A secured compression technique based on encoding for sharing electronic patient data in slow-speed networks

Subrata Kumar Das *, Mohammad Zahidur Rahman

Dept. of Computer Science and Engineering, Jahangirnagar University, Savar, Dhaka, Bangladesh

**Abstract**

Healthcare systems capture patients’ data using different medical equipment and store it in the databases with a continual increase in data volume. The continuous processing and sharing of this massive data are rising concerns in live data transferring over the networks. Sending patient data to the distant remote user without proper compressing format requires high latency in the communication channels. Any alteration of data transmitted via the communication medium may also cause issues in ensuring data authentication and integrity. For solving the problems, watermarking method is being applied to ensure such security, which has a cheaper computational cost. Various watermarking mechanisms are available for ensuring health data security, especially for medical images. Watermarking on the text was not used yet due to the lack of efficient technique. This paper proposes a secured compression technique for patient live-text data while sharing them remotely over a bandwidth-deficient channel. To test the proposed system, we use patient data. The result indicates that the proposed strategy outperforms the existing compression methods and is robust enough to provide data integrity and authentication.

1. Introduction

Healthcare data are patient-related information containing physical assessments, for example, glucose, arterial blood pressure, blood particles ratio, etc. These data are gathered directly from the Internet of Things (IoT) and other medical equipment. The digitization of health data provides immediate support to the daily life of people. Distant experts should have the facility to process and share the health data for quality treatment and advanced services. Such digital data access can save the patient time and costs. The transmission of patient data is significant in networks having low bandwidth. This transmission becomes crucial to provide timely data to care providers during patients’ urgent treatment who become sick or are injured while traveling or who have not taken medication under their care. However, the bad-quality networks and security are significant barriers to providing those services, especially telemedicine services offered by the organizations in rural locations [1, 2, 3]. Shukla et al. [4] demonstrate the large dumping of data leads to more traffic that gets high latency and network congestion in data processing and sharing time. They also indicate that healthcare data becomes inadequate for end-users when an increasing round-trip time delay is created to transfer extensive data from source to destination. A higher error probability makes because of transmitting large data over a slow-speed network.

Additionally, the geographically distant accessing of data raises the risk concerning security. Requested data could come from an unreliable source or be altered by the attackers [5, 6]. Consultancy based on that manipulated data may destine wrong treatment and cause the severe illness of the patient [7]. Thus, remote data exchange may yield issues in ensuring data authentication and integrity. Authentication provides data source identification, while integrity guarantees faultless data communication between sites. Those security issues should be eradicated as patient data are more sensitive [8]. Therefore, a significant concern is to design a system to pass data in low latency through the communication media and assure data authentication and integrity.

Different research works have been done on patient data for reducing their size using LZW [9], LZMA with LZ77 [10, 11], Deflate/Huffman coding [12, 13, 14, 15, 16], Run-length encoding (RLE) [17], LZ4 [18, 19], LZO [18, 19], and BZ2 [11, 18]. Although these existing techniques are being used for compressing patient data, they do not provide better performance due to having fewer identical samples or multiple occurrences of repetitive substrings in data [15, 20]. In the security concern, the watermarking strategy is being applied to provide security in e-healthcare systems with a cheaper computational cost [21, 22]. Wa-
termarking is the procedure of inserting an identification message into the digital data. The embedded data is referred to as a watermark that can be a logo, text, or pattern. Digital data—also known as host data or media files—may be the image, audio, video, or text. Depending on the categories of digital data, the watermarking is classified into four types: image watermarking, audio watermarking, video watermarking, and text watermarking [23]. Different image watermarking methods are being used to secure medical images in health systems [24, 25, 26, 27, 28]. To the best of our knowledge, no text watermarking scheme is existed for patient live-text data to ensure their integrity and authentication. Watermarking on text is a highly complex task [29] and limited research is performed for documents text only [23, 30, 31]. Three text watermarking strategies are currently used for text data: (i) Structural based text watermarking, (ii) Linguistic based text watermarking, and (iii) Image based text watermarking. Structural based text watermarking is not suitable for patient live-text data because of being raw text. Most research articles on this approach demonstrate for the document (pdf, word) by shifting lines and spaces. But those shifting and formatting are not possible for raw text patient data. In image based text watermarking, a text is interpreted as an image to embed a watermark to a series of text-image [32, 33]. As a result, image based strategy does not apply to plain medical text. Moreover, linguistic-based text watermarking uses the semantic and syntactic nature of the media text to watermark them [23, 34, 35]. In this approach, grammatical alternation is applied without affecting the meaning of the text by using a synonym or changing words (nouns, pronouns, verbs, adjectives, prepositions, etc.). This text watermarking is also done by transforming the text by appending the subject and altering the sentence formation from active to passive. The problem with this technique is that the process of embedding a watermark should comply with the grammar rules to preserve the readability of the data. However, the patient’s live-text data does not follow the standard grammatical rule to insert a watermark. This method is not also resilient to random word substitution attacks and can destroy both the semantic connotation and the sensitive nature of the data. As the patient data is sensitive, any data alternation using a linguistic approach could cause patients severe illness. In this situation, ensuring data integrity and authenticating reliable source information has become a concern for the lack of suitable strategies. Therefore, a secured new compression technique requires communicating patient data in low latency via the networks.

This research proposes a new encoding technique instead of traditional ones to compress patient data to solve current problems significantly. The study also introduces a new approach for text watermarking strategy to ensure the security of live-text data while sharing remotely. The experimental results indicated that our designed method transformed data over the networks in a more highly compressed format than other existing techniques. The system could also judge the information about the source rightly and showed better accuracy against different intended attacks: transformation attack, rearrange attack, insertion attack, deletion attack.

