A Novel Video Steganography Technique using Dynamic Cover Generation

Vivek Sampat
Saraswati College of Engineering
Kharghar,
Navi Mumbai,
Maharashtra, India

Kapil Dave
PVPPCOE, Sion,
Mumbai,
Maharashtra, India

Jigar Madia
SAKEC, Chembur,
Mumbai,
Maharashtra, India

Parag Toprani
Siddharth College of Engineering,
Hyderabad, Andhra Pradesh, India

ABSTRACT
Since the dawn of technology, communication has always been in need of novel techniques of data security. The confidentiality, integrity and authenticity of data are of prime importance while communicating over a network. To satisfy these requirements, we propose a steganographic system using an advanced approach which generates a personalised video cover to hide the data within itself in accordance to multiple factors such as the data to be transmitted and key selected. This approach will make sure that the transfer of data becomes more secure against security breaches providing privacy and safe communication environment.

General Terms
Experimentation, Algorithm

Keywords
Digital Steganography, Digital Video, Cover, Block-Based Transformation

1. INTRODUCTION
Confidentiality is a crucial communication issue in diverse environments. In academia, for example, the details of a research are often top secret, as well as in the business world, where those details are related to the development of a product. According to Siponen et al. [1], a major challenge in communication with information technology tools is related to information security. Specifically Security and privacy issues have long been investigated in the context of a single organization exercising control over its users' access to resources [2]. Due to the issues in the field of information security methods like cryptography and steganography have been used. However, application of cryptography implies the awareness of a secret cipher [12]. Though the cipher may not break but, it may still be possible to intercept it and to corrupt the message, making the information useless. A complementary approach to deal with this security question is to hide the secret information in a way that users are not aware of its existence, i.e. as users do not know about the information, the secrecy is kept. This is done through steganography.

Steganography is the art and science of hiding communication; a steganographic system thus embeds hidden content in unremarkable cover media so as not to arouse an eaves dropper's suspicion. Today, computer and network technologies provide easy-to-use communication channels for steganography. The majority of work in literature reports on the use of steganography on compressed images as a cover. However, there is a limitation on how much information can be hidden into an image [4], makes it difficult to use those methods. Since video can be seen as a collection of images, a possible solution to overcome that limitation is to use video as a media for information hiding. Two main advantages in the use of videos for steganography are:
It is possible to hide a higher volume of information. It is harder to find the secret (into which video frame it would be the secret?).
Most steganographic systems today hide messages by slightly modifying an existing cover object, such as a digital image or a video. In such system cover used to be only a mere carrier of information.
In this research we propose a new steganographic system wherein the cover media itself is generated by the system instead of using an existing cover and partial of data is the cover itself and rest is embedded inside the cover. This makes steganalysis more difficult as what data cover itself means is difficult to judge thus improving security.
This paper is organized as follows: Section 2 we describe related work, Section 3 describes the proposed architecture for security and Section 4 presents the developed steganography technique.

2. RELATED WORK
Steganographic systems evolved into a lot of levels. The first level of protection is determined only by the choice of embedding algorithm. This may be the least significant bits modification algorithm, or algorithms for modifying the frequency or spatial-temporal characteristics of the container.
The second protection level of the steganographic system, as well as all levels of protection of the higher orders, is characterized by the use of Key (password) via steganographic modification. Steganographic data channels that use key schemes based distribution of a message through

the container and or preprocessing of an embedded message for data hiding are more secure. When the third protection level key scheme is used it affects the distribution of a message through the container.

The difference between the fourth protection level scheme and the third one is that in steganographic system there are two distribution functions of a message within a container are used. The first is responsible for a message samples selection according to some function G (Q, N), and the second function F (P, L) is responsible for position selection in a container for message sample hiding. Here Q – the size of message block to be inserted; N – the size (in bits) of one sample of the message file.

Problems with the existing systems:

- In this system the cover file acts as a mere carrier of message and is a big overload in videos.
- The system cannot encompass two algorithms at once
- As all of it is hidden across the cover file using a single algorithms. If this method of hiding breaks then there is no other security.

Table 1. Key Steganographic Schemes Classification

Steganographic system protection level	SAU	KU	KIMD	KIMSD	KIC
1	+	-	-	-	-
2	+	+	-	-	-
3	+	+	+	-	-
4	+	+	+	+	-
5	+	+	+	+	+

- SAU-Steganographic algorithm usage
- KU-Key (password) usage
- KIMD-Key influence on a message distribution
- KIMSD-Key influence on a message selection and distribution
- KIC- Key influence on cover generation

3. PROPOSED ARCHITECTURE

Steganographic [3] data channels that use key schemes based distribution of a message through the container and or\ preprocessing of an embedded message for data hiding are more secure. In our proposed system the key and message are used to generate a cover video using some function X(A,D) where X is the function to generate the container file using partial message A is the number of samples required to hide the message and D is bits in message to be hidden.

