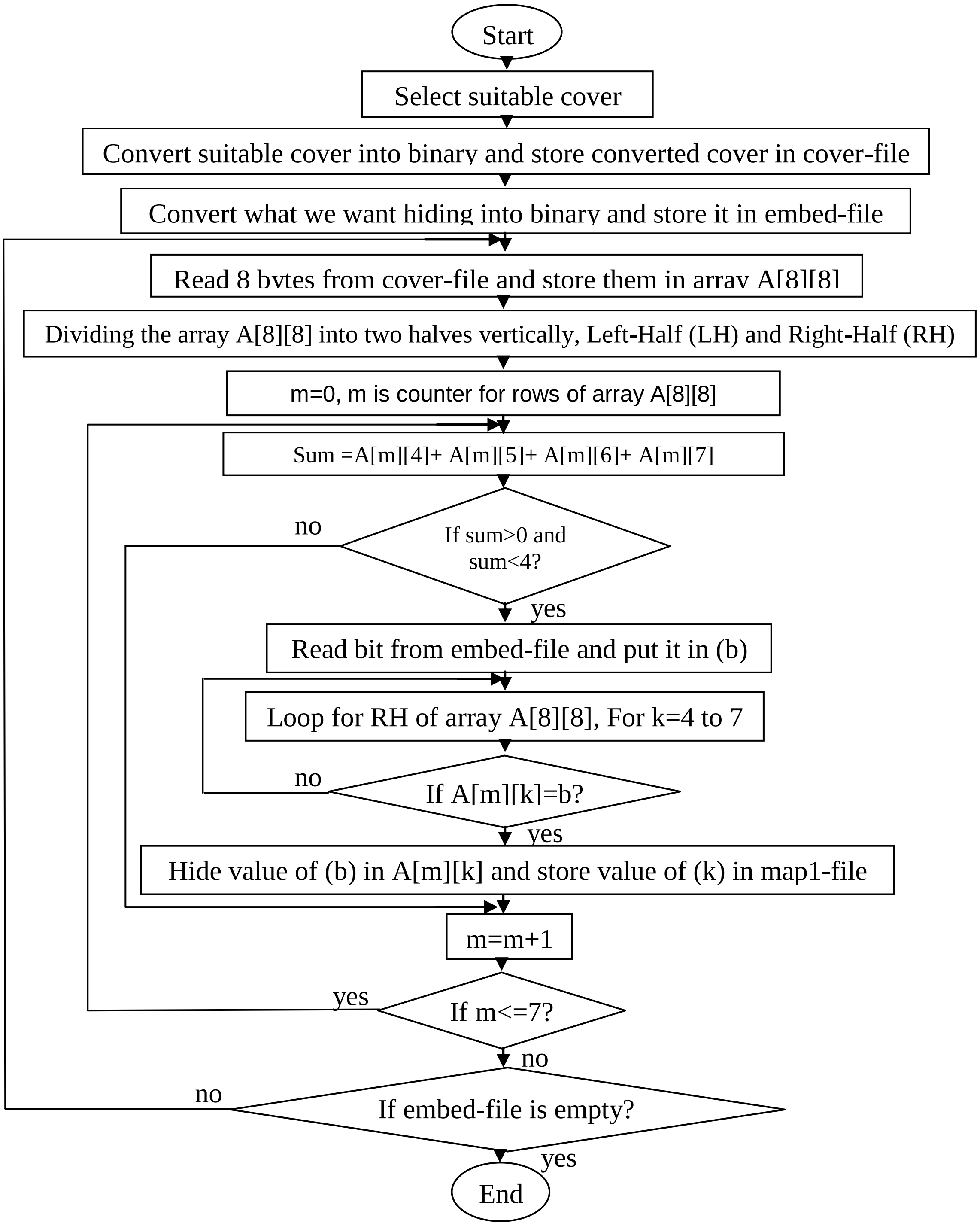
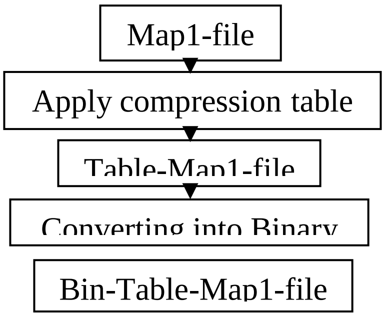