The remaining article is structured as follows. Related research works are presented in section 2. Section 3 points out the traditional algorithms for data compression, watermarking, and their problems. The strategy of the proposed approach is detailed in section 4. Section 5 demonstrates the data processing and communication flow. The result found from the research is described in the result and discussion section 6. Finally, the article provides a conclusion in section 7 by summarizing the work.

2. Related work

Many research works are published on data compression and data security. Hanoune and Lysmos [18] cite a compression technique for an E-health gateway for monitoring applications related to the medical sector to improve data processing and collection. Their goal is to compress data volume and transfer it to the distant node for storing, but their system lacks to send data in compressed format from the server to the end-users. Ni et al. present a health monitoring system to reduce and reconstruct data depending on auto-encoder form [36]. However, they only define the data compression technique in storage time. A data reducing technique is identified by Arican and Polat only for fetal-ECG data collected from the PhysioNet database. Their approach compresses data using the variance and neighborhood compression algorithm [37]. Deepu et al. propose a paper to shrink ECG signals with lossless and lossy techniques by allowing hybrid transferal form and basing the selection of adaptive data rate [38]. Abdelatif et al. publish an article applying the traditional compression method before transmitting data by choosing an efficient network from multiple ones [39]. But, their technique does not work in reducing the volume of health data. A lossless data reduction technique is demonstrated by Tekette et al. using the first derivative and entropy encoding of ECG signal [40]. Sridhar and Lakshmi [9] conduct research using some lossless compression techniques that act efficiently to reduce medical data. They find that the LZW technique compresses the health data better than other approaches (Run-length coding and Huffman coding). However, the data reduction is dependent on the redundancy and repetition in the patient data. Cao et al. define a classification and data reduction scheme that points to data properties, accepted data, and data delivery with efficient energy [41]. A data shrinking approach is introduced for the transmission of selective data considering the detected patient’s conditions [42].

Abdmouleh et al. present a way to reduce the size of healthcare images using the Discrete Cosine Transform to decrease the accessing time of data encryption-decryption [43]. They find many strategies to lessen the image volume before warehousing it on the storage or exchanging it over communication media. Ahilan et al. define a lossless data reducing approach only on DICOM images for applications serving telemedicine, which is essential in data storing [44]. Amri et al. [45] present a scheme using standards of lossless compression to reduce the medical data space. That approach only minimizes image volume by resizing, but not to lessen the data as user requirements. Sawanah also defines a data compression mechanism for E-medical images with keeping their quality [46]. Azar et al. [47] publish a fast error-bounded lossy shrinkage method and apply it for recording medical information in the IoT before transmitting and rebuilding the transferring data at the end site. However, to our knowledge, the traditional compression methods are still used to reduce medical text before passing it to the remote end. Therefore, the first research motivation is to compress data through encoding to send the end-users over the networks.

On the other hand, the researchers are conducting research to ensure the security of patient data by basing different strategies. Aotaibi and Elfouafi [48] introduce text watermarking to improve the embedding capacity using two methods. They replace normal-space with open-space between words. Although the proposed scheme works against formatting and tampering attacks, it is permeable to retying attacks. Yingjie et al. [49] develop a zero-watermarking approach for ensuring data security. The adjectives’ proportional feature with the set of keywords and core verb is used as a watermark. Their method lacks robustness and capacity issues. Almehmadi and Gutub [50] propose an Arabic e-text watermarking using the extension character “Kashida” to secure data. Alkhafaji et al. [51] demonstrate a watermarking strategy to improve Arabic text security in the Holy Quran, which enhances the watermarking way based on a reversing technique. Wen et al. devise a two zero-watermarking scheme [52] appending extra codes with the file data based on the feature of the file. Despite having a good real-time performance, their technique has a low capacity. Khadami et al. [23] define a watermark embedding method for securing text data and perform a test in a local as well as a cloud environment. Another watermarking approach is proposed by Ali et al. [53] for healthcare data by fusing the patient identity into data through zero-watermarking. The identification message is encrypted by applying Shamir’s secret sharing technique before adding to the data. Li et al. [54] present a reversible and robust watermarking strategy by summing every continuous col-
umn and considering the group of columns having the most extensive sum. Xiao et al. [31] demonstrate a mechanism for inserting identification information with the text document, and it keeps the watermarked record perturbed from the main text. This strategy is developed for one font, and it has a low capacity issue. Hilal et al. [55] cite a watermarking on the text by taking a word mechanism (RETWNLPA) based on the Natural Language Processing. They emphasize improving the accuracy of sensory text from English in tamper detection. The RETWNLPA method allows a watermark embedding and extracting logically with the original data without affecting them.

Nagm et al. [27] propose a technique for image content protection by synthesizing an encrypted identification message from the actual data. They embed no watermark with the original data. Aljuaid and Parah [56] introduce an interpolation approach to secure the health data in two layers. The system generates the quality cover media and upholds the reversible embedding of medical data. Fares et al. [57] publish an article for ensuring patient data security exchanged in telemedicine. Malayli and Vedhanayagam [58] demonstrate a technique for reversible watermarking by measuring a watermark code from health data to transfer and authenticate them. The majority of the research article published on healthcare data taking images, but not on text data. Thus, a new text watermarking approach is urgent to develop for securing patent text data, which is our second goal.

A summary of selected paper from literature review is presented in Table 1. The table includes the recent articles on data compression, data security, and their focus and gaps.