The following notations are used in the architecture:
c - is a container file;
F - steganographic channel space;
SC - steganographic system;
m - message to be embedded;
E - embedding method;
ĉ - modified container file.

F (P, L) – distribution function of a message within a container; P – minimum number of container samples that are needed to embed one message sample; L – step of a message distribution within a container. In the proposed system there are two distribution functions of a message within a container are used. The first is responsible for a message samples selection according to some function G (Q, N), and the second function F (P, L) is responsible for position selection in a container for message sample hiding. Here Q – the size of message block to be inserted; N – the size (in bits) of one sample of the message file. The function F(P,L) and G(Q,N) also X(A,D) are customizable according to the developer of the system. The function F(P,L) and G(Q,N) relate to the digital steganography algorithm used. The function X(A,D) relates to the cover generation process discussed hereafter.

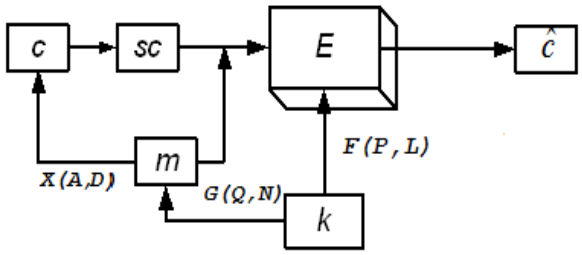

Fig. 1 General Architecture (Level 5)[3]

On the basis of the general architecture we base our proposed architecture in Fig.2 and Fig. 3.

The proposed technique is initiated with verifying all the inputs which are required. The inputs are provided to the Preprocessing block which does 3 tasks. It divides the key into 3 sub keys as required. Second step is generating the allocation table which will make a sub database of 256 images. This set of 256 images is used to hide up to 4MB of data. If the data size is larger than 4MB, it will generate another table. The 256 image set is again broken up into 4 set of 64 images and each set of 64 again into 2 sets of 32 images. The third step is dividing the data file into chunks of 32 bytes. The chunks are shuffled using a suitable block based transformation and then each chunk is divided into 2 parts iChunk and dChunk. The size of iChunk is 10 bits.