Start
Select suitable cover
Convert suitable cover into binary and store converted cover in cover-file
Convert what we want hiding into binary and store it in embed-file
Read 8 bytes from cover-file and store them in array A[8][8]
Dividing the array A[8][8] into two halves vertically, Left-Half (LH) and Right-Half (RH)
m=0, m is counter for rows of array A[8][8]
Sum =A[m][4]+ A[m][5]+ A[m][6]+ A[m][7]
If sum>0 and sum<4?
no
yes
Read bit from embed-file and put it in (b)
Loop for RH of array A[8][8], For k=4 to 7
If A[m][k]=b?
no
yes
Hide value of (b) in A[m][k] and store value of (k) in map1-file
m=m+1
If m<=7?
yes
no
If embed-file is empty?
no
yes
End

Figure (5): Embedding One Algorithm



▼

Figure (6): Block diagram of Compression-One-Phase

Sequences of values	Single letter
44-00	A-M
444-000	B-N
4444-0000	C-O
55-11	D-P
555-111	E-Q
5555-1111	F-R
66-22	G-S
666-222	H-T
6666-2222	I-U
77-33	J-V
777-333	K-W
7777-3333	L-X

Table (1): Compression Table

▼

3.2.3 Embedding-Two-Phase

In this phase, the Bin-Table-Map1-file will hide at various locations of left half (LH) of array according to the algorithm that is shown in figure (7). The output of this phase is map2 file that contains addresses of hiding locations for Bin-Table-Map1-file at left half (LH) of array. The recipient receive map2 file from sender through communication channel. The time of embedding Bin-Table-Map1-file is less than time of embedding Map1-file since Map1-file is not compressed, while Bin-Table-Map1-file is compressed by compression table as shown above.

3.2.4 Compression-Two-Phase

There are two operations in this phase that are compression table and winRAR compression.

1. Compression Table: The contents of map2 file are sequences of (0, 1, 2, 3) values. Applying the compression table that is shown in table (1) will reduce the size of file to 1/3 of the total size of file approximately. All sequences of (0, 1, 2, 3) values in this file are converted to a new single letter; all single letters are stored in "table-map2-file" then representing this new file digitally.

2. Compression by winRAR: the result of above step is "table-map2-file", we can apply winRAR program on it. The Compression rate by winRAR program is 1/3 of total size of file. The rate of total Compression (Compression table + winRAR program) is equal more than half of total size for file approximately. The result of this phase is"com-table-map2-file" and this file will transmit into recipient via a communication channel. All operations of this phase are shown in figure (8).