### 3. Existing techniques and its issues

This section overviews the existing strategies for data compression, security, and its issues. The article review finds out several lossless data compression techniques: Lempel–Ziv–Welch (LZW), Run-length encoding (RLE), Lempel–Ziv–Oberhummer (LZO), Lempel–Ziv Markov chain Algorithm (LZMA), Lempel–Ziv–Storer–Szymanski (LZSS), LZ4, Deflate, BZ2. Those methods are currently being used to compress patient data. Most algorithms are designed depending on a dictionary using Lempel–Ziv (LZ) compression methods. LZ is the well-preferred technique for two main criteria: no data is lost while compressing data and the adaptability to different data formats. Data shrinking strategy using dictionary acts by creating a dictionary containing short sequences of strings referred to as phrases. The compression gain is achieved with that approach by taking shorter code words in place of variable-length strings. The compression rate mainly depends on repetitive sequences and sub-strings among the data. However, major compression strategies developed based on dictionaries are useless for small amounts of data or in the absence of repetitive sequence. For that reason, different algorithms are developed by considering the system requirements and modifying the creation of a dictionary basing the LZ algorithm. The existing algorithms used to compress patient data and their issues are presented in Table 2.

On the other side, the image watermarking approach is currently applied to ensure the security of image health data [24, 25, 26, 27, 28]. To the best of our knowledge, no existing text watermarking approach is available for patient text data to provide their authentication and integrity. Although few text watermarking techniques (linguistic-based text watermarking, structural based text watermarking) are used to ensure the security of document text, those are not suitable to secure patient text data. The traditional text watermarking procedures face some problems in embedding a watermark into patient text data that are summarized in Table 3. As the patient data is sensitive, correct data sharing is significant to maintaining data integrity and authentication. The watermarking technique into patient data considering text data usability and properties could reveal a way to solve the existing challenges and provide data security. Therefore, the paper aims to develop a strategy to meet the minimum latency to securely access and share patient data over the networks.
Table 2. Analysis of different lossless compression mechanisms used in healthcare system.

| Algorithm | Strategy | Issue                                                                 |
|-----------|----------|----------------------------------------------------------------------|
| LZW       | LZW is an improvement of LZ78 and a heuristic algorithm that tries to search increasingly longer repetitive phrases and encode them. | The algorithm allows finding a match always and is not effective in compressing data without multiple occurrences of substrings. |
| RLE       | To reduce data based on repetitive sequences containing many consecutive data known as runs. The algorithm store the sequence of data as a single value and count. | It may yield ambiguity for strings with numerical characters. It may increase data even double the size of original data for the lack of repetitive sequences. |
| LZO       | It is a block-level compression and implemented based on a 64 KB compression dictionary. LZO uses LZ77 with a small hash table. | It provides lower compression gain and wastes more space. |
| LZSS      | LZSS is a derivative of LZ77 and works by checking whether a substitution decreases data size or not. | The algorithm has poor anti-error performance and does not provide good performance to compress data. |
| LZMA      | LZMA is a dictionary compression scheme yielded modifying LZ77 at a bit rather than byte level. | The slow compression speed is the main problem of LZMA. |
| LZ4       | It compresses data based on LZ77. LZ4 represents a series of data sequences that contain a one-byte token in the starting and split into two 4-bit fields. | The algorithm is not able to perform well if identical characters are less in the data. |
| Deflate   | Deflate compression approach consists of an order of blocks analogous to successive input data blocks. That method is developed with a combination of Huffman coding and the LZ77 algorithm for compressing each block. | The central problem of Deflate is the searching way of the longer and duplicate substring. This algorithm can get a larger volume of compressed data or even shows lower performance. |
| BZ2       | This takes multiple layers encoding atop one another. The method leads to the run-length encoding of the elementary data, uses the Burrows-Wheeler scheme, then implicates the Huffman coding prior to associating various tables. | It is slower and not suitable with small data and for less repetitive sequence data. |

Table 3. Analysis of different existing text watermarking techniques.

| Algorithm                | Strategy                                                                 | Issue                                                                                                                        |
|--------------------------|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Structured based text    | This approach transforms the format or structure of the media text or selects a keyword from the text to add the watermark with it. The watermark is embedded by applying the shifting of lines and spaces of the document text. | The shifting or formatting is not possible for patient live-text data because this is raw text. This will not be robust against formatting attacks and retyping attacks. Using keyword could increase the length of the watermark highly and is not robust enough against deletion and insertion attacks. |
| text watermarking        |                                                                         |                                                                                                                              |
| Linguistic based text    | This technique uses the semantic and syntactic nature of the media text to watermark them. Grammatical alternation is applied without affecting the meaning of the text by using a synonym or changing words (nouns, pronouns, verbs, adjectives, prepositions, etc.). This text watermarking is also done by transforming the text, such as adding the subject, altering the sentence formation from active to passive, or vice-versa. | The process of embedding a watermark should comply with the grammar rules to preserve the readability of the data that do not follow patient text data. This approach is not resilient to random word substitution attacks. This procedure can destroy both the semantic connotation and the sensitive nature of the data that could cause patient illness severely. |
| text watermarking        |                                                                         |                                                                                                                              |
| Image based text         | A text is interpreted as an image into a series of text-image.          | This watermarking approach is not applicable for patient text data because of being the plain text instead of the image. |

4. Methodology

The research is conducted to transfer patient data in a more compressed format over the networks with low latency by ensuring data integrity and authentication. We lead our work in two phases: encoding the patient data to reduce their size and ensuring the security of live-text data while transmitting over the communication channels. Section 4.1 details the design strategies of the proposed solution for compressing the patient data by considering its properties. Besides, the data security mechanism is introduced in section 4.2.