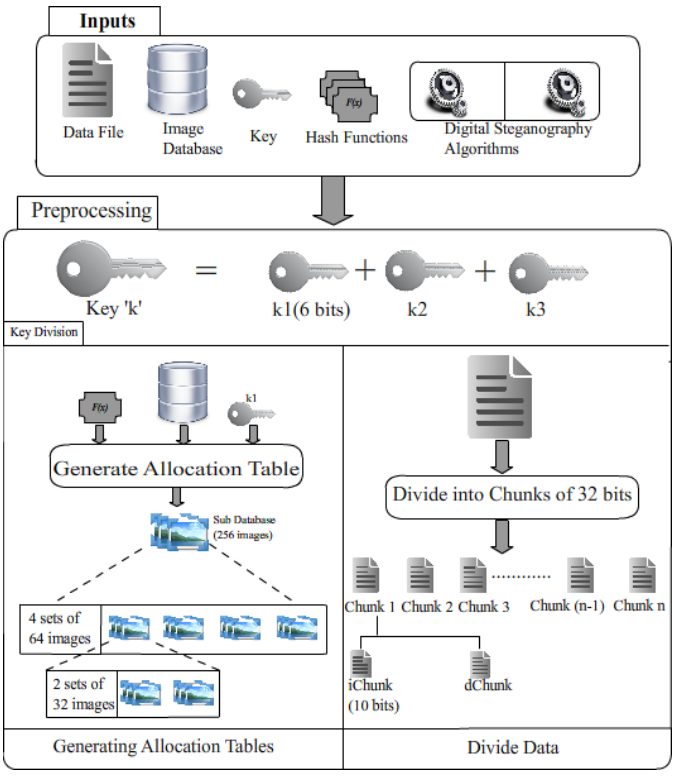

Fig. 2 Inputs and Preprocessing

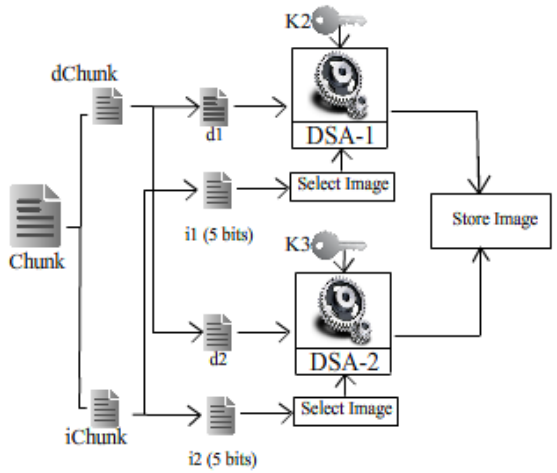

Fig. 3 Data Hiding

Next block is data hiding in which we select image from allocation table and hide the data chunks in it. Each data chunk takes two images and the images are selected using the iChunk. The iChunk is divided into 2 of 5 bits and it is used to select the 2 images from the allocation table. The dChunk is divided into half and hidden in the images selected. After all the data chunks are hidden, the images are collected back and made into a digital video. We set the appropriate parameters and add dummy audio to the video to make it blend with the normal videos. Due to the use of key for cover generation the resulting output may be distorted however suspicion can be reduced by providing a database of relevant images. Also if the selected dummy audio mentioned above has relevant to the images the chances of suspicion reduces further.

4. PROPOSED ALGORITHM

The proposed algorithm is divided into 3 procedures. Initial required inputs are Image databases, data file to be transmitted and the key selected by the user. The algorithm works for a data file of 4MB and for next 4MB of data it generates different set of images thus providing flexibility. The Bitwise shift operator shifts the key k to left by 6 bits for each 4MB of data.

A. *Procedure 1:Inputs*

1. A database of images common to sender and receiver. The images minimum size and dimension of the images used depend on the requirement of the digital steganography algorithm used.
2. Data file to be transmitted
3. Two digital steganography algorithms of image and a Block Based Algorithm common to sender and receiver
4. Key 'k' of size Key1+Key2+6 bits where Key1 and Key2 are key sizes required for digital Steganography algorithms.
5. Hash functions common to sender and receiver.

End Procedure

B. *Procedure 2:Preprocessing*

Input: Image Database 'I', Data File 'F', Key 'k'

Output: Allocation Table, Data chunks, Key k1, k2, k3

Sub DivideKey ()

1. Accept key 'k' from user
2. Divide it into 3 parts such that

Key k1 – First 6 bits of k

Key k2 and k3 are both derived by dividing the remaining bits into equal parts.

End Sub

Sub GenerateAllocationTable ()

1. Select the first image from database by taking the 'k1' th image where k1 is the 6 bit key.
2. Hide the generated random seed value in the image selected.
3. For each i of 1 to 255

Select the function f(x) from multiple options based on the value of first 2 bits of k1.

Table 2. Function Selection

Bit 1	Bit 2	Function Selected
0	0	F1
0	1	F2
1	0	F3
1	1	F4

Where F1, F2, F3, F4 are hash functions.

Use the hash function f(x) and the seed value to select an image from database.

If (Image is repeated)

Take 3 bits of the remaining 4 bits of k and based on last bit decide to add or subtract 3 bit value to hash value.

If (Hash value out of range)

Use a cyclic function to get the value in the range

End If
End If
End For

4. Each set of 256 images will be used to hide exactly 4MB of data.
5. Divide the 256 images selected from database into 4 sets S1,S2,S3,S4 of 64 images and further divide each set of 64 image into 2 sets S(n)a and S(n)b of 32 images where n is 1..4 .
6. Assign a 5 bit op-code to each image in each 32 bit set where op-code is the binary value assigned in sequence of their occurrence in the set.

End Sub

Sub DataDivision ()

1. Divide data into chunks of size 32 bytes.
2. Apply any Block Based transformation on chunks to interchange the bits within the chunk.
3. Divide the chunk into iChunk of size 10 bits and dChunk of remaining bits.

End Sub

Procedure Preprocessing()

1. For each 4MB of data
   DivideKey ()
   GenerateAllocationTable ()
   DataDivision ()
   End For

End Procedure

C. *Procedure 3: Hide Data*

Input: S1, S2, S3, S4, Data chunks, k1, k2, k3
Output: Images with hidden data

Procedure HData ()

1. Select each S (n) where n = 1...4 for 64 images in a round robin.
2. For each Data Chunk
3. Divide the iChunk into i1 and i2each of 5 bits.
4. Select the image x1 from S(n)a by matching the op-code and x2 from S(n)b.
5. While (op-code is repeated)
6. Increment i with 1 bit in a cyclic function of 32.
7. End While
8. Divide dChunk into 2 equal parts and apply any digital steganography algorithm to hide each dChunk part into x1 and x2 using k2 and k3 respectively. The two steps for x1 and x2 can use any algorithms to provide digital Steganography.
9. End For

End Procedure

Apply HData () for each 4MB of data.