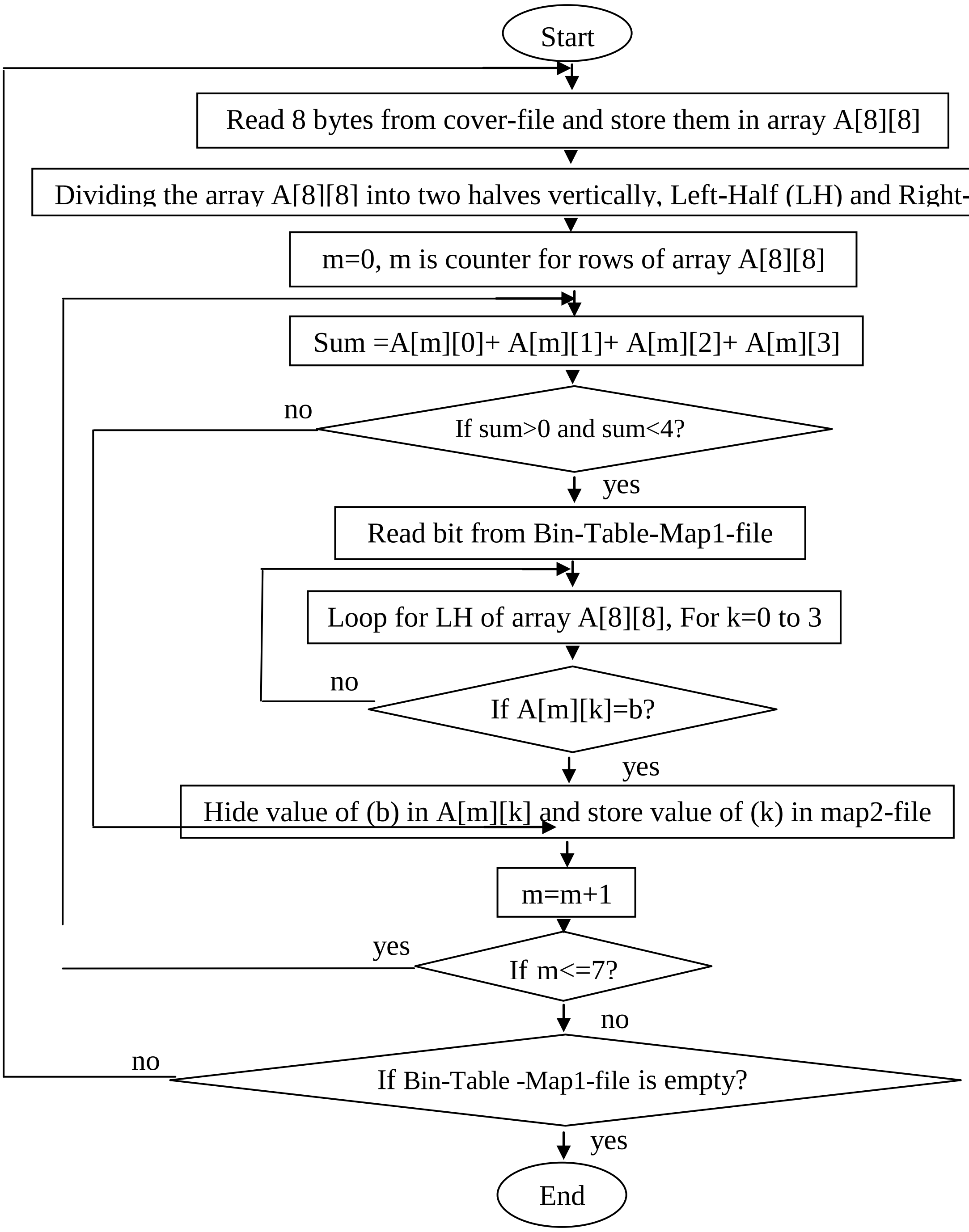

Start
Read 8 bytes from cover-file and store them in array A[8][8]
Dividing the array A[8][8] into two halves vertically, Left-Half (LH) and Right-Half (RH)
m=0, m is counter for rows of array A[8][8]
Sum =A[m][0]+ A[m][1]+ A[m][2]+ A[m][3]
no
If sum>0 and sum<4?
yes
Read bit from Bin-Table-Map1-file
Loop for LH of array A[8][8], For k=0 to 3
no
If A[m][k]=b?
yes
Hide value of (b) in A[m][k] and store value of (k) in map2-file
m=m+1
yes
If m<=7?
no
no
If Bin-Table -Map1-file is empty?
yes
End