4.1. Data compression process

The traditional compression methods take the data responded from the server directly and transmit them over the communication channel in compress format. Maximum algorithms for lossless data compression use a dictionary approach while encoding and decoding. Those algorithms are useful for reducing data size by taking lots of repetitive sequences or sub-string described in Section 3. But, the issue in reducing medical data is mainly the scarcity of repetitive sequences or sub-strings with them. The space and computational complexity of data compression strategies depend purely on the efficient formation of the data structure. Due to having a few identical sequences with patient data, the existing compression techniques do not ensure better performance by decreasing their size. Therefore, the paper first proposes a new solution scheme to compress patient data by making a dictionary shown in Algorithm 1 considering data properties.

The proposed technique would use a dictionary to reduce data depending on medical data structure and carry out various operations: the data conversion based on the dictionary made from different diagnostic assessment charts; to replace information by the code of a conferred pattern while getting in the dictionary. This observation inspires to development of a new method for compressing patient data with encoding shown in Algorithm 2.

The health data, for example, can consist of blood pressure, glucose, blood particle ratio, etc., is necessary to process remotely by the health practitioners to know the patients’ previous condition. The electronic data is usually transmitted to the end-users in the improper compressing format using the conventional technique. For that reason, a large amount of information is transferred to the users via communication
Algorithm 1 To create formatted data or tables supposing the prior practitioner knowledge (I₁), medical terms (I₂), diagnostic indicator charts (I₃), medication products (I₄), databases’ data (I₅),... and more information (I₆).

**Input:** Information from different sources (I₁, I₂, ... I₆); Hash map H[K].

**Output:** Dictionary (Dᵢₑᵢ) collected from the information along with the identification code.

1: Collecting different information (I₁) from the various sources related to medication.
2: Making Dᵢₑᵢ from collected data (I₁) using Hash Map.
3: If no dictionary existed then
   4: While I₁ not null do
   5: If (k, v) ∈ I₁ then
   6: For k = 1 to n do
   7: H[K] ← v
   8: End for
   9: End if
10: If (k, v) ∈ I₁ then
11: For k = 1 to n do
12: H[K] ← a value within map size
13: End for
14: End if
15: End while
16: Else
17: Appending data to Dᵢₑᵢ.
18: End if
19: Using the dictionary to encode data next time.

Algorithm 2 To encode data for compressing before watermarking it.

**Input:** Data from the database basis of a user query.

**Output:** Data (Pₑᵢ) received from the database in encoded format.

1: Looking for data as per a user request from the source database.
2: Making dataframe (Dₑᵢ) from replied data (Pₑᵢ) and remove unnecessary part.
3: While Dₑᵢ is not empty do
4: For each dᵢ in Dₑᵢ do
5: If dᵢ ∈ Dₑᵢ then
6: Identifying Dₑᵢ(w) of dᵢ from Dₑᵢ
7: Encoding dᵢ by Dₑᵢ(w)
8: Else
9: Continue
10: End if
11: End for
12: End while
13: The data (Pₑᵢ) found from step 3 to 12 use for watermarking.

Where, dᵢ denotes field of Dₑᵢ and Dₑᵢ(w) the corresponding value of dᵢ from Dₑᵢ(w).

channels. Say, a patient current diastolic is δ₀ and systolic σᵣ. Additionally, the normal diastolic is δ₀ and systolic σᵣ as per the medical diagnostic chart. In the proposed method, the respected patient glucose will be altered as follows:

diastolic, Δₐ = δ₀ - δₐ
systolic, Δₜ = σᵣ - σₜ

These reduced values (Δₐ, Δₜ) will be transferred over the communication channels in a compressed format. In such a way, other prescription-related data would be encoded by replacing their respected code as per the dictionary and passed over the networks.

In the user end, the converted data is again reversed to yield the original data by matching the code with the dictionary. For decoding received data, the procedure is depicted in Algorithm 3. A preliminary work of our research on data compression was presented at a conference [59]. In the previous work, we researched to see whether the proposed encoding technique provides any spare performance with the existing scheme. We tested our proposed system with limited state-of-art works in prior research [59]. In the current research, we experimented with our proposed method to compress patient data and analyzed the results compared with more state-of-art works. We also worked to communicate patient data securely in this paper (section 4.2).

4.2. Security ensuring process

Watermarking is generally applied to ensure authentication by recognizing data sources and integrity by tracking manipulated data. The watermarking technique is also applied to data for stopping illegal distribution of them [60]. Although different watermarking techniques are used for images and documents based on their data properties, no approach has been implemented yet to secure live-text data while communicating over the networks. We present a watermarking procedure to secure patient live-text data while processing it remotely. The watermarking approach has two phases: embedding the identification information with original data; extracting the fused watermark later from the watermarked data to get actual data. The embedding and extracting process of the watermark is presented in Fig. 1. The identification message integrated with the actual data is known as a watermark.

The proposed watermark technique includes two phases: watermark generation and watermark embedding. Our strategy is designed to append both fragile and robust watermarks into health data for security. The robust watermark (Wₚₚ -hospital or health institution or domain information, or more) is inserted to detect authentic sources. Alongside, a fragile watermark (Wₑₑ) is created for identifying data alternation that may happen when transmitting them over the networks. This watermarking is applied without affecting the actual data. Algorithm 4 presents the watermark inserting procedure. The original encoded data with adding a watermark is referred to as watermarked data (Pₑₑₛₑ) and shared later from remote locations.

Algorithm 3 To decode data before passing the end-users.