D. *Procedure 4: Generate Video*

Input: Steganographed Images
Output: Steganographed Video

Procedure GenerateVideo ()

1. Recollect all steganographed images and generate a digital video.
2. Set the FPS as 32 thus generating 1s of video for each chunk and a 128s of video for a 4MB data.
3. Add dummy audio.
4. Generate checksum of key and data and store it in the dynamically selected frame.

End Procedure

The above procedures generate a digital video with hidden data which can be safely transmitted on a network. The receiver uses the video and key given by sender along with the image database, hash functions and steganography algorithms which are common to both parties. To recover the data we perform the procedures in reverse and recollect the data. We regenerate the allocation table and compare images to get the data from their difference. The chunks are then regrouped back to the data file.

5. EXPERIMENTAL RESULTS

The proposed algorithm focuses on generation of a digital video to hide data and can use available options to hide data as required by the user. The primary implementation of the algorithm is done using the JAVA programming language and the database used is Oracle SQL Plus. Experimental results were obtained on Intel i3 2.20 GHz processor hardware with 4 GB of RAM. The operating system used was Windows 7. We implement the algorithm and after that we experiment the implementation. The key used was a 128 bit passkey for sample and a basic digital steganography algorithm was selected to hide data into images Experimental results showed that the videos were able to run on fairly all players with no visible sign of hidden data. It also did not yield results of steganalysis of steganography algorithms confirming its high level of security. The advantage of generating our own cover was one of the reasons of successful sustenance of steganalysis which did not provide the intruder with the original images to derive data.

6. CONCLUSION

To summon up creation of the digital video along with adaptable features to input data are sure features to make this algorithm a very good Steganography approach and provide it with very good opportunities. In future this algorithm can be made to hide data in audio, use higher mathematical functions, better validation along with increase in number of frames.

7. ACKNOWLEDGEMENT

The Authors would like to thank the institutes: Saraswati College of Engineering, Padmabhushan Vasantdada Patil Pratihans College of Engineering and Shah and Anchor Kutchhi Engineering College for providing its valuable resources and assistance to the authors in their research work.

8. REFERENCES

[1] Siponen, M. T. and Oinas-Kukkonen, H. 2007. A review of information security issues and respective research contributions. SIGMIS Database 38, 1 (Feb. 2007), 60-80.

[2] Yang, S. 2003 Security and Trust Management in Collaborative Computing. Doctoral Thesis. UMI Order Number: AAI3127699, University of Florida.

[3] Wikipedia [Online]. Available: http://www.wikipedia.org/Steganography

[4] C. L. Chang, et al. "A steganographic method based upon JPEG and quantization table modification". An International Journal - Information Science., 2000.

[5] Wang, C., et. al "New image steganographic methods using un-length approach". 2004, Information Science, Elsevier Inc. All Rights Reserved, p. 16, 2004.

[6] I. Avcibas, N. Memon, and B. sankur, \Steganalysis using image quality metrics." Security and Watermarking of Multimedia Contents, San Jose, Ca., Feruary 2001.

[7] J. Fridrich and M. Goljan, \Digital image steganography using stochastic modulation," SPIE Symposium on Electronic Imaging, San Jose, CA,, 2003.

[8] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, \Quantitive steganalysis of digital images: Estimating the secret message lenght," ACM Multimedia Systems Journal, Special issue on Multimedia Security, 2003.

[9] Petitcolas, FAP; Anderson RJ; Kuhn MG (1999). "Information Hiding: A survey" (pdf). Proceedings of the IEEE (special issue) 87 (7): 1062–78.doi:10.1109/5.771065. Retrieved 2008-09-02.

[10] Bartosz Jankowski, Wojciech Mazurczyk and Krzysztof Szczypiorski (11 May 2010). "Information Hiding Using Improper Frame Padding".

[11] Fenlin Liu; et al., B. "Real-time steganography in compressed video". MRCS, NCS, Springer, p. 6, 2006.

[12] Video Steganography for Confidential Documents: Integrity, Privacy and Version Control Diego F. de Carvalho1, Rafael Chies,Andre P. Freire, Luciana A. F. Martimiano and Rudinei Goularte