Mathematical Model that using scrambling and Message Integrity Methods in Audio Steganography

Mohammed Salem Atoum
Department of Computer Sciences
Irbid National University
Moh_atoom1979@yahoo.com

Abstract: The success of audio steganography is to ensure imperceptibility of the embedded message in stego file and withstand any form of intentional or un-intentional degradation of message (robustness). Audio steganographic that utilized LSB of audio stream to embed message gain a lot of popularity over the years in meeting the perceptual transparency, robustness and capacity. This research proposes an XLSB technique in order to circumvent the weakness observed in LSB technique. Scrambling technique is introduce in two steps; partitioning the message into blocks followed by permutation each blocks in order to confuse the contents of the message. The message is embedded in the MP3 audio sample. After extracting the message, the permutation code book is used to re-order it into its original form. Md5sum and SHA-256 are use to verify whether the message is altered or not during transmission. Experimental result shows that the XLSB performs better than LSB.

Keywords: Mathematical Model, Scrambling, Integrity

Introduction:

Information is share globally through the Internet, in digital form (Fricker and Rand, 2002). There are issues and challenges regarding the security of information in transit from senders to receivers. The major issue is the protection of digital data against any form intrusion, penetration, and theft. The major challenge is developing a solution to protect information and ensure their security during transmission (Feruza and Kim, 2007). Three components of information security are confidentially, integrity, and availability (Feruza and Kim, 2007). . Confidentiality ensures that information is kept secret from any unauthorized access. This could be done through information hiding techniques, namely cryptography and steganography (Lenti, 2000).

Cryptography involves the act of encryption and decryption of a digital data. The major weaknesses of such techniques are that even though the message has been encrypted, it still exists. Steganography dwells on concealing any digital data in a in an innocuous digital carrier, the word steganography is derived from an old Greek word which means covered writing (Katzenbeisser and Petitcolas, 2000). Steganography has been used as a mean of concealing secret messages during ancient times (Rahim, Bhattacharjee and Aziz, 2014). It was used by Histiaeus, the tyrant of Miletus who, in 499 BC, tattooed the scalps of his slaves with a hidden message with a command for his men to attack the Persian (Ricardo et al., 1999; Huayin and Li, 2008; Emelia et al., 2008; Yu et al., 2010). The message became hidden when the slaves' hair grew back.

According to researchers, steganography can be described as a study of the means of hiding secondary information within primary information without affect the size of information nor the cause of any form of distortion which could be perceived (Francia et al., 2006; Qiao, Sung and Liu, 2009b; Ganeshkumar and Koggalage, 2009; Petrovic and Yann, 2009; Liu, Qiao and Sung, 2009; Jangra, and Singh, 2014). The primary information, known as the carrier or host, was embedded within the secondary information, which is typically hidden and could be in the form of a file or message. The media with the embedded information is called stego
signal, file, bit stream or sequence (Basu and Bhoumik, 2010; Khairullah et al., 2009; Alla, Parsad and Siva, 2009; Chander, Debnath and Ghosh, 2009; Qi, Ye and Liu, 2009; Farouk, 2014).

Steganography is one of two techniques used in covert communication. However, watermarking is the second technique that can be embedded watermark into host cover to keep copyright for the hosts. Steganography typically establishes point-to-point data security (Mandala, Kotagiri and Kapala, 2013). The strength of steganographic technique in keeping the data in the carrier medium against attacks or alteration is weak during transmission, storage or format conversion is weak (Katzenbeisser and Petitcolas, 2000).

The process of embedding information in host media in steganography technique and watermarking are usually done transparently (Manimegalai et al., 2014; Koziel, 2014). The difference between steganography and watermarking is that while steganography is a technique which hides the information, watermarking actually allows the third person to see the message (Cvejic and Sebbanen, 2004; Neeta, Snehal and Jacobs, 2006). Thus, in terms of watermarking, the process needs to ensure robustness so that any intentional attacks would not compromise, remove, or cause destruction of the information in any way in the marked media while at the same time preserving the quality of the signal (Scagliola, Berez and Guccione, 2009; Bhattacharyya and Sanyal, 2010). Watermarking is the most suitable technique in cases where knowledge of the hidden information could cause possible manipulations (Mitchell, 2003; Avcibas, Memon and Sankur, 2003; Yusnita and Othman, 2007; Naji et al., 2009).

