VOL. 10, NO. 23, DECEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

# ENHANCEMENT OF CONFIDENTIALITY AND INTEGRITY DURING BIG DATA TRANSMISSION USING A HYBRID TECHNIQUE

Shiladitya Bhattacharjee, Lukman Bin Ab. Rahim and Izzatdin B. A. Aziz
Computer and Information Science Department, Universiti Teknologi PETRONAS, Seri Iskandar, Perak, Malaysia
E-Mail: shiladityaju@gmail.com

**ABSTRACT**

The most fundamental issues of any data transmission over the internet are maintaining its confidentiality and integrity. The data integrity may suffer due to unauthentic interferences or various transmission errors. These issues increase when the transmitted file size is extremely large. Many researches have been performed to address these issues individually. However, there is no integrated technique being suggested in previous researches to address both these issues in big data transmission system. Therefore, we have proposed a new error control technique and a unique LSB (Least Significant Bit) based audio steganography technique and combined them to develop a hybrid technique. The proposed control technique is designed to remove all discrete or continuous data errors and to provide a backup system for accidental data loss. The proposed steganography technique is developed to offer high confidentiality, protect various security attacks and to enhance robustness against various errors. The result section shows its capacity to produce robustness against various errors in terms of signal to noise ratio, uncorrectable error rate and percentage of data loss. The confidentiality level is shown by calculating frequency and amplitude difference between the original and stego samples. Its capacity to protect various attacks has been tested by calculating entropy values. All tests are performed in wireless environment using different types and different sizes of input files.

**Keywords:** confidentiality, Integrity, transmission errors, data loss.

## INTRODUCTION

In this modern age of high speed communication, the transmission of data over the internet is very common phenomena. In the meantime, data confidentiality may suffer during the transmission due to hardware or software limitations, various transmission errors, and due to various security attacks by illegal users. Data integrity also suffers due to data loss. These issues are becoming more critical with the increment of transmitted data sizes.

In big data transmission systems, data bit errors play a very important role during transportation of any huge amount of data. Data loss can occur due to limitation of communication channel and transmission hardware or software. Adverse effects of various channel noise and environmental noise hamper data integrity [6]. Usually data link layer takes care of such errors [8]. However, it is not efficient to remove all errors and the remaining errors may cause complete or partial data loss in big data transmission [1-2]. Hence, many error control techniques have been invented to control such data errors. Among them Parity bit checking, Checksum method are used for error detection and calculation Hamming Distance, cyclic redundancy checking are used for correction errors. If $r$ is the redundancy bit and $D$ is the number of information bits, then redundancy bit can be generated by,

$$2^r = D + r + 1 \quad (1)$$

Generally the most popular used error control techniques in data link layer are Automatic Repeat Request, Forward Error Correction Code, and Low Density Parity Checking [4]. Hash function is also used for multiple bits error control. If h maps an input x of arbitrary finite bit length to an output h(x) of fixed bit length then,

$$h : \{0,1\}^* \rightarrow \{0,1\}^n \quad (2)$$

Data confidentiality is another aspect of big data transmission system. A transmission system is considered as trustful when it maintains high confidentiality level during transportation. In big data application such as transportation of financial information for any banking system, research data, governmental data and others, the confidentiality level suffers due to various security attacks whether they are manmade or machine made [14]. Sometimes, the data confidentiality liquefies due to illicit application of viruses, worms or adverse effect of different transmission errors and channel noises. Cryptography and steganography are widely used to solve confidentiality issues [8]. However, few of them are too simple that they cannot produce adequate confidentiality level and others are too complex to implement. Due to high complexity, these techniques are not applicable for big data transmission system [11].

Therefore to solve these issues in an integrated way, we have developed a hybrid technique which will

- Offer high level confidentiality for big data transmission by protecting data from various security attack and adverse effect of viruses, worms and various transmission errors.
- Produce adequate integrity level by detecting and correcting data errors efficiently during transportation of huge size of data and reduce data loss by

minimizing transmission errors, effect of various channel noises and security attacks.
- Increase processing speed to reduce the time complexity, specifically time delay during the transportation.

The following sections are including Literature Review to find the current research gap, Proposed Technique which present a new integrated technique to cover the current research gap, Assessment Platform to discuss about the experimental setup and required parameters for analyzing the performances of proposed integrated technique, Result Analysis to show the efficiency of proposed technique over the existing, Conclusion and Future work to summarize our research work in terms of its strengths and limitation and to suggest the required modification to improve it.

## LITERATURE REVIEW

Here is the list of nomenclature which will be used throughout all the text.

**Nomenclature:**

| | |
|---|---|
| $\sum$ | Denotes Finite set of all numbers and characters |
| $\phi$ | Denotes Null set |
| $\lfloor\ \rfloor$ | Denotes Floor Function |
| $\oplus$ | Denotes XOR Operation |
| *modulo* | Denotes reminder division |
| $\langle\ \rangle$ | Denotes the elements sequence of a set |

According to the Introduction section, data integrity and confidentiality are two most important and fundamental aspects of any big data transmission technique. According to the definition, data integrity is the property of maintaining originality of transmitted information having without modification or distortion, whereas, confidentiality can be maintained by converting the original information into some cryptic text or hiding the original information into any cover media such as image, video, audio or text.

From the last few decades, various researches have been conducted to control the different kinds of burst errors, which can be of either multiple bits or single bits. The nature of such errors is either discrete or continuous. In [3], the author proposed a modified decoding algorithms for Difference Set Codes to detect and correct data errors when the numbers of correctable bit errors are exceeded from one. This technique combines error detection and correction capability in a modified decoder, which makes the proposed scheme suitable for memory applications. However, it cannot detect multiple bit errors at the same time and requires multiple iterations to detect erroneous bit, which makes the process slow. In [4], the author proposed a new technique called Quick Error Detection (QED), which transforms existing post- silicon validation tests into new validation tests that significantly reduce error detection latency. QED transformations allow flexible tradeoffs between error detection latency, coverage, and complexity. It can also be implemented in software level with little or no hardware changes. QED tests also improve coverage by detecting errors that escape the original non-QED tests. . However it imposes high data overhead and requires high execution time.