Figure (7): Embedding Two Algorithm

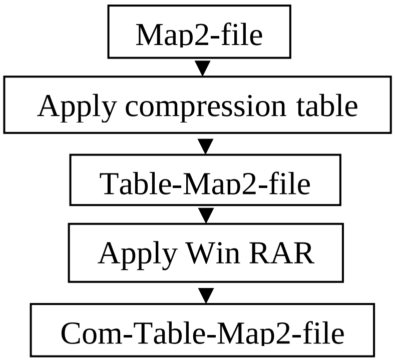


Figure (8): Block diagram of Compression-Two-Phase

3.3 Extracting Stage

In this stage, all operations of information extracting must be performed by a recipient completely. This stage consists of four phases that are: 1.Uncompressing-One-Phase 2.Extracting-One-Phase 3.Uncompressing-Two-Phase 4. Extracting-Two-Phase, these phases are shown in figure (4), each one of them will explain in the next sections.

3.3.1 Uncompressing-One-Phase

There are two operations in this phase that are uncompressing by applying winRAR program and uncompressing by table.

1. Uncompressing by winRAR: Applying winRAR program on "Com-Table-map2-file"to produce "Table-map2-file", another words, uncompress or remove compression of "Com-Table-map2-file" caused by winRAR program that performed in compression-two-phase as we have seen in the embedding stage previously. Now, "Table-map2-file" is back to its normal size that had before applying winRAR program on it.

2. Uncompressing by Table: The contents of Table-map2-file are collection of letters and numbers. Applying the table (1) will uncompress the compression of Table-Map2-file via substitution each single letter with equivalent sequence of values as illustrated in table (1) to produce map2-file that contains addresses of hidden locations for data of map1-file as shown in figure (9).the output of this phase represents the key that used to extract the original data that hided in embedding stage.

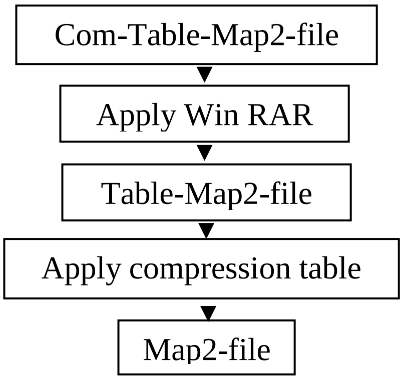


Figure (9): Block diagram of Uncompressing-One-Phase

3.3.2 Extracting-One-Phase

In this phase, the cover media and map2-file must be represented digitally; the cover media is divided into blocks, each block will store in two-dimension array, the number of rows equals (8) and the number of columns equals (8). The extracting process starts with dividing an array into two halves vertically, the first half is left half (LH) of array and the second half is right half (RH) of array. The map2-file used to extract map1-after-file that hided in left half (LH) of array. The extracting algorithm that used to extract map1-after-file is shown in figure (10). The map1-after-file is compressed file by compression table and it is the output of this phase that contains addresses of hiding locations for embedded data at right half (RH) of array. The recipient used map2-file as key in extracting process.