**Input:** Received encoded data (Pₑᵢₒᵢ) after extracting watermark.

**Output:** Normal data (Pₑᵢᵢ) as per user request.
1: State Taking the encoded data (Pₑᵢₒᵢ).
2: While Pₑᵢₒᵢ is not empty do
3: For each dᵢ in Pₑᵢₒᵢ do
4: If dᵢ ∈ Dₑᵢ do
5: Identifying Dₑᵢ(k) of dᵢ from Dₑᵢ
6: Decoding dᵢ by Dₑᵢ(k)
7: Else
8: Continue
9: End if
10: End for
11: End while
12: Replying the resultant in data (Pₑᵢᵢ) found from step 2 to 11 to the user.

Where, dᵢ denotes field of Dₑᵢ and Dₑᵢ(k) the corresponding key of dᵢ from Dₑᵢ(k).

Algorithm 4 To insert watermark to encoded data yielded using Algorithm 2.

**Input:** Encoded patient data (Pₑᵢₒᵢ).

**Output:** Patient encoded data with watermark (Pₑᵢₑₛₑ).
1: To take the encoded data (Pₑᵢₒᵢ) from the source.
2: While Pₑᵢₒᵢ is not empty do
3: Selecting an identification message (Wₑₑ) from the data source node basing Algorithm 6.
4: Making a fragile watermark (Wₑₑ) from (Wₑₑ) and Pₑᵢₒᵢ basing Algorithm 7.
5: Generating watermarked data, Pₑᵢₑₛₑ ← Pₑᵢₒᵢ + Wₑₑ + Wₑₑ. * + * indicates embedding watermark with encoded data.
6: End while
7: Sending Pₑᵢₑₛₑ to the requester via the network in compressed format.

A reverse process is applied for extracting the watermark, and an extra step is used to check data manipulation. The attackers may alter the data during sharing, and tamper detection is needed by equalizing the fragile identification mark. The purpose of identifying tamper is to find any unauthorized and accepted data alternation. The recipient can generally know whether they receive correct or tamper watermarked data. As a result, the tamper detection step ensures to get the correct data by matching the previous fragile watermark with the newly generated one. The watermark extraction procedure is shown in Algorithm 5. This process takes the watermarked data (Pₑᵢₑₛₑ) as input and recovers the actual data again if their alternation has not happened.

The main issue of the proposed technique is forming a fragile and robust watermark from the received source data. We design an algorithm for generating a robust watermark fused with health data to gain
The first goal. Algorithm 6 lists different steps for making a robust watermark [61].

Algorithm 6 Creating a robust watermark to embed into actual data.

```
Input: Organization logo, hostname, or another identification messages of the source.
Output: Robust watermark (W_r) for inserting into patient encoded data (P_enc).

1: Getting different identification information of the sources.
2: To generate W_r by unifying the information got in step 1.
3: To use W_r for embedding it later with patient data.
```

We also design an algorithm to create a fragile watermark without influencing original data for achieving the second goal. We presented preliminary research work on the proposed new watermarking approach to a conference [61]. The algorithm implemented for generating a fragile watermark (f) in the previous work depended on a random number (R) and used an equation expressed in Equation (1). Our challenge was in what range we would select the random number. Thus, in our previous work, we emphasized the selection of R in different lengths and measured whether this R could detect tamper data. We also searched if they had any effect on manipulated data detection. Analyzing the previous result, we improved the algorithm by specifying the R in the current work and again performed our experiment. In addition, we sketched a flowchart to define operation procedure flow and a communication diagram to show dynamic interaction among different levels of the previous works.

\[ f = \mu + m + \zeta_{\text{len}} \]  

(1)

Here,

\( f \): generated fragile watermark

\( \mu \): character code average of chosen words

\( \zeta_{\text{len}} \): total character number of P_enc

Algorithm 7 presents the procedure of creating \( f \) to insert into data.

Algorithm 7 Making a fragile watermark taking encoded patient data (P_enc) and robust watermark W_r.

```
Input: Taking P_enc and W_r, from the source node.
Output: A fragile watermark (f) to insert into data.

1: Finding \( \phi_{\text{len}} \) first word length of \( P_{\text{enc}} \).
2: Generating \( \zeta_{\text{len}} \), the character length of the data.
3: Getting a random number R, where \( \phi_{\text{len}} < R < \zeta_{\text{len}} \), using \( \zeta_{\text{len}} \) as seed.
4: if data ! = null then
5: while data not terminate do
6: Splitting the data into blocks considering each word where \( \zeta_{\text{len}} \rangle R \).
7: Counting total chosen words, \( \theta_i = \sum_{i} \theta \), where \( \theta_i \) is the starting word of each data block and \( \theta \) is the number of chosen words
8: Calculating the character code average (\( \bar{\mu} \)) from chosen words \( \bar{\mu} = \frac{\bar{\mu} \cdot \bar{\zeta}}{\dot{\zeta}} \), where, \( \theta \) is the chosen word number, \( m \) number of characters for each word, and \( \dot{\zeta} \) code of each character.
9: end while
10: Measuring the character code average (\( \bar{\mu} \)) of the whole words of \( W_r \), using step 8
11: Creating a fragile watermark, \( f = \bar{\mu} + m + \zeta_{\text{len}} \)
12: end if
13: Inserting \( f \) later to \( P_{\text{enc}} \).
```

5. Data processing and communication flow

After receiving the user request, the system performs the different operations at different levels. The process of data flowing in various steps and the dynamic communication among them are stated in this section.