Among the various existing techniques of correcting error bits in transmitted data stream, in [7], the author proposed a graphical model of linear finite-state machines (LFSM) for cyclic code based on zero and a unit cycle. It defines the correcting capabilities and it provides uniform approach to correct various errors. However, redundancy of same code requires larger time for the execution and it is too complex to build using logic gates. In [8] the author proposed a scalable multiple description coding (MDC) scheme and the forward error correction (FEC) coding using Raptor code for the scalable extension of H.264/AVC (SVC). An efficient Forward Error Correction Code (FEC) scheme for WSN's has been developed by [9] to avoid retransmission, which saves not only energy but it extends its functionalities to handle burst errors. The FEC scheme is effective for correcting burst errors when the input string lengths are even 8 bits. Here, a common channel coding standard is developed for transmitting data using nodes of Wireless Sensor networks. It has a wide range of applications such as in weather and environmental monitoring, healthcare, positioning and tracking, spacecraft and ground system communications. In order to further facilitate partial error recovery in short FEC blocks, we have studied the use of sparse RS generator matrices. These codes can easily be extended to provide unequal error protection where they permit flexible adjustment of different proportional priority levels even when short FEC blocks are used but it offers high complexity for error computation and high delay.

According to the concept of cryptography, encryption is the process of converting plain text into cryptic text or cipher text. Commonly, encryption of input data is done using a certain length of binary key. Based on the key types, cryptography can be categories into public and private. In public key cryptography, two different keys are used for encrypting the original information and decrypting the original information from the cryptic text. In public key cryptography, different keys are used for encryption and decryption. According to [11], if the separate encryption key is $K_1$ and the decryption key is $K_2$, then the cipher text ($C$) is generated from the input message ($M$) by applying encryption technique ($E$) and reversely the $M$ is generated from $C$ by applying decryption ($D$) as shown in Equations (3) and (4).

$$\mathbf{E}_{\mathbf{K}_1}(\mathbf{M}) = \mathbf{C} \quad (3)$$

$$\mathbf{D}_{\mathbf{K}_2}(\mathbf{C}) = \mathbf{M} \quad (4)$$

In private key cryptographies, the secret keys are needed to be sent to the receiving end. In such cryptography, if the secret key is denoted by *K*, where *K* belongs to the certain range of finite string values. The possible range of key value is known as key space. In symmetric key cryptography, cipher text (*C*) is generated by the encryption technique (*E*) from the input message (*M*) and reversely the *M* is generated from *C* by applying decryption (*D*) as shown in Equations (5) and (6).

$$E_K(M) = C \quad (5)$$

$$D_K(E) = M \quad (6)$$

In steganography technique, the original information is embedded within the cover media. Depending upon the cover media, the steganography can be categorized into text, audio, video and image steganography. Various researches have invented several techniques to solve the confidentiality issues by using different available security techniques separately or in combination with their original form or after partial modification of them. According to [12-13], if the cover media is *C*, secret message is *M*, stego function is *Fe* and inverse stego function is $Fe^{-1}$, optional stego key is *K*, and if stego function operates over cover media to embed secret message using secret key to produce stego message, then relation among these parameters and stego functions is shown by following Figure-1.

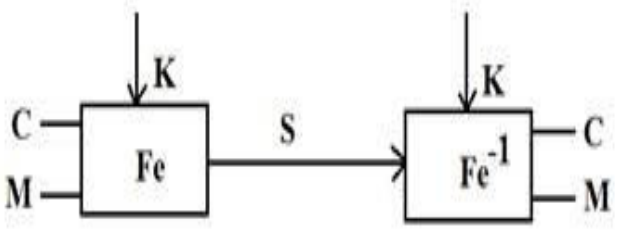


**Figure-1.** Steganographic Operation.

Among the various existing recently used cryptography techniques, in [19], the author proposed a new solution that treats data confidentiality problem by exploiting a very important ad hoc network characteristic, which is the existence of multiple paths between nodes. In [14], the author proposed an improved H.264/AVC comprehensive video encryption scheme. Here a novel hierarchical key generation method is likewise proposed. Here, the encryption keys are generated based on the cryptographic hash function. Generated frame keys are consistent with the corresponding frame serial numbers. It can ensure frame synchronization in the decrypting process when frame loss occurs. In [14], the author proposed a new framework of combinational domain encryption that encrypts significant data in spatial domain and insignificant data in wavelet domain. It offers significant reduction in computational time without compromising the security. It delegates mostly overhead to the cloud servers at the time of computation.

The existing steganography techniques have also been modified according to the requirements of the transmission system. In [15], the author proposed a low-complexity data hiding algorithms with graph-based parity check (GBPC) for gray-scale images. Based on it, two low complexity GBPC based data hiding algorithms are also proposed. However, it offers the low hiding capacity and it is not robust against the errors. This algorithm improves the security and the quality of the stego image and is better in comparison with other existing algorithms. However it offers low hiding capacity and it is inefficient to offer adequate data integrity. In [16], the author proposed an optical crypto technique with adaptive steganography (AS) for audio/video sequence encryption and decryption. This audio/video crypto-technique is based on the AS data embedding technique, the double random phase encoding algorithm, and the asymmetric encryption method. This audio/video crypto-technique offer data hiding characteristics of content dependence, less distortion, and more security. However, it involves high Incorporation time. In [17], the author proposed an approach for text steganography that uses reflection symmetry of the English alphabets. It checks the vertical and horizontal reflection symmetry properties of the characters, presented in each sentence of the text. In [18], the author proposed a text steganography method that employs a lossless compression. As this technique is language specific, it can be applied to any language by reconstituting the text database. It does not produce noise during hiding the information with in the cover media but it is inefficient to produce adequate robustness against the various channel noises.