**Proposed Scheme**

This paper describes in detail the design of the proposed scheme, which is developed to enhance the security and robustness of the basic steganography model. The proposed scheme comprises of three phases: scrambling message, embedding and extraction algorithm and message integrity. Scrambling message and message integrity are the two phases added to the basic model of steganography in order to improve security and robustness. In addition XLSB is presented, this is scheme is the main part of the contribution of this research, because it extends the basic LSB technique, through incorporating high security level for the secret message before embedding and message integrity method after extraction. Figure 1 shows the proposed scheme.
The model is presented in three different parts, which consist of the basic model of steganography, scrambling of message and message integrity respectively. The following variables were involved in the interrelationships between each section of the model. Thus, \( c \) represent the cover file, and \( C \) is the set of all possible cover files for which they belong to \( c \), therefore mathematically it can be represented as \( c \in C \). Before embedding, there is a need to prepare the secret message, thus if \( m \) represent the secret message, and \( M \) represents the set of all possible secret messages for which they belong to \( m \), mathematically it can be represented as \( m \in M \). The length of the cover file is given by \( L_c = \text{length of } c \), whereas the length of the secret message is given by \( L_m = \text{length of } m \). After preparing the dataset, the scrambling message \( M_s \), which is one of the most important aspect of the general models start with partitioning process \( P_a \), where \( n_p \) represent the number of partitions within \( n_p = \text{ceil}(L_c/L_m) \), for each dataset, \( P_a \) is the Partition function of the message \( m \), which produces \( M_b = \text{message blocks after partitions } M_b = \{ m_1, m_2, m_3, \ldots, m_{np} \} \), now the \( M_b = P_a(m) \) undergoes permutation step, where \( P_r \) is the Permutation function of the message block. \( P \) is the codebook of permutation, it is a random equation. Thus a \( M_s \) is generated which is the secret message after scrambling. The message scrambled after permuted \( M_b \) are presented as \( M_s = P_r(M_b,P) \). The embedding of the of the secret message on the cover file, dwells on the \( F_e \) denoted as the Embedding Process, for which \( F_e(M_s,C) = SO \), where \( SO \) is the Stego Object. After embedding the extraction process \( F_x \), for which \( F_x(SO) = M_s \) and \( C \) will be carried out, thereafter, the extraction function \( F_x(SO,P) = M_b \) makes it possible for the extraction of the secret message. The extracted message undergoes a process of validation, where \( V_1 \) represent the validation value one, thus the validation procedure is presented below:

\[
\text{Md5sum} = \text{Checksum function using MD5 method} \\
\text{Md5sum}(M_b) = V_1 \\
\text{SHA-256} = \text{Secure hashing algorithm 256 method} \\
\text{SHA-256}(V_1) = V \\
V: \text{Validation value to check message integrity.} \\
M_t: \text{Message integrity function.} \\
V_i: \text{validation value generated from end user.} \\
\text{Md5sum}(M_b) = V_i' \\
\text{SHA-256}(V_i') = V' \\
V': \text{Validation value generated from end user.} \\
\text{If } M_t(V') = V, \text{ then the message is intact,} \\
\text{Else the message is being altered}
\]

In general the model explores the framework of the research and the steps requires to for the conceptualization of the different sections of the model. By implementation, consider \( P \) and \( V \) which are secret information required to send to the receiver through a secure channel. If message scrambling comprises of two processes: the partition process \( (P_a) \), which partition \( M \) into message blocks \( (M_b) \); and the permutation process \( (P_r) \) which re-orders \( M_b \) to construct \( M_s \), and generate the first part of secret information, \( P \).

A new scheme was developed based on the basic steganography model, which was described in detail in section 2.2 of chapter two. The basic steganography model utilizes two processes, namely embedding \( (F_e) \), and extraction \( (F_x) \). In \( F_e \) both \( M \) and \( C \) are converted into bits stream, and one of the known embedding algorithms (XLSB) is used to embed \( M \) into \( C \). \( F_e \) includes the same steps in \( F_x \) to extract the message but in an inverse way.
Scrambling Method

Algorithm 1 Scrambling Algorithm

1: Inputs: C is cover, M is message, Lc is length of C, Lm is length of M,
2: Outputs: Ms, P
3: Parameters: np is a number of partitions, Mb is message blocks, S is the size of blocks, P is P_Codebook.
4: // Partition process
5: np= ceil(Lc\Lm)
6: S= Lm/ np
7: for i=1 to np
8: for j=1 to S
9: Mb[i]= M[j]
10: End for
11: End for
12: // Permutation process
13: RandomPerm(Mb[i], j)
14: For i= 1 to np
15: choose j uniformly at random from [i,...,n]
16: // for example can choose j= ((np-i)+1) or any P= j
17: swap Mb [i] and Mb [j]
18: Ms[i]= Mb[i]
19: End for

Message Integrity Method

Algorithm 2. Message integrity algorithm

1: Input: Mb is message blocks.
2: Outputs: True or False
3: Parameters: Md5sum is function to generate checksum, SHA-256 is a function to generate verification value, V1 is first verification value, V is the second verification value.
4: // Md5sum process
5: For i=1 to np
6: V1= Md5sum( Mb(i))
7: End for
8: // SHA-256 process
9: V= SHA-256(V1)
10: Validatation Check
11: If V=V' then T else F
12: End

Conclusion

This research has explored and reviewed MP3 audio steganography, particularly with respect to MP3 files after compression. LSB in time domain has been extended and XLSB algorithm is formed. The new technique aims at meeting the three most important audio steganography requirements, which are imperceptibility, capacity, and robustness. Any technique tries to enhance the capacity or robustness should preserve imperceptibility. This research increased the capacity and robustness as well as improved the imperceptibility. Two algorithms, standard least significant bit (SLSB) and extended least significant bit algorithm (XLSB) were implemented; the first algorithm was implemented based on the general idea of LSB to
be a benchmark for the new extended algorithm. SLSB has sufficient information about cover format, and avoids manipulating those bits in samples, which cause larger error or distortion. Second approach was a proposed algorithm based on the concept secure least significant bit. XSLB has been proposed, implemented and tested.

References