5.1. Data processing flow

The stated algorithms define how data will be compressed and watermarked in different phases after getting requests from users. This part presents the overall procedure flow of the data encoding and watermarking. The operation flow of the proposed technique is presented in Fig. 2. The figure shows what stage of the processing includes the different algorithms to perform the system operation. In the system, the Algorithm 1 is included to make dictionaries from the collected data. The Algorithm 2 works to encode the data received from the server with the help of dictionaries. The encoded data is then fed to Algorithm 4 for
embedding the watermark. This watermarked data pass over the communication channel for the user-end. The adding watermark is then extracted from the received data using the Algorithm 5 and checked whether the data has been altered or not? If the data is correct, the Algorithm 3 decodes them and responds to the user.

5.2. Data communication diagram

A communication diagram is presented in Fig. 3 to show dynamic interaction among different levels in data execution time. The boxes indicate the objects and arcs joining boxes refer to the interconnection between the objects. The direction of the forwarding message is shown using arrows adjacent to the arcs and labels for the message name. The exchange of message sequence is also presented with the number transmitted between the objects. According to Fig. 3, the user request is passed to the ServerInterface object sequentially through the UserInterface and then MiddlewareInterface objects. Data retrieved from the server is then encoded by the Encoder with the help of Dictionary objects and watermarked by the WatermarkEmbedder. The MiddlewareInterface again passes the watermarked data to UserInterface for extracting the watermark and decoding the data before responding to the user.

6. Results and discussion

We conducted a trial on anonymised patient data to test the performance of the proposed algorithms. We implemented a server and a client socket program using the python language to experiment on computers running by Ubuntu operating system. The anonymised patient prescription data was taken in JSON format and stored in MongoDB, a NoSQL database. The experiment was set up to test both data reduction and security. We conducted the lossless data compression as the patient data is more sensitive, and any alternation of data might cause severe illness in the patient. We tested the proposed approach and other eight existing methods (RLE, LZW, LZO, Deflate, BZ2, LZ4, LZMA, LZSS) currently used as state-of-the-art techniques for lossless patient data reduction. The compression performance of the proposed strategy was evaluated using various standard metrics-compression gain; compression time by comparing it with other similar compression mechanisms. The result found from the experiment tabulated in Table 4. The table shows that our proposed technique yields more compressed data than other existing ones for various data. The table data also notice that most conventional algorithms achieve more or less data compression except RLE and LZW, which were not fit for prescription data to reduce them. RLE and LZW data were excluded for further comparison in graphs as they increased data size instead of compressing.
**Fig. 3.** Communication diagram for data compression and watermarking.

**Table 4.** Compressed data using different techniques.

| Data | Size(B) | RLE | LZW | Deflate | LZO | LZ4 | BZ2 | LZSS | LZMA | Proposed Tech |
|------|---------|-----|-----|---------|-----|-----|-----|------|------|---------------|
| D1   | 979     | 1905| 3752| 322     | 446 | 459 | 389 | 416  | 393  | 278           |
| D2   | 1477    | 2897| 4840| 367     | 528 | 544 | 458 | 505  | 433  | 328           |
| D3   | 2738    | 5403| 7976| 523     | 780 | 805 | 613 | 764  | 589  | 387           |
| D4   | 3439    | 6779| 9016| 613     | 915 | 930 | 713 | 904  | 665  | 472           |
| D5   | 5418    | 11697| 13672| 1238   | 1264| 873 | 1261| 841  | 613  |               |
| D6   | 7221    | 14943| 14672| 1002   | 1583| 1619| 1041| 1606| 1037| 750           |
| D7   | 8657    | 15981| 16552| 1155   | 1854| 1874| 1162| 1882| 1177| 825           |
| D8   | 10882   | 20881| 18664| 1376   | 2234| 2246| 1324| 2299| 1361| 959           |
| D9   | 12397   | 24565| 21040| 1516   | 2493| 2543| 1454| 2567| 1501| 997           |
| D10  | 14815   | 28997| 23720| 1740   | 2853| 2924| 1653| 3071| 1701| 1145          |

**Fig. 4.** Comparison of compression gained by different techniques for data (D1-D5).

The compression gain of the different strategies was checked using the two Equations (2) and (3). The obtained results are presented in Fig. 4 and 5. The figures indicate that the compression rate using the proposed technique is higher than the other existing approaches for all sizes of data. The highest compression rate using the proposed method is more than 92%, while the highest rate is below 89% by the traditional other ones.

\[ r = \frac{\theta}{\psi} \]

(2)

Here, \( r \): compression ratio, \( \psi \): original data volume, \( \theta \): size of data after compression

**Compression gain = (1 – r) \times 100%**

(3)

An analysis was also done to compare the compression time of various conventional approaches and our proposed one required for data reduction. The time needed to compress data is presented in Fig. 6 and 7. According to those figures, the proposed technique shows a moderate compression time. Although some methods (Deflate, LZ0, LZ4) take lesser time than the proposed one, their compression gain is lower than our technique revealed from Fig. 4 and 5.

We tested the system performance in terms of data volume and accuracy of the watermarking technique used for ensuring integrity and authentication of live-text data while passing over the networks. The experimental procedure inserted the watermark into health data from the server-side before sending watermarked data via the networks. The user application received the watermarked data and extracted the watermark from them. The received information was then checked using the application to detect any data tampering on the user side. We conducted the test on various sizes of patient data and evaluated the performance in three sections: the first part measured the watermarked data proportion surged to embed a watermark with the patient data; the second...
section searched the required time to compress watermarked data and actual data; the last part considered the deliberate attacks to check the accuracy of detecting the tamper.

In the first phase, our focus was an analogy on the ratio between the watermarked and original data. The experiment aimed to find the increased data volume after adding the watermark to the actual data. Fig. 8 presents the size of the original data, watermark, watermarked data, and the ratio of the watermarked data and watermark. The figure shows that their overall proportion is small, which would not significantly influence the data passing in low latency. Additionally, it is noticeable that their ratio is decreased gradually with surging the volume of sample data.