Therefore, from the above discussion, we can see that there are various limitations exist among the various preexisting error control, steganography and cryptography techniques. Among the various existing error control techniques, few can control a limited numbers of errors and others are not suitable for big data as they offer high time and space complexities. Therefore, further research is required to develop an integrated techniques which can control all kind of data errors, occurred during the big data transmission and offer adequate confidentiality.

## PROPOSED TECHNIQUE

In the literature review we have seen that there are several issues exist in big data transmission system which hampers the confidentiality level and the integrity level of any data transmission system. Therefore, to fulfill the research gap we have proposed a hybrid technique which includes a new error control technique to remove all kinds single or multiple burst error and it also includes a new least significant bit (LSB) based audio steganography to enhance the confidentiality level of the transmission system during big data transportation. Functionalities of proposed hybrid technique have been shown by taking input text file as an example by the following Figure-2.

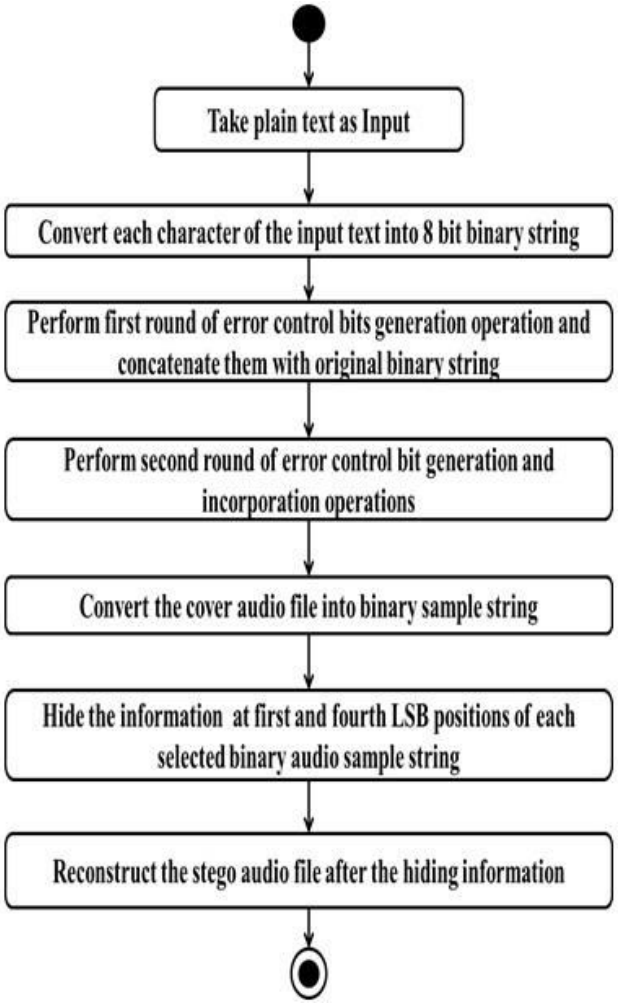


**Figure-2.** Proposed Hybrid Model

In the above Figure-2 we can see that there are several steps involve in error control bit generation and hiding the original information within cover audio file after incorporation of error control bits after dual round of error control bit generation operation. Each step of error control bit generations, their incorporation with the original string and the generation of stego audio file by hiding the concatenated string into the cover file are vividly depicted in the following sub section.

**Generation and Concatenation of Error Control Bits:**

To generate the error control bits in the first round, we need to convert each character of the input text ($Text_m$) into 8 bit binary string and store each binary string into string array $STR_m$. Here, $m$ is the total numbers of characters belong to the input text. Concatenate the two consecutive 8 bit strings taken from $STR_m$ into 16 bit binary string and store each of them into $Str_n$ where $m \leq (2 \times n)$. The Error control bit generation and its incorporation with the each character string of input text is shown by the following steps.

1. Create the first round of error control code and incorporate them with each input character.

$$\left.\begin{aligned} Str_n &= Text_m \times (10)^{l(Text_{(m+1)})} + Text_{(m+1)} \\ Err_n &= Text_{mj} \oplus Text_{mj} \\ Err'_n &= Str_n \\ Err'_n &= Err'_n \times (10)^{l(Err_n)} + Err_n \end{aligned}\right\} \quad (7)$$

$$where,\ 0 \leq j \leq 7,\ m \leq (2 \times n)\ \ and$$

$$l(Text_{(m+1)}) = \left( \left\lfloor log \begin{matrix} Text_{(m+1)} \\ 10 \end{matrix} \right\rfloor + 1 \right)$$

In Equation (7), the first part concatenates the binary string of two continuous characters of input text into *Str* string. Therefore total number of *Str* will be $n$ where $n \leq (m \div 2)$. The second steps generate the error control bits by performing XOR operation among each bit of these binary strings generated from the both continuous characters of input text. Finally the error control bits are concatenated with each *STR* to accomplish the first round of error control operation to create corresponding $Err'_n$. This step offers a back up system for accidental data loss.

