MPEG Video Security Using Motion Vectors and Quadtrees

Anshul SINGHAL, Manik Lal DAS
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar-382007, INDIA
anshul_singhal@daiict.ac.in, maniklal_das@daiict.ac.in

Abstract: Securing multimedia data communication over public channel is a challenging task for protecting digital content from content piracy. Anti-piracy, digital watermarking and ownership verification are some mechanisms for authenticating digital content. The size of multimedia data being generally huge makes it difficult to carry out data encryption and compression for real time data communication. In this paper, we discuss the partial encryption approach in which a part of the compressed data is encrypted and rest of data remains unencrypted. The approach results in significant reduction of computation and communication time.

Key-Words: Quadtree, Video compression, Motion vectors, Partial encryption, Authentication.

1. Introduction

The use of multimedia applications involving images and videos has increased drastically in recent years. Increase in distribution of such multimedia content over network is due to applications like video on demand, video telephony and online video sharing. Recent advances of digital technologies allow alteration of multimedia data simpler and sometimes undetectable by human audio-visual system. One possible reason could be digital contents are easily accessible through Internet by illegal means, and using latest software, tools, one can try to duplicate and redistribute digital content without being detected. Therefore, securing multimedia data from adversary is necessary for protecting digital content rights, detecting data piracy and ensuring content ownership. In order to provide data confidentiality encryption is an important cryptographic primitive. As multimedia data (e.g. image, audio, video) comes in large size, conventional encryption process would consume huge bandwidth of communication medium when data is being transmitted over insecure channel like Internet. The compression algorithm reduces the actual data size and has been used in several applications before enciphering the data. Typically, multimedia data is fed as an input to a compression algorithm. Then, the compressed data is being encrypted by a standard encryption algorithm [1]. The sequence of data flow is provided in Figure 1. The process is then reversed at the decoder-end. Over the years, researchers have proposed algorithms that combine compression and encryption algorithms [2] to reduce the processing time. Selective encryption like encryption for the I-Intra frames [3] of the MPEG video provides a very limited level of security [4] as the rest of the parts of the video can be generated using rest of the frames. There are many methods [5], [6], [7], [8] that discuss the similar approach but none of them is able to bring down the processing time significantly.

![Figure 1: (a) Traditional approach, (b) Partial encryption approach](image-url)
In this paper, we discuss an approach called Partial Encryption [5], where a part of the data is encrypted to make the whole multimedia data secure. The entire data is divided into two parts - Important data and Unimportant data. The important portion of data is then stored in a data structure called Quadtree and then the tree is encrypted by a standard encryption algorithm. The unimportant data is left unencrypted while the multimedia data is transmitted over insecure channel such as Internet. We show that even a large part of the data is left unencrypted, it is difficult to recover the original data without decrypting the encrypted part. The proposed approach of partial encryption using Quadtree is secure and efficient. The remaining of the paper is organized as follows. Data decomposition is discussed in Section 2. The MPEG video compression scheme is discussed in Section 3. The approach of Partial encryption on MPEG-video is discussed in Section 4. We conclude the paper with Section 5.

2. Data Decomposition and Encryption

Compression algorithms decompose various logical parts of multimedia data like homogeneous areas, pixel information and colour information. After decomposing, video algorithm like MPEG [9] transforms information into various coefficients. Some of these information are regarded as important part. Considering all such logical important parts to be as one important unit, it gives significant amount of information about the original multimedia data, which can be used for recognizing, approximation and reconstructing the original multimedia data. As a result, only the important part is required to be encrypted and rest of other parts of the video remains unencrypted. It is noted that having all unencrypted portions available one cannot figure out the meaning of the original multimedia data. A significant reduction in processing time can be achieved when the relative size of the important data is small. When communicating over a public channel (e.g. Internet), a symmetric key algorithm such as AES, IDEA [1] can be used to encrypt the data.

3. MPEG Video Compression

Motion Pictures Experts Group (MPEG) [10] is the name for family of Standards used for coding audio visual format. In general, video compression works with throwing as much information as possible that has very minimal effect on the viewing experience of the users. A video can be imagined as a stream of images (frames) viewed in a fast manner, around 24-30 frames per second. The compression scheme takes advantage of two redundancies. One is the spatial redundancy which allows duplicate information present within the frame such as for the colour information. For example, the pixels belonging to same object will have the same colour information. This type of redundancy is overcome by using Discrete Cosine Transform [11]. The other one is the temporal redundancy, which allows information that repeats itself over subsequent frames. For a video of order of 25 frames per second, the subsequent frames will be very similar. Therefore, instead of coding each frame separately the difference between the frames is encoded. The MPEG layered format is given in Figure 2. The Sequence layer contains general information about the video such as the vertical and horizontal heights of the frames, height/width ratio or the bit rate. Group of pictures (GOP) are grouped into frames such as the frames in a GOP do not have a significant amount of change as compared to the frames in the other GOPs.