In the second part, our goal was to compare compression time between the original and watermarked data received after adding the watermark. The compression time was measured firstly on various data sizes, and the same work was then done by adding a watermark with the original data. Fig. 9 depicts the comparison chart of required time in both cases. The figure indicates that the extra time needed for compressing watermarked data is minimal and moves in parallel with the normal data.

In the last section of our trial, we measured the accuracy of the watermarked data by altering them. The watermarked data could be exchanged by the attackers in the way of data communication. Therefore, we evaluated the accuracy of tampering with data taking into account different probable attacks that may cause. We implemented a Python simulator and performed the test by manipulating the data in different ways: insertion, transformation, rearrangement, and deletion. Later we altered the data 100000 times differently and tabulated the results from each deliberate attack to calculate the accuracy. We used the same data sample for various categories of attacks to generate a better comparison. We used Equation (4) to measure the accuracy.

\[
\text{Accuracy} = \frac{\sum C_i}{\sum I_i} \times 100\% 
\]

Here, \(C_i\) and \(I_i\) indicate the ‘Total number of detected data correctly’ and ‘Total iteration of manipulated data’ respectively.

The accuracy obtained from the four kinds of possible attacks is presented in Fig. 10. The figure indicates that transformation provides more than 99.63% accuracy for all data except D1, which shows 98.35%. The remaining three data manipulation experiments yields about 100% accuracy.

The analysis of the resulting data provided the following indication for the system. By comparing the compression rate from Fig. 4 and 5, and compression time from Fig. 6 and 7, we observed that the LZ0 contained the lowest time among the all techniques, but its compression rate (around 80%) was far below than the proposed approach (about 92%). The two other traditional methods (Deflate and LZ4) showed a lower compression time, but their compression rates were also poor than the proposed technique. Therefore, our method outperformed all other existing ones in terms of compression rate by taking moderate compression time. Moreover, the ratio of the watermark embedded with the original data shown in Fig. 8 was small enough, which did not affect the system significantly. The comparison of compression time presented in Fig. 9 provided a slight gap between them, which also had no significant effect on the system. In addition, the accuracy of identifying the
watermarked data was around 100% except only for small data D1 for transformation with 98%. The result reveals that the current proposed technique is robust enough to ensure the integrity and authentication of live data transferred over slow-speed networks. In the previous article [61], the tamper detection accuracy ranged from around 90% to 100% based on the random number (R). In the current method, the accuracy rate was improved and showed approximately 98% to 100%. In addition, we received a better performance for the ratio of watermark and watermarked data. In the recent work, their ratio got a maximum of 0.06, but it went up to 0.08 in the previous article. In this paper, the evaluation result of our proposed encoding strategy with the existing state-of-art works also provided a better performance to compress data.

The study results indicate that the system designed for our proposed algorithms performs well in communicating patient data securely in a more compressed format. The strength of the research is that it develops a watermarking method for live-text data while transmitting over the communication channels to ensure data authentication and integrity. To our knowledge, although there are various watermarking strategies for text documents and images, this is a novel watermarking approach for live-text data. Additionally, this study proposed a new encoding technique based on a dictionary considering the patient data property, which could reduce data size higher than existing techniques (RLE, LZW, Deflate, LZO, LZ4, BZ2, LZSS, LZMA). We here only use lossless compression methods on patient data. Our proposed algorithms perform better than other conventional strategies because they provide two advantages when replacing the data with encoding. First, the data size would be short than the original one. Second, the data consist of the more repetitive sequences or sub-strings that help compress themselves later using dictionary methods. Besides, the proposed watermarking technique shows an accuracy of around 100% in the case of tamper detection. This is caused due to making a fragile watermark by considering the live patient data that will be transferred over the communication channels. Therefore, our proposed system would help the end-users to get patient data in low latency without any alternation of data, which improves the quality of patient medication.

7. Conclusion

The paper aims to compress patient data highly to share securely while transferring them via the bad networks. The compression technique is more important in decreasing the data volume and leading to faster data transmission over the communication channel with low latency. Besides, data integrity and authentication are also essential to secure them for passing over the network in real-time. This paper defines algorithms to compress patient data through encoding and recovery from the user side. Unlike the existing compression methods, which are usually dictionary-based schemes, the proposed method encodes the data to compress them using different diagnostic data and measurement charts. The experimental results show that the defined technique outperforms the current traditional methods. In addition, a novel watermarking process is introduced to ensure live-text data integrity and authentication. The outcome tested based on the proposed approach is robust enough to identify the possible data alternation caused by the attackers. The results also recognize that the watermark volume does not affect the system performance significantly due to having a small size and lower compression time. Overall, the proposed method could support the user to receive correct data without any alternation in low latency and get information geographically from a distant authentic source.

Declarations

Author contribution statement

Subrata Kumar Das: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Wrote the paper.
Mohammad Zahidur Rahman: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

This work was supported by the Information and Communication Technology Division, Bangladesh.

Data availability statement

Data will be made available on request.