2. In the Second round of error control operation each bit of *Str* is protected by the corresponding error control bits. The second round of error control operation is performed as,
3.

$$\left.\begin{aligned} x_k &= Err'_{ni} \oplus Err'_{n(i+1)} \\ ErrCon_n &= Err'_{n0} \\ ErrCon_n &= ((ErrCon_n \times 10) + Err'_{n(i+1)}) \times 10) + x_k \end{aligned}\right\} \quad (8)$$

$$where,\ 0 \leq i \leq 15\ \ and\ \ 0 \leq k \leq 14$$

In Equation (8), the XOR operations are performed between each two continuous bits of each *Err'* and then they are concatenated after these two continuous bits. This round adds the error control bits to address each individual data bits. In the result analysis section, the concept of proposed error control technique has been justified in terms of its performances.

**Integration of Final Integrated String into Audio File:**

We have used WAV audio file to incorporate the final integrated string. This process involves four major steps such as, Binarization of Audio File, Selection of Audio samples, hiding of integrated string into selected audio strings, and Reformation of Stego Audio file. All of these steps are depicted in the following sub section.

*Binarization of input Audio File*

Conversions of input audio (.wav) file into binary strings are done by breaking the audio file into several small audio samples using sampling theorem. It (sampling theorem) sets a threshold value depending upon the each of the peak amplitude values. The normalization is performed with these newly generated audio samples to calculate the peak amplitude values of each audio sample

in the range of -255 to 254. These normalized values are then multiplied with 10 to make these values in integer form. After converting these values into integer, they are again converted into 8 bit binary strings.

*Selection of Binary Audio Strings*

The incorporation of final integrated string within the cover audio strings has been done by performing the insignificant changes in the audio sample strings. The integration of such incorporated string is done by selecting some particular sample strings based on their prime position values. The audio samples are stored into string array $\langle A_n \rangle$ where $n$ represents the position value of each samples. All prime values of $n$ have been collected into an integer array $\langle A_m \rangle$ and the position value of each $A_m$ has been collected into an integer array $\langle P_m \rangle$. The position final audio strings in $\langle A_n \rangle$ for hiding the final incorporated string have been selected by the following Equation (9),

$$P'_m = A_m + P_m \quad (9)$$

Where, $0 \le m < n$

Hiding of integrated string into selected audio strings

The final integrated bit strings are incorporate at the first and forth least significant bit (LSB) positions of the selected audio samples. The embedding operations are done by taking each two consecutive bits form the final integrated strings and incorporated into the selected audio sample strings by performing some minor changes. To perform the embedding operation, we take the final integrated string array *ErrCon*, the integer array $P'_m$, which contains the position values for selecting audio samples and string array $A_n$, which contains the all audio string after the Binarization operation are taken as the input. The detail steps of incorporation such integrated strings into the selected audio samples are shown by the following algorithm -1.

**Algorithm-1: Hiding technique**

(B1) Select the sample according to the position values $\langle P_m \rangle$ from $\langle A_n \rangle$ and denotes it as *Samp* and the bit sequence of *S* is represented by $\langle S_k \rangle$ where $0 \le k \le 8$.

(B2) When the fourth bit of audio sample string is modified during incorporation from 0 to 1 then,

*if*$((S_2 = 1 \text{ and } S_4 = 1) OR (S_2 = 1 \text{ and } S_4 = 0))$
  *for*$(\text{int } i = 0 \text{ to } 2)\ S_i = 0;$
  *End for*
*End if*
*if*$(S_2 = 0 \text{ and } S_4 = 1)\ S_4 = 0;$
  *for*$(\text{int } i = 0 \text{ to } 2)\ S_i = 1;$
  *End for*
*End if*
*if*$(S_2 = 0 \text{ and } S_4 = 0)$
  *for*$(\text{int } i = 0 \text{ to } 2)\ S_i = 1;$

**Algorithm-1: Hiding technique (Contd…)**

*End for*
*for*$(\text{int } i = 4 \text{ to } 7)$
  *if*$(S_i = 1)$ *then* $S_i = 0;$
  *break;*
*End if*
*Else* $S_i = 1;$
*End for*
*End if*

(B3) When the fourth bit has been modified from 1 to 0 during data incorporation then,

*if*$((S_2 = 0 \text{ and } S_4 = 1) OR (S_2 = 0 \text{ and } S_4 = 0))$
  *for*$(\text{int } i = 0 \text{ to } 2)\ S_i = 0;$
  *End for*
*End if*
*if*$(S_2 = 1 \text{ and } S_4 = 0)\ \ S_4 = 1;$
  *for*$(\text{int } i = 0 \text{ to } 2)\ S_i = 0;$
  *End for*
*End if*
*if*$(S_2 = 1 \text{ and } S_4 = 1)$
  *for*$(\text{int } i = 0 \text{ to } 2)\ S_i = 0;$
  *End for*
  *for*$(\text{int } i = 4 \text{ to } 7)$
  *if*$(S_i = 0)$ *then* $S_i = 1;$
  *break;*
*End if*
*Else* $S_i = 0;$
*End for*
*End if*

(B4) When fourth lest significant bit (LSB) is similar to original bit, then no modification is required.

(B5) Change the first LSB bits according to the first LSB value of the data bit to be incorporated.

In Algorithm-1, step (B1) describes about the string variable *S*. The Step (B2) describes the required changes when the fourth bit of selected sample has been changed from 0 to1. Step (B3) includes the description about the incorporation of data bits at the fourth LSB position of selected samples. It describes the required changes when the value of fourth LSB value changes from 1 to 0. Step (B4) describes about the required changes when the fourth LSB bits is similar to the data bits to be incorporated. Step (B5) describes the required changes when the data bits needs to be incorporated at the first LSB position of selected each audio sample.