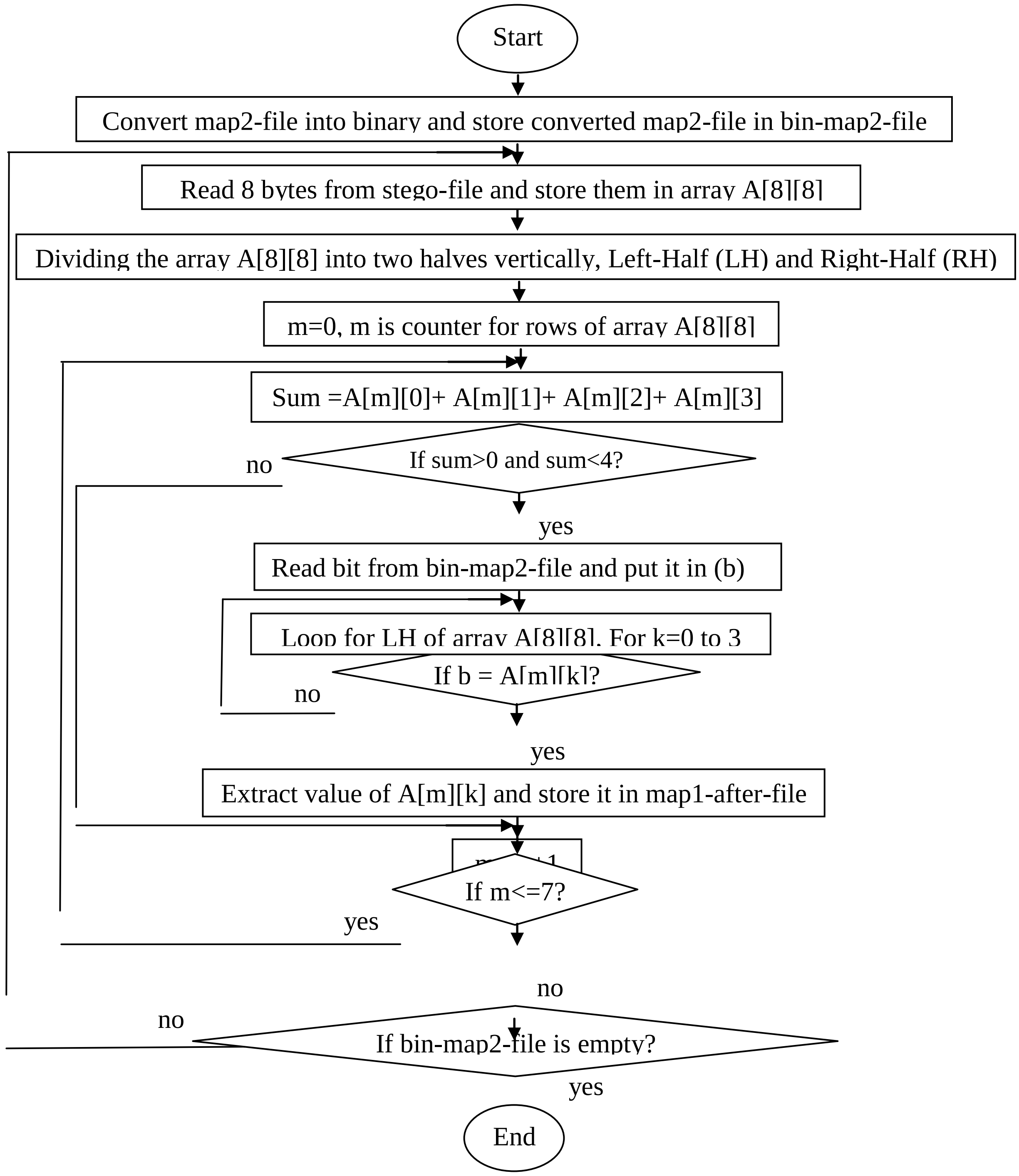

Start
Convert map2-file into binary and store converted map2-file in bin-map2-file
Read 8 bytes from stego-file and store them in array A[8][8]
Dividing the array A[8][8] into two halves vertically, Left-Half (LH) and Right-Half (RH)
m=0, m is counter for rows of array A[8][8]
Sum =A[m][0]+ A[m][1]+ A[m][2]+ A[m][3]
If sum>0 and sum<4?
no
yes
Read bit from bin-map2-file and put it in (b)
Loop for LH of array A[8][8], For k=0 to 3
If b = A[m][k]?
no
yes
Extract value of A[m][k] and store it in map1-after-file
m=m+1
If m<=7?
yes
no
no
If bin-map2-file is empty?
yes
End