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] H.K.Y. Almamthi, K.T. Win, E. Vlahu-Gjorgievska, Barriers and facilitators that influence telemedicine-based, real-time, online consultation at patients’ homes: systematic literature review, J. Med. Internet Res. 22 (2) (2020) e16407.
[2] G.G. Sagaro, G. Batineni, F. Amenta, Barriers to sustainable telemedicine implementation in ethiopia: a systematic review, Telemed. Rep. 1 (1) (2020) 8–15.
[3] I. Keshet, A. Odoh, Security and privacy of electronic health record concerns and challenges, Egypt. Inform. J. 22 (2) (2021) 177–183.
[4] S. Shukla, M.F. Hassan, M.K. Khan, L.T. Jung, A. Awang, An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment, PLoS ONE 14 (11) (2019) e0224934.
[5] I. Alsmadi, R. Burdwell, A. Alrduw, A. Wahbeh, M. Al-Qudah, A. Al-Omari, Network security, in: Practical Information Security, Springer, 2018, pp. 121–138.
[6] S. Josha, N. Nishanth, A study on different attacks on transport, network and data link layer in tcp/ip, Int. J. Innovat. Res. Comput. Commun. Eng. 5 (6) (2017).
[7] J. Andress, Chapter 1 - what is information security?, in: J. Andress (Ed.), The Basics of Information Security, second edition, Syngress, Boston, 2014, pp. 1–22, https://www.sciencedirect.com/science/article/pii/B9780128007440000014.
[8] D. Mata-Mendoza, M. Cedillo-Hernandez, F. Garcia-Ugarte, A. Cedillo-Hernandez, M. Nakano-Miyazato, H. Perez-Mena, Secured telemedicine of medical imaging based on dual robust watermarking, Vis. Comput. (2021) 1–18.
[9] A.P. Srithar, F. Lakshmi, An efficient lossless medical data compression using lzw compression for optimal cloud data storage, Ann. Roman. Soc. Cell Biol. 25 (6) (2021) 17144–17160.
[10] U. Narayanan, V. Paul, S. Joseph, A novel system architecture for secure authentication and data sharing in cloud enabled big data environment, J. King Saud Univ. Comput. Inf. Sci. 34 (6, Part B) (2022) 3121–3135.
[11] V.F. Rodrigues, E.P. Paim, R. Kunst, R.S. Antunes, C.A. da Costa, R. da, Ronz Righi, Exploring publish/subscribe, multilevel cloud elasticity, and data compression in telemedicine, Comput. Methods Programs Biomed. 191 (2020) 105403.
C. Khanduja, A. Arican, M. Hashim, 2022, *IEEE Transactions on Image Processing*.

S.-J. K. Das and M. Z. Rahman, 2021, *Heliyon*.

A. A. Abdellatif, A. Emam, C.-F. Chiaierini, A. Mohamed, and M. Ismail, 2018, *IEEE Transactions on Biomedical Engineering*.

A. K. Chaudhary, R. Mehrotra, M. Ansari, and P. Tripathi, 2021, *International Journal of Biomedical Engineering and Computing*.

X. Liu, P. An, Y. Chen, X. Huang, 2021, *IEEE Access*.

Y. Cao, H. Zhang, Y.-B. Chei, H. Wang, S. Xiao, 2020, *IEEE Access*.

C. M. E. G. Lymos, 2020, *Data Engineering and Technology*.

J. Dong, 2020, *Content-aware compression for big textual data analysis, Ph.D. thesis, University of Cork*.

H. Jiang, S.-J. Lin, 2020, *IEEE Transactions on Information Forensics and Security*.

M. Hashim, A. H. M. Aman, A. H. A. Hashim, and S. I. Rahman, 2021, *International Journal of Advanced Computer Science and Applications*.

V. Khanduja, S. Chakraverty, O. P. Verma, 2020, *IEEE Transactions on Information Forensics and Security*.

U. Khadom, M. M. Ishba, M. A. Asam, S. Khalidi, S. Rho, N. Chailamkurti, 2020, *IEEE Access*.

D. S. Chauhan, A. K. Singh, B. Kumar, and J. P. Saini, 2020, *IEEE Transactions on Information Forensics and Security*.

D. S. Choi, 2020, *IEEE Transactions on Information Forensics and Security*.

A. N. S. A. Hamza, 2020, *IEEE Transactions on Information Forensics and Security*.

S. Thakur, A. K. Singh, S. P. Gherera, A. Mohan, 2020, *IEEE Transactions on Information Forensics and Security*.

N. S. Kumaruddin, A. Kamsin, L. Y. H. Hor, and H. Rahman, 2020, *IEEE Transactions on Information Forensics and Security*.

A. N. S. A. Hamza, 2020, *IEEE Transactions on Information Forensics and Security*.

Q. W. Liang, V. Imanloues, 2020, *IEEE Transactions on Information Forensics and Security*.

G. W. Liang, V. Imanloues, 2020, *IEEE Transactions on Information Forensics and Security*.

O. G. W. Liang, V. Imanloues, 2020, *IEEE Transactions on Information Forensics and Security*.

C. Xiao, C. Zhang, C. Zheng, 2020, *IEEE Transactions on Information Forensics and Security*.

G. Tayan, M. N. Khatir, Y. M. Alginabi, 2020, *IEEE Transactions on Information Forensics and Security*.

K. Thongkorp, T. Amornsakas, 2020, *IEEE Transactions on Information Forensics and Security*.

F. N. Li, J. Zhang, M. N. Noori, 2020, *IEEE Transactions on Information Forensics and Security*.

M. Arican, K. Polat, 2019, *IEEE Transactions on Information Forensics and Security*.

C. J. Deepu, C.-H. Heng, Y. Lian, 2018, *IEEE Transactions on Information Forensics and Security*.

T. Tekeste, H. Saleh, B. Mohammad, and M. Ismail, 2018, *IEEE Transactions on Information Forensics and Security*.

S. K. Das, M. Z. Rahman, 2018, *IEEE Transactions on Information Forensics and Security*.

S. K. Das, M. Z. Rahman, 2018, *IEEE Transactions on Information Forensics and Security*.