*Reconstruction of Stego Audio File*

After the embedding operation, the stego string array (*A*) is taken as input to perform the conversion of string array to audio file in the same format as the cover audio file. Initially the decimal value of each string element of *A* is converted into decimal value and

store each converted value into the integer array *AS*. The entire decimal values of *AS* are then divided by 10 to make them fractional. These fractional values are then stored into a double array *AS′*. The denormalization operations are performed with each element of *AS′* to get the stego audio sample by depending upon the predefined threshold values. All the stego audio samples are then concatenated to build the final stego audio file. In the result analysis section, the concept of proposed audio steganography has been justified in terms of its performances in different aspects.

**Retrieval of Original Data form the Stego Audio File**

Retrieval of integrated string from the received stego audio files requires resampling, which can be performed with the help of sampling theory depending upon the amplitude value. After the sampling process, the normalization is done with each sample. The normalization technique is performed for each sample, with the help of a predefined threshold value. The normalized values are then multiplied by 10 to convert their value to integer in the range of -255 to 254. The integer values of the audio samples are then converted into 8 bit binary strings. The appropriate sample strings are needed to select for extracting the integrated bit stream. All integrated bits are then combined into a single string.

In the next stage, crate the final error control array by separating each 48 bits string from integrated string. Initially we take the first element of error control array and checked that any bit is modified or not. Each odd bit string from the third bit in the taken string considered as the XOR bits for error control. XOR operations are done between first, second bits and every two even bits. If the result bits of such XOR operations match with the corresponding previously incorporated XOR bit, then keep these input two bits as same as they were. If it does not match, then check the pervious and next bit sequence to detect which bit is corrupted. Afterwards change the detected corrupt bit accordingly. Afterwards, we eliminate all the XOR bits from the input character string. The same operations are repeated with all the elements, i.e. character string of error control array. Thus the first round of error checking is accomplished. The two continuous 8 bits original strings and 8 bits XOR string are then separated from each input each element of error control array in second round of error control operation. Other dual sets of dual 8 bits continuous original strings are generated by performing the XOR operation between the XOR string and the separated two sets of Original strings. After generation of two sets of continuous original string, convert them into character. After then match them to get the actual pair of original character. Finally, all retived characters are concatenated into single string to build the original text file.

**ASSESSMENT PLATFORM**

In this section, few important parameter are defined which will be used in the result analysis section to analyze the efficiencies of proposed technique in different aspects. Apart from that, this section also includes the required experimental setup to perform the different experiment related to our proposed techniques to justify its efficiencies.

**Experimental Setup**

To perform our experiments we have used core java a programming language. JDK 7.0 is used to run all the java codes. We have used UBUNTU 12.04 LTS as operating system. Though all the experiments have been done in LINUX environment but these codes can be easily run with another environment like Windows or MAC. Only JDK needs to be installed in these operating systems. The concept of parallel processing is used here to make the overall process faster. Here we have used Java Thread to apply such parallel processing concept. The HPCC Cloud, provided by the Universiti Teknologi PETRONAS, is used to store the input and output files. 32 GB DDR3 RAM and Intel® Core™ i8 Processors are used as hardware parts. All files are transferred during the experimental work in wireless environment. We have tested the performances of proposed techniques with various types of input file up to the 1TB of sizes. WAV audio files are used as cover audio files to implement the proposed audio steganography.

**Definition of Some Important Parameters**

*Entropy*

In information theory, *Entropy* is used to measure the uncertainty associated with a random variable. In terms of Cryptography, *Entropy* must be supplied by the cipher for injection into the plaintext of a message so that it can neutralize the amount of structure that is present in the unsecure data. Based on Shannon's theory, to calculate *Entropy H(S)* of a source *S*, we have,

$$H(S) = \sum_{i=0}^{2^N-1} P(S_i) \log_2 \left( \frac{1}{P(S_i)} \right) \quad (10)$$

In Equation (10), $P(S_i)$ is the probability of symbol $S_i$. The ideal *Entropy* value for an encrypted message should be 8.

**Frequency Difference (FD)**

Frequency represents the number of waves that pass a fixed place in a given amount of time. F*requency Difference* signifies the difference of such wave numbers belong to the output and output samples. If f is the *Frequency Difference* can be measured in hertz (Hz) and w is the wavelength as measured in meters, then,

$$F_D = F_S - F_A \quad (11)$$

Here, $F_D$ denotes *Frequency Difference*, $F_S$ is frequency of output sample and $F_A$ is frequency of input sample. *Frequency Differences* between original audio samples

and to stego audio samples to show similarity between the original and stego samples.

**Amplitude Difference (AD):**

Amplitude of any audio samples determines its strength of producing sound. In other words it is the size of vibration which determines how loud the sound is. So, higher amplitude produces louder sound or vice versa. If any sound wave travels *D* distance with *F* frequencies, then the amplitude (*A*) can be calculated as,

$$A = \frac{D}{F} \quad (12)$$

*Amplitude difference* (*AD*) refers to the deference of absolute amplitude value between the original sample and the stego sample. The *Amplitude Difference* (*AD*) can be formulated as,

$$AD = \left| Stego\ Amplitude - Actual\ Amplitude \right| \quad (13)$$

Any steganographic technique should have minimum *AD*. If it produces high *AD*, then the steganographic technique is said to be less efficient to reduce perceptual differences between the original and stego audio samples.