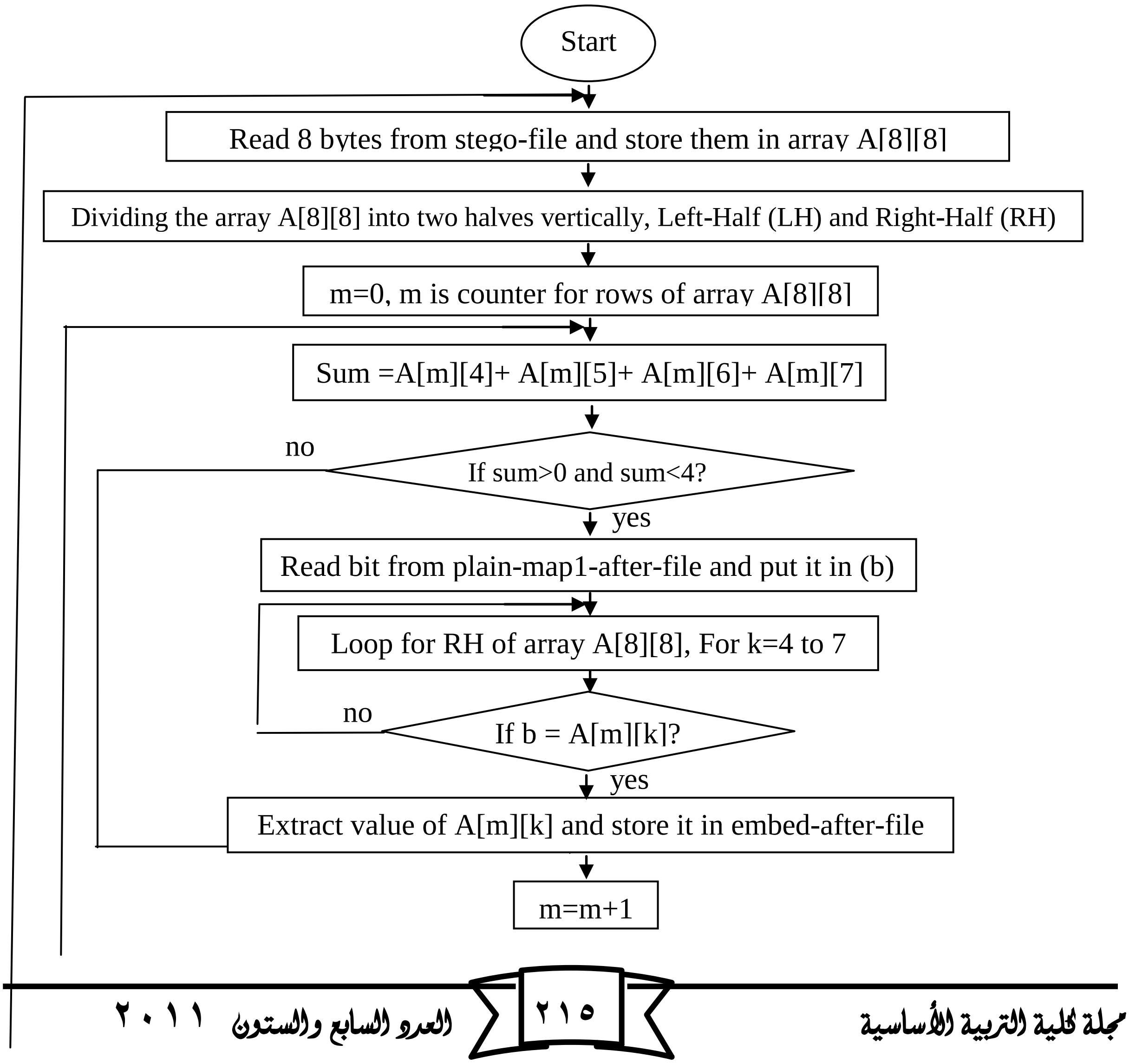
Figure (10): Extracting One Algorithm

3.3.3 Uncompressing-Two-Phase

The output of previous phase is an input for this phase; this file is map1-after-file that is compressed by compression table. In this phase, map1-after-file will be uncompress by using table (1) to produce Plain-map1-after-file that contains addresses of hidden locations for embedded data at right half (RH) of array. Plain-map1-after-file used in next phase to extract embedded data that hidden in right half (RH) of array.