Frame layer is primarily the coding unit. It contains the picture’s position in the display order. Slice layer handles errors; Macro-block layer is for coding the motion of objects in the frames; and Block Layer carries the colour information. A frame is divided into a grid of macro-block of a particular size.
The frames within a GOP may be of various types depending upon how they are coded. The frame can be an I-frame, a B-frame or a P-frame. In one GOP there is one I-frame and rest are B or P frames. An I-frame is divided into grid of macro-block and each pixel value is encoded for the block. But, this is not the same for B or P frames (Inter frames). The encoder will try to match the block on inter frames to the previously encoded frames known as the reference frame. This is done by Block matching algorithm [12] that output motion vectors, which is the change in position of the current block with respect to the matching block in the reference frame. This process is termed as motion estimation. Once the matching block is found, encoder compares the two blocks and obtains the differences between them which may be the difference in the pixel information. Therefore, from frame to frame within a GOP residual errors and motion vectors are stored.

4. Partial Encryption of MPEG Videos

Video compression consists of two parts - one is the motion estimation (motion
vectors) and other is the residual error. The motion vectors contain important information and they must be encrypted. Otherwise, an image frame may be used to provide approximations to successive frames. The motion vectors belonging to a particular solid object may be same and can be grouped together in certain data structure. Figure 3 shows a candidate motion vector of all macro-block in a 4 x 4 frame. It is observed that blocks 1, 2, 5 and 6 have a motion vector in common and so does blocks 11, 12, 15 and 16. This redundancy can be removed by using Quadtree data structure [5]. A Quadtree is a rooted tree in which each internal node has exactly 4 children. The algorithm that stores data in a quadtree is usually followed a top-down approach. If the entire grid is homogeneous then only one node (root node) is used to represent the whole grid. Otherwise, the image is partitioned into four quadrants and four corresponding children are added to the root of this tree. The algorithm then recursively examines each quadrant using each of the four children as the root of the new subtree. The values attached with the nodes are the parameters describing the block. This whole process is known as Quadtree compression. Such a compression can be of two types. One is defined as the Lossless compression and the other one is defined as the Lossy compression scheme. The lossy version is almost similar to it lossless counterpart except for the test where a specific threshold is kept. Blocks are partitioned further if they differ over a certain value. Therefore, the test of homogeneity is replaced by the test of similarity. If the internal nodes are labelled as 0 and leaf nodes 1, then the skeleton of the tree can be written as 0101011111111 [5], [7]. The residual error also provides the outline of the moving object that sometimes are not accurately predicted by motion vectors. As a result, coding only motion vectors and intra frames (reference frames) may lead to vulnerable state and thereby, residual errors are being encoded. As residual errors are simply an image frame, they can easily be compressed using the quadtree compression. The Quadtree decomposition and Quadtree structure of an image are shown in Figure 4.

![Figure 3. Motion vector of all macro-block](image)

![Figure 4. (a) Quadtree decomposition, (b) The Quadtree](image)
leaves from left to right, regardless of the level of the tree. The leaf ordering I is being illustrated with an example as shown in Figure 5. The leaf ordering I for this example will be 1100001011, where 0 represents a black pixel and 1 represents a white pixel. For the case of leaf ordering II, it is based on tree levels. Furthermore, in the Leaf ordering II leaves are encoded one level at a time.

\[ \begin{array}{cccc}
1 & 7 & 10 \\
 & 8 & 9 \\
2 & 5 & 6 \\
3 & 4 \\
\end{array} \]

**Figure 5: Leaf nodes ordering using Leaf Ordering I**

5. Conclusion

We discussed the approach of partial encryption of videos using Quadtree in which the compression and encryption based on data decomposition are being explained. With the help of partial encryption only important part of the multimedia data is encrypted, which is significantly less size than the whole multimedia data size. It is observed that from frame to frame there is a minor change in the movements of object. The motion vectors belonging to the same object are same and same motion vectors can be grouped together using Quadtree compression algorithm. Figure 6 shows motion vectors of blocks from one frame to another.

When applying the Quadtree algorithm, the blocks 11, 12, 15 and 16 are grouped together in one quadrant. If the object in the next frame moves to the left a little then the motion vectors for the next frame is obtained as shown in Figure 7. These blocks are not grouped together as they do not fall in the same quadrant. The Quadtree structure for the Figure 7 may have more number of bits. Therefore, what is required is to identify the motion vectors corresponding to the object.

\[ \begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
0 & 01 & 11 & 12 \\
13 & 14 & 15 & 16 \\
\end{array} \]

**Figure 6. Motion vectors of blocks from one frame to another**

\[ \begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
0 & 01 & 11 & 12 \\
13 & 14 & 15 & 16 \\
\end{array} \]

**Figure 7. Motion vectors of blocks**

If an object is moving with an increasing velocity at a constant rate, for example 2 in the frame 1; 4 in the frame 2; 6 in the frame 3 and so on; one can encrypt the initial velocity with the acceleration and therefore, can save a significant number of bits.

References