**Information loss (IL)**

Information loss is the certain amount of transmitted data which cannot retrieved at the receiving end due to various unwanted circumstances such as it may corrupted due to various limitations of transmission system, due to various security attacks and so on. It can be formulated as,

$$IL = \frac{Actual\ file\ size - Retrived\ file\ size}{Actual\ file\ size} \times 100 \quad (14)$$

**Uncorrectable Error Rate (UER):**

A certain amount of data error cannot be corrected and remains the same as its original form after various error control operations are performed. Uncorrectable error rate (UER) is used to calculate the total amount of uncorrected errors exist in a data file after applying any error correction algorithm. Uncorrectable error rate of an error correction algorithm with respect to any file size is calculated as,

$$UER = \frac{Total\ errors - Corrected\ errors}{Total\ errors} \times 100 \quad (15)$$

***Signal to Noise Ratio (SNR)***

*Signal to noise ratio* (*SNR*), produced by any error control technique or any steganography can be expressed logarithmically in decibels (dB) and formulated as:

$$SNR_{dB} = 10 \times log_{10} \left\{ \sum_n x^2(n) / \sum_n [x^2(n) - y^2(n)] \right\} \quad (16)$$

In the Equation (16), $x(n)$ is the original input sample length or the amplitude of the cover files, whereas $y(n)$ is the sample length of the output sample file or amplitude of the stego sample file. Our proposed technique incorporates a unique error checking technique and a steganography technique to enhance the robustness and confidentiality of data.

*Throughput*

*Throughput* or *TP* is the amount of work is done in a given time. It is measured to calculate the time efficiency of a certain technique. The throughput produced by any technique can be calculated as,

$$TP = \left( \frac{Output\ file\ size}{Execution\ time} \right) \quad (17)$$

**RESULTS ANALYSIS**

In this section the performances of the proposed technique has been analyzed in the basis of experimental result. According to our objectives, our experimental result has been plotted by strictly focusing on potentiality to offer confidentiality, efficiency to offer data integrity and robustness against various transmission errors and capacity to produce high processing speed during the execution. We have tested the efficiencies of various exiting techniques and our proposed techniques with our own experimental setup and the results have been compared and tabulated in the following discussion of this section.

**Confidentiality Level offers by Proposed Technique**

According to the definition section, the low *Frequency Difference (FD)* and *Amplitude Difference (AD)* between original and stego signal offers low perceptual differences. And the perceptual difference is low between the original and stego samples, facilities higher confidentiality level. *FD and AD* offered by the proposed and other existing audio technique have been calculated using Equation (11) and Equation (12). The results are tabulated in the following Table-1 and Table-2.

**Table-1.** *AD* produced by different steganography techs.

| Steganography Techniques | Frequency Differences (in Hz) | | | | | |
|---|---|---|---|---|---|---|
| | S1 | S2 | S3 | S4 | S5 | S6 |
| Existing LSB Technique | 3.27 | 3.21 | 3.32 | 3.11 | 3.27 | 3.26 |
| Parity Coding | 2.91 | 2.84 | 2.89 | 2.83 | 2.94 | 2.86 |
| Phase Coding | 2.82 | 2.83 | 2.81 | 2.87 | 2.81 | 2.85 |
| Spread Spectrum | 2.67 | 2.61 | 2.65 | 2.69 | 2.71 | 2.73 |
| Echo Hiding | 4.61 | 4.65 | 4.56 | 4.72 | 4.64 | 4.59 |
| Proposed Technique | 0.32 | 0.39 | 0.36 | 0.29 | 0.27 | 0.35 |

**Table-2.** FD produced by different steganography techs.

| Steganography Techniques | Absolute Amplitude Differences (in dB) | | | | | |
|---|---|---|---|---|---|---|
| | D1 | D2 | D3 | D4 | D5 | D6 |
| Existing LSB Technique | 1.27 | 1.34 | 2.20 | 2.10 | 1.87 | 1.66 |
| Parity Coding | 1.71 | 2.11 | 2.05 | 1.83 | 1.91 | 2.16 |
| Phase Coding | 1.05 | 1.23 | 1.11 | 1.17 | 1.21 | 1.24 |
| Spread Spectrum | 1.07 | 1.11 | 1.15 | 1.19 | 1.21 | 1.17 |
| Echo Hiding | 1.61 | 1.45 | 1.54 | 1.71 | 1.67 | 1.53 |
| Proposed Technique | 0.30 | 0.31 | 0.11 | 0.22 | 0.17 | 0.33 |

From the Table-1 and Table-2, we can see that our proposed technique offers low *AD* and *FD* than the other existing steganography techniques. A steganography technique is efficient to protect various security attacks can be determined by calculating its capacity to produce entropy values. Entropy values are calculated using Equation (10) and tabulated in the following Table-3.

**Table-3.** Entropy values are offered by security techs.

| Cryptography Techniques | Entropy Values | Steganography Techniques | Entropy Values |
|---|---|---|---|
| AES | 7.73 | Existing LSB Technique | 7.43 |
| DES | 7.53 | Parity Coding | 7.56 |
| 3-DES | 7.67 | Phase Coding | 7.61 |
| Blowfish | 7.52 | Spread Spectrum | 7.63 |
| RSA | 7.51 | Echo Hiding | 7.49 |
| Entropy Value produced by Proposed Integrated Technique is 7.77 | | | |

In Table-3, we can see that our proposed technique offers better entropy values than other existing cryptography as well as steganography techniques. There for according to the definition sub section, our proposed technique is more efficient to protect various security attacks than the existing. Therefore from the Table-1, Table-2 and Table-3, we can claim that our proposed technique is much more efficient to offer high confidentiality level in big data transmission than the other existing security techniques. Thus we achieve our first objective.

**Integrity and Robustness offer by Proposed Technique**

Robustness and data integrity can be tested with the calculation of uncorrectable error rate. According to the definition section, percentage of uncorrectable error rate determines the total number of errors remain in the information after any error control operation is performed. Therefore, we have calculated the efficiency to minimize the uncorrectable error rate by various existing error control techniques as well as the proposed technique have been calculated with the help of Equation (15). All the results are tabulated in the following Table-4.