3.3.4 Extracting-Two-Phase

In this phase, the cover media and Plain- Map1-After-file must be represented digitally; the stego media is divided into blocks, each block will store in two-dimension array, the number of rows equals (8) and the number of columns equals (8). The extracting process starts with dividing an array into two halves vertically, the first half is left half (LH) of array and the second half is right half (RH) of array. The plain-map1-after-file used as key with proposed algorithm that is shown in figure (11) to extract embedded data that are hided at right half (RH) of array.


Start
Read 8 bytes from stego-file and store them in array A[8][8]
Dividing the array A[8][8] into two halves vertically, Left-Half (LH) and Right-Half (RH)
m=0, m is counter for rows of array A[8][8]
Sum =A[m][4]+ A[m][5]+ A[m][6]+ A[m][7]
no
If sum>0 and sum<4?
yes
Read bit from plain-map1-after-file and put it in (b)
Loop for RH of array A[8][8], For k=4 to 7
no
If b = A[m][k]?
yes
Extract value of A[m][k] and store it in embed-after-file
m=m+1

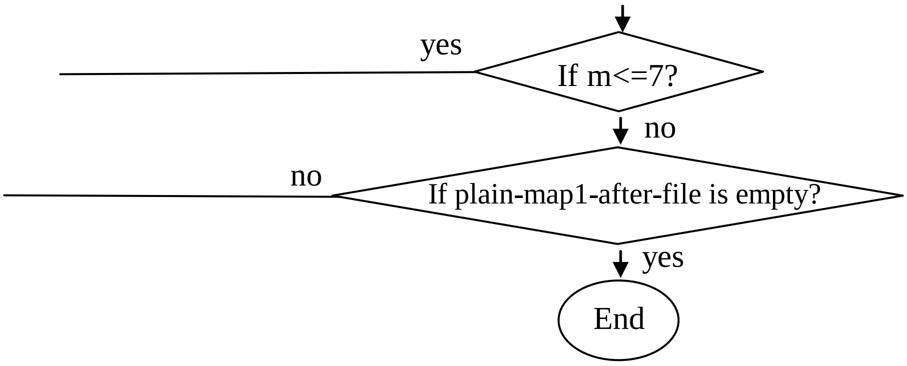
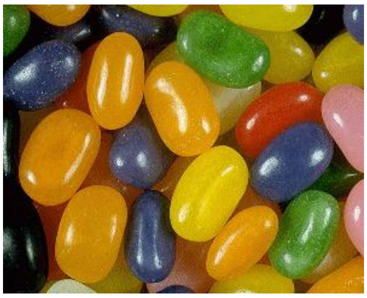
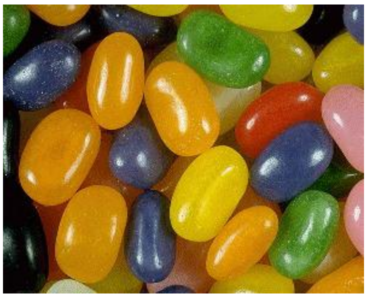


Figure (11): Extracting Two Algorithm



Before hiding



After hiding

The type of above image is (bmp) while the type of below image is (Gif) and so on for other types of images.

Before hiding

After hiding

So**,** the proposed steganographic method applied on audio file, movie, and different types of files such as *.txt, *.PDF and so on with same level of security and accuracy.

4. Conclusions

1. The proposed steganographic method can hide data from any type of digital media such as text, all types of images, audio and video that used as cover are give the same level of security and accuracy.
2. The proposed steganographic method can hide embedded data in digital media without any changing and affecting the quality of the data of digital media that used as cover completely.
3. The proposed steganographic method can hide embedded data in headers and features of digital media that used as cover. Therefore, the information hiding storage becomes too much with this method.
4. It is not necessary sending a cover if sender and recipient agree about a specific type of digital media and then downloading it from the internet at any time before sending "com-table-map2-file" by sender to recipient.
5. Designing or developing the methods of compression [compression table and winRAR program] used in this research to increase the rate of compression.
6. Designing a proposed method to a void sending the "com-table-map2-file" from sender to recipient via communication channel finally.

5. References

1. Johnson, Neil F., Duric, Zoron, Jajodia, *“Information Hiding Steganography and Watermarking–Attacks and Countermeasures”,* Kluwer Academic Publishers, 2001.

2. Fridrich, J., *“Application of Data Hiding in Digital Image”*, 1998, PDF.

3. D. Kahn, The Code breakers-the comprehensive history of secret communication from ancient times to the Internet, Scribner, New York (1996).

4. Katzenbeisser S. and Petitcolas F.(eds), *“Information Hiding Techniques for Steganography and Digital Watermarking”,* Artech House, USA 2000.

5. Johnson, Neil F. and Jajodia, Sushil, *“Steganalysis of Images Created Using Current Steganography Software”*, Information Hiding: Second International Workshop, proceeding, vol. 1525 of lecture notes in computer science, springger, 1998, pp. 273-289.

6. Zollner J., H. Federrath H., Klimant H., Pfitzman A., Piotraschke R., Westfeld A., Wicke G., and Wolf G., *“Modeling the Security of Steganography System”,* Information Hiding: Second International Workshop, proceeding, vol. 1525 of lecture notes in computer science, springer, 1998, pp. 344-354.

طريقة مقترحة للإخفاء المعلوماتي في الوسائط الرقمية

الخلاصة

وسام عبد شكر

جامعة بغداد - كلية التربية – ابن الهيثم

الهدف من هذا البحث هو تصميم وتنفيذ طريقة مقترحة للإخفاء ألمعلوماتي. إن طريقة الإخفاء ألمعلوماتي المقترحة لا تستخدم نوعا محددا من الوسائط الرقمية كغطاء، ولكنها تستخدم جميع أنواع الوسائط الرقمية مثل الملفات الصوتية، كافة أنواع الصور، مقاطع الفيديو وجميع أنواع الملفات كغطاء وبنفس مستوى الأمنية والدقة ونوعية البيانات الأصلية(الغطاء). آخذين بنظر الاعتبار إن حجم البيانات المراد إخفاؤها يجب أن يكون اصغر من حجم الغطاء. إن طريقة الإخفاء ألمعلوماتي المقترحة تخفي البيانات في الوسائط الرقمية بدون أي تغيير وتأثير على نوعية بيانات الغطاء. هذا يعني إن نسبة الفرق بين الغطاء قبل إجراء عملية الإخفاء وبين الغطاء بعد إجراء عملية الإخفاء تساوي صفر. إن هذه الطريق تخفي البيانات في مواقع مختلفة في الغطاء وبصورة غير منتظمة أو عشوائية. حيث أن مواقع الغطاء لإخفاء البيانات ليست ثابتة. وهذه الميزة تزيد من مستوى الأمنية لهذه الطريقة المقترحة.في هذه الطريقة المقترحة، المرسل يحتاج إلى إرسال ملف له حجم صغير خلال قناة اتصال معينة، والذي يمكن اعتباره كمفتاح في حين إرسال الغطاء ليس ضروريا إلى المستلم إذا اتفق كلاهما على تحميله من الانترنيت قبل إرسال الملف. إن محتويات هذا الملف غير ذات قيمة بالنسبة للمهاجم. لغة لبرمجة المستخدمة في برمجة هذه الطريقة هي لغة سي++. إن طريقة الإخفاء ألمعلوماتي المقترحة والتي اشعر إنها قوية في الإخفاء ألمعلوماتي ومقاومة لعمليات المهاجم عليها. انه من الممكن تصنيف هذه الطريقة المقترحة كنظام إخفاء ذا المفتاح العام ونظام الإخفاء التعويضي في نفس الوقت.