**Table-4.** UER offered by various error control techniques.

| Error Control Technique | Input File Size in TB | | | | | | |
|---|---|---|---|---|---|---|---|
| | 0.1 | 0.3 | 0.5 | 0.6 | 0.8 | 1.0 | |
| CRC | 0.054 | 0.051 | 0.050 | 0.049 | 0.049 | 0.048 | % Of UER |
| FEC | 0.045 | 0.043 | 0.045 | 0.043 | 0.042 | 0.041 | |
| Hamming Distance | 0.038 | 0.036 | 0.035 | 0.037 | 0.035 | 0.033 | |
| ARQ | 0.029 | 0.026 | 0.027 | 0.026 | 0.027 | 0.026 | |
| Proposed Technique | 0.009 | 0.008 | 0.008 | 0.007 | 0.007 | 0.008 | |

In Table-4 we can see that our proposed technqe offers low percentage of UER among the other existing error control techniques. This fact justifies that our proposed error control technique is efficient to minimize the data errors more than the other existing error detection and correction techniques.

Another important parameters to justify the robustness against various errors offer by any error control technque is *Signal to Noise Ratio(SNR)*. According to the defition section, if any error control technique offers high *SNR,* then it reflects that the used error control technique is efficient to offer high robustness agaist the various data errors. The SNR offers by the proposed technique and the other error control technque have been calculated using different input binary file sizes using Equation (14). The results are shown in the following Figure-3.

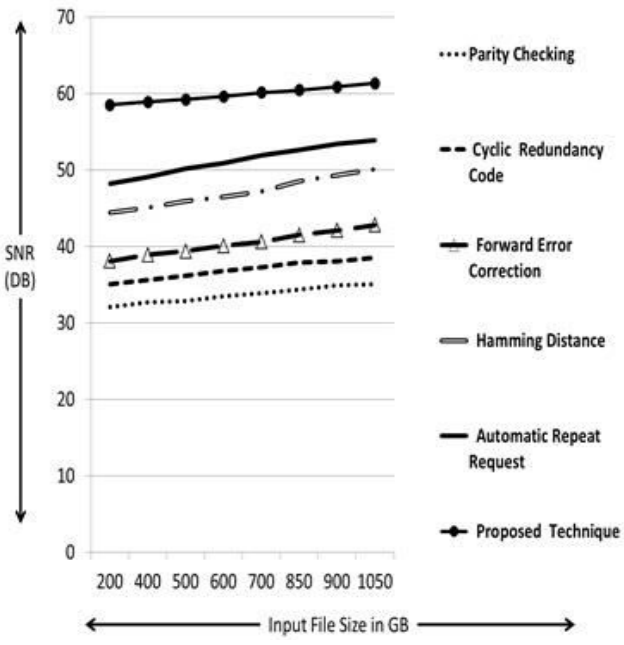


**Figure-3.** SNR offer by various error control techniques.

The data integrity offerd by the proposed technique can be tested by calculating its capacity to minimize the information loss. The information loss produced by the proposed integrated technique and the other steganography techniques are calculated with the help of Equation (14) and shown in the follwing Figure-4.

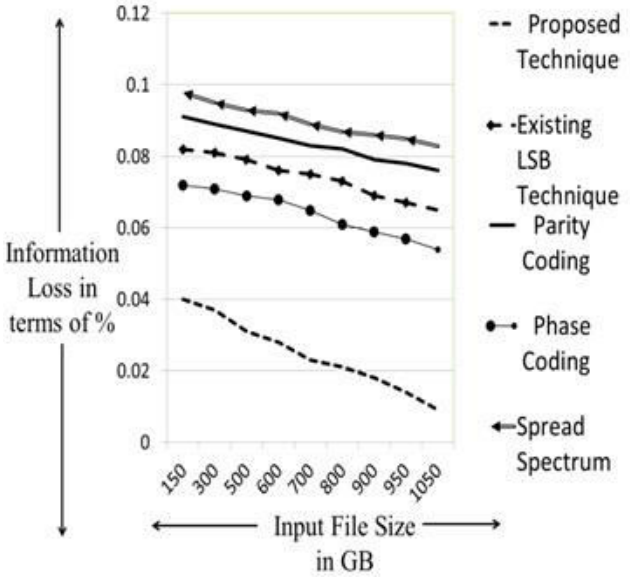


**Figure-4.** Data loss offer by various security techniques.

From the Table-4, Figure-3 and Figure-4 we can see that our proposed technique is efficient to produce higher robustness against various data errors and more efficient to reduce the data loss during big data transmission system. These facts reflect that our proposed technique offer higher integrity than the other existing techniques. Therefore, our proposed technique can fulfill our second objective.

**Processing Speed offer by Our Proposed Technique**

To reduce the time delay, increment of processing speed is an important factor. The processing speed can be measured by calculating the *Throughput* or *TP*. The *TP* produced by various error control technique and the various steganography techniques have been calculated with the help of Equation (17) and tabulated in the following Table-5 and Table-6.

**Table-5.** TP offered by different error control techniques.

| Error Control Techniques | Generation and Incorporation (MB/Sec) | Detection and/or Correction (MB/Sec) |
|---|---|---|
| CRC | 4.24 | 4.22 |
| FEC | 4.59 | 4.57 |
| Hamming Distance | 5.21 | 5.23 |
| ARQ | 3.21 | 3.20 |
| Parity Checking | 6.21 | 6.23 |
| Proposed Tech. | 6.15 | 6.13 |

**Table-6.** TP offered by different steganography.

| Steganography Techniques | Hiding Data (MB/Sec) | Retrieval of Data (MB/Sec) |
|---|---|---|
| Existing LSB Tech. | 2.54 | 2.51 |
| Parity Coding | 2.76 | 2.77 |
| Phase Coding | 1.76 | 1.75 |
| Spread Spectrum | 1.54 | 1.51 |
| Echo Hiding | 1.87 | 1.85 |
| Proposed Tech. | 3.45 | 3.47 |

From the Table-5 and Table-6, we can see that our proposed error control technique and the proposed steganography technique offer better processing speed in terms of offering higher *Throughput* than other existing various error control techniques and steganography techniques. Therefore, it can be said that our proposed error control and audio steganography technique is efficient to reduce the time delay during transmission process. Thus, the proposed technique fulfills the final objective.

## CONCLUSIONS

Among the various aspects of big data transmission, data confidentiality and integrity are the primary properties. Generally, the data confidentiality and integrity suffer due to limitation of hardware and software, various security attacks, channel and environmental noises. To resolve these issues we have designed and developed an integrated technique in which we have incorporate an error control technique to remove the limitations of data link layer by detecting and preventing maximum data errors from transmitted data. The confidentiality level and robustness of the transmission system are further increased by including a distinct LSB based audio steganography technique. We have calculated the performances of proposed technique in different aspects with the help of various measuring parameters and then it is compared with the other existing technique to prove its efficiency over them. All the data transmissions are conducted in wireless environment during the implementation stage.

In result analysis section, we can see that our proposed technique offer high signal to noise ratio (i.e. up to 60.91 dB) in all circumstances among the rest. Similarly the proposed integrated technique produces less amplitude difference between the cover and stego audio file and it offers better avalanche effect against the various security attacks. It also yields low perceptual difference between the cover and stego audio files. These justify that the proposed integrated technique is able to produce higher confidentiality level in comparison of other existing techniques. However, the proposed technique cannot control as well as protect complete data loss. Furthermore, this integrated technique is not efficient to resist all types of security attacks and data errors, as it produces up to 0.007% of uncorrectable error rate. The processing speed offer by the proposed technique is better than the existing

but it is not up to the mark. Therefore, further improvements in the proposed technique are still needed to enhance its efficiency. Another future work related to proposed integrated technique to discuss and analysis about the reliability of the transmission system which can be controlled by the proposed steganography technique.

**REFERENCES**

[1] Nagarajan A., V. Varadharajan, Tarr N. 2014. Trust enhanced distributed authorisation for web services, Journal of Computer and System Sciences, 80(5), 916-934.

[2] Haddad M., Hacid M., Laurini R. 2012. Data Integration in Presence of Authorization Policies, Trust, Security and Privacy in Computing and Communications (TrustCom), 92-99.

[3] Reviriego P., Flanagan M. F., Liu S.-F., Maestro J. A. 2012. Error-Detection Enhanced Decoding of Difference Set Codes for Memory Applications, Device and Materials Reliability, IEEE Transactions on , 12(2), 335-340.

[4] Hong T., Li Y., Sung-Boem P., Mui D., Lin D., Kaleq Z. A., Hakim N., Naeimi H., Gardner D.S., Mitra S. 2010. QED: Quick Error Detection tests for effective post-silicon validation, Test Conference (ITC), 2010 IEEE International , 1-10.

[5] Pinto P. E. D., Protti F., Szwarcfiter J. L. 2012. Exact and approximation algorithms for error-detecting even codes, Theoretical Computer Science, 440–441, 60-72, 6 July .

[6] Breitinger F., Stivaktakis G., Roussev V. 2014. Evaluating detection error trade-offs for byte wise approximate matching algorithms, Digital Investigation, 11( 2), 81-89.

[7] Semerenko P. 2009. Burst-error correction for cyclic codes, EUROCON 2009, EUROCON '09. IEEE , 1650-1655.

[8] Zhao Z., Wang L., Tao J., Chen J., Sun W., Ranjan R., Kolodziej J., Streit A., Georgakopoulos D. 2014. A security framework in G-Hadoop for big data computing across distributed Cloud data centres, Journal of Computer and System Sciences, 80(5), 994-1007.

[9] Singh M.P., Kumar P. 2012. An Efficient Forward Error Correction Scheme for Wireless Sensor Network, Procedia Technology, 737-742.

[10] Korhonen J., Frossard P. 2009. Flexible forward error correction codes with application to partial media data recovery, Signal Processing: Image Communication, 24(3), 229-242.

[11] Lee C. C., Chen H. H., Liu H.T., Chen G.W., Tsai C. S. 2014. A new visual cryptography with multi-level encoding, Journal of Visual Languages & Computing, 25(3), 243-250.

[12] Das R., Tuithung T. 2012. A novel steganography method for image based on Huffman Encoding, Emerging Trends and Applications in Computer Science (NCETACS), 2012 3rd National Conference on , 14-18.

[13] Karaman H. B., Sagiroglu S. 2012. An Application Based on Steganography, Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on , 839-843.

[14] Wang X. S.2011., Panel: research agenda for data and application security, In Proceedings of the first ACM conference on Data and application security and privacy (CODASPY '11), 283-284.

[15] Marwaha P. 2010. Visual cryptographic steganography in images, Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on, 1-6.

[16] Guizani S., Nasser N.2012. An audio/video crypto —Adaptive optical steganography technique, Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International , 1057-1062.

[17] Majumder A., Changder S. 2013. A Novel Approach for Text Steganography: Generating Text Summary Using Reflection Symmetry, Procedia Technology, 10, 112-120.

[18] Satir E., Isik H. 2012. A compression-based text steganography method, Journal of Systems and Software, 85(10), 2385-2394.

[19] Meena S., Daniel E., Vasanthi N. A. 2013. Survey on various data integrity attacks in cloud environment and the solutions, Circuits, Power and Computing Technologies (ICCPCT), International Conference on, 1076-1081.

[20] Xu H., Rui J. 2012. Attacks and improvements on Data Integrity as a Service protocols, Network Infrastructure and Digital Content (IC-NIDC), 3rd IEEE International Conference on , 242-